МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
МАШИНОСТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ (МАМИ)

О. В. Миловзоров, И. Г. Панков

ОСНОВЫ ЭЛЕКТРОНИКИ

УЧЕБНИК ДЛЯ СПО

5-е издание, переработанное и дополненное

Рекомендовано Учебно–методическим отделом среднего профессионального образования в качестве учебника для студентов образовательных учреждений среднего профессионального образования

Книга доступна в электронной библиотечной системе biblio-online.ru

Москва ▪ Юрайт ▪ 2016

УДК 621.38(075.32)
ББК 32.85я723
М60

***Авторы*:**

Миловзоров Олег Владимирович — профессор, кандидат технических наук, профессор Рязанского института (филиала) Московского государственного машиностроительного университета (МАМИ), член-корреспондент Российской инженерной академии;

Панков Иван Григорьевич — кандидат технических наук, профессор, заведующий кафедрой механико-технологических дисциплин факультета дневного обучения, директор Рязанского института (филиала) Московского государственного машиностроительного университета (МАМИ), член-корреспондент Российской инженерной академии.

***Рецензенты*:**

кафедра автоматизации информационных и технологических процессов Рязанской государственной радиотехнической академии (заведующий кафедрой — доктор технических наук, профессор *Мусолин А. К.*);

Белов Б. И. — доктор технических наук, профессор Московского государственного технического университета им. Н. Э. Баумана.

Миловзоров, О. В.

М60 Основы электроники : учебник для СПО / О. В. Миловзоров, И. Г. Панков. — 5-е изд., перераб. и доп. — М. : Издательство Юрайт, 2016. — 407 с. — Серия : Профессиональное образование.

ISBN 978-5-9916-5351-0

В учебнике рассмотрен широкий круг вопросов, касающихся основ полупроводниковой электроники, аналоговой и цифровой схемотехники. Описана работа полупроводниковых приборов — диодов, биполярных и полевых транзисторов и схем на их основе; схемотехника аналоговых устройств на основе операционных усилителей, силовая электроника. Освещены вопросы цифровой схемотехники, включая основы алгебры логики, простейшие логические элементы, комбинационные и последовательностные устройства, полупроводниковые запоминающие устройства, микропроцессоры и интерфейсные схемы, программируемые логические интегральные схемы. Рассмотрены микроархитектуры современных процессоров.

Для студентов образовательных учреждений среднего профессионального образования, обучающихся по инженерно-техническим направлениям и специальностям.

УДК 621.38(075.32)
ББК 32.85я723

© Миловзоров О. В., Панков И. Г., 2013
© ООО «Издательство Юрайт», 2016

ISBN 978-5-9916-5351-0

Оглавление

Предисловие 9

Глава 1. Полупроводниковые элементы и основы микроэлектроники 12

1.1. Физические основы полупроводниковой электроники 12
1.1.1. Виды электронных приборов 12
1.1.2. Электрофизические свойства полупроводников 13
1.1.3. *P-n*-переход и его свойства 17
1.1.4. Основные технологические процессы изготовления *p-n*-переходов 20
1.2. Полупроводниковые диоды 21
1.2.1. Выпрямительные диоды 23
1.2.2. Стабилитроны 24
1.2.3. Диоды Шоттки 26
1.2.4. Варикапы 28
1.2.5. Светодиоды 28
1.2.6. Фотодиоды 29
1.2.7. Оптроны 30
1.3. Биполярные транзисторы 31
1.3.1. Схема с общей базой 33
1.3.2. Схема с общим эмиттером 35
1.3.3. Входные и выходные характеристики схемы с общим эмиттером 37
1.3.4. Схема с общим коллектором 40
1.3.5. Технология изготовления биполярных транзисторов 42
1.4. Полевые транзисторы 43
1.4.1. Полевые транзисторы с управляющим *p-n*-переходом 44
1.4.2. Полевые транзисторы с изолированным затвором 46
1.5. Тиристоры 50
1.6. Основы микроэлектроники 52
1.6.1. Технология полупроводниковых ИМС 53
1.6.2. Компоненты ИМС 55
Контрольные вопросы и задания 59

Глава 2. Аналоговая схемотехника.....61
2.1. Усилители 61
2.1.1. Усилители и их место в электронных устройствах 61
2.1.2. Основные параметры усилителей.....63
2.2. Каскады усилителей низкой частоты 66
2.2.1. Каскады на биполярных транзисторах 66
2.2.2. Отрицательная обратная связь в усилителях.....69
2.2.3. Усилительные каскады на полевых транзисторах... 71
2.2.4. Дифференциальный каскад 74
2.3. Выходные каскады усилителей 76
2.3.1. Режимы работы выходных каскадов усилителей....76
2.3.2. Однотактные и двухтактные выходные каскады.....79
2.4. Операционные усилители 81
2.4.1. Свойства операционных усилителей.....81
2.4.2. Основные схемы включения операционных усилителей 84
2.5. Вычислительные схемы на основе операционных усилителей 87
2.5.1. Схема суммирования.....88
2.5.2. Схема вычитания 88
2.5.3. Схемы интегрирования и дифференцирования 90
2.5.4. Функциональные преобразователи 94
2.6. Генераторы на основе операционных усилителей 97
2.6.1. Мультивибратор 97
2.6.2. Генератор пилообразного напряжения 98
2.7. Компараторы, цифроаналоговые и аналого-цифровые преобразователи 100
2.7.1. Цифроаналоговые преобразователи 101
2.7.2. Аналого-цифровые преобразователи..... 103
2.8. Источники вторичного электропитания 106
2.8.1. Структуры источников электропитания..... 106
2.8.2. Выпрямительные устройства 108
2.8.3. Сглаживающие фильтры 111
2.8.4. Линейные стабилизаторы напряжения 113
2.8.5. Импульсные стабилизаторы напряжения 115
2.9. Силовые устройства на основе тиристоров и мощных транзисторов 117
2.9.1. Управляемые выпрямители..... 117
2.9.2. Инверторы и преобразователи частоты..... 120
2.9.3. Тиристорное управление двигателем постоянного тока..... 122

2.10. Электромагнитная совместимость электронных устройств ... 125
2.10.1. Помехи в цепях питания ... 125
2.10.2. Экранирование электронных устройств ... 128
2.10.3. Использование гальванической развязки ... 129
2.10.4. Конструктивные методы борьбы с помехами ... 132
Контрольные вопросы и задания ... 134

Глава 3. Основы схемотехники цифровых устройств ... 136
3.1. Двоичная система счисления ... 136
3.2. Алгебра логики ... 141
3.2.1. Основные теоремы и положения алгебры логики 142
3.2.2. Булевы функции ... 144
3.2.3. Минимизация булевых функций ... 146
3.2.4. Минимизация булевых функций с помощью карт Карно ... 147
3.2.5. Алгебра логики и цифровые электронные схемы ... 150
3.3. Ключевые схемы ... 154
3.3.1. Ключевая схема на биполярном транзисторе ... 155
3.3.2. Ключевая схема на комплементарных транзисторах ... 158
3.4. Логические элементы интегральных микросхем ... 160
3.4.1. Транзисторно-транзисторные логические элементы ... 160
3.4.2. Логические элементы на КМОП-транзисторах ... 164
3.5. Дешифраторы и шифраторы ... 166
3.5.1. Дешифраторы ... 167
3.5.2. Шифраторы ... 172
3.6. Распределители и мультиплексоры ... 174
3.6.1. Распределители ... 174
3.6.2. Мультиплексоры ... 175
3.6.3. Реализация логических функций на основе мультиплексоров ... 176
3.7. Сумматоры ... 180
3.7.1. Синтез одноразрядного сумматора ... 180
3.7.2. Последовательный многоразрядный сумматор ... 182
3.7.3. Параллельные сумматоры ... 183
3.8. Арифметико-логические устройства и матричные умножители ... 186
3.9. Триггеры ... 189
3.9.1. Триггер с установочными входами (*RS*-триггер) ... 190

3.9.2. Триггер задержки (*D*-триггер) ... 193
3.9.3. *T*-триггер ... 194
3.9.4. *JK*-триггер ... 195
3.10. Счетчики ... 197
3.11. Регистры ... 203
3.11.1. Регистровые файлы ... 204
3.11.2. Регистры сдвига ... 205
3.11.3. Универсальные регистры ... 206
Контрольные вопросы и задания ... 207

Глава 4. Схемотехника программируемых цифровых вычислительных устройств ... 210
4.1. Запоминающие устройства ... 210
4.1.1. Важнейшие параметры ЗУ ... 211
4.1.2. Классификация полупроводниковых ЗУ ... 213
4.1.3. Структура адресных ЗУ ... 217
4.2. Запоминающие устройства для хранения постоянной информации ... 219
4.2.1. Масочные ЗУ ... 220
4.2.2. ЗУ типа PROM ... 221
4.2.3. ЗУ типов EPROM и EEPROM ... 222
4.2.4. Флэш-память ... 225
4.3. Запоминающие устройства для хранения оперативной информации ... 227
4.3.1. Статические ЗУ ... 227
4.3.2. Динамические ЗУ ... 229
4.4. Микропроцессоры ... 231
4.4.1. Структура и принципы работы микропроцессорной системы ... 233
4.4.2. Режимы обмена в микропроцессорной системе ... 239
4.5. Однокристальный микропроцессор Intel 8086 (К1810 ВМ86) ... 242
4.5.1. Структура МП Intel 8086 ... 243
4.5.2. Мультиплексирование информационных линий 246
4.5.3. Сегментация памяти ... 246
4.5.4. Управляющие сигналы МП Intel 8086 ... 248
4.5.5. Структура команды МП Intel 8086 ... 249
4.6. Режимы адресации и система команд микропроцессора Intel 8086 ... 250
4.6.1. Режимы адресации ... 250
4.6.2. Система команд ... 253

4.7. Тенденции развития однокристальных микропроцессоров и систем на их основе ... 265
4.7.1. Основные этапы развития однокристальных микропроцессоров ... 270
4.8. Микроконтроллеры ... 283
4.9. Интерфейсные устройства ... 286
4.9.1. Интерфейсы и интерфейсные БИС ... 289
4.9.2. Шинные формирователи и буферные регистры ... 290
4.9.3. Параллельные периферийные адаптеры ... 291
4.9.4. Программируемые связные адаптеры ... 293
4.10. Программируемые контроллеры ... 297
4.10.1. Программируемые контроллеры прерываний ... 297
4.10.2. Контроллеры прямого доступа к памяти ... 301
4.10.3. Программируемые интервальные таймеры ... 302
4.11. Средства программируемой матричной логики ... 304
4.11.1. Программируемые логические матрицы ... 304
4.11.2. Программируемая матричная логика ... 307
4.11.3. Модернизация схем ПЛМ и ПМЛ ... 308
4.11.4. Базовые матричные кристаллы ... 311
4.12. Программируемые логические интегральные схемы ... 314
4.12.1. Средства обеспечения программируемости ПЛИС ... 314
4.12.2. Сложные программируемые логические устройства (CPLD) ... 317
4.12.3. Программируемые пользователем вентильные матрицы (FPGA) ... 319
4.12.4. СБИС программируемой логики «система на кристалле» ... 325
4.13. Средства автоматизированного проектирования электронных устройств ... 328
4.13.1. Автоматизированное проектирование электронных устройств ... 328
4.13.2. Обзор современных пакетов автоматизированного проектирования электронной аппаратуры ... 330
Контрольные вопросы и задания ... 333

Глава 5. Микроархитектуры процессоров ... 336
5.1. Микроархитектура P6 ... 336
5.1.1. Структура процессоров P6 ... 338
5.1.2. Подсистема упорядоченной предварительной обработки ... 338

5.1.3. Подсистема исполнения с изменением последовательности ... 341
5.1.4. Набор исполнительных блоков ... 342
5.1.5. Подсистема упорядоченного завершения ... 344
5.1.6. Подсистема памяти ... 345
5.1.7. Порядок выполнения команд программы ... 346
5.2. Микроархитектура NetBurst ... 351
5.2.1. Структура процессоров NetBurst ... 352
5.2.2. Входная подсистема ... 352
5.2.3. Подсистема исполнения с изменением последовательности ... 356
5.2.4. Подсистема исполнения ... 357
5.2.5. Подсистема памяти ... 359
5.2.6. Описание конвейера ... 360
5.2.7. Система повторного исполнения микроопераций 362
5.2.8. Технология Hyper Threading ... 364
5.3. Микроархитектура многоядерных процессоров ... 366
5.3.1. Микроархитектура Nehalem ... 367
5.3.2. Перспективы развития многоядерных процессоров Intel ... 380
Контрольные вопросы и задания ... 381

Заключение ... 383
Приложения ... 385
Приложение 1. Примеры условных обозначений некоторых активных и пассивных элементов на принципиальных электрических схемах ... 385
Приложение 2. Примеры условных обозначений некоторых аналоговых и цифровых интегральных элементов на принципиальных электрических схемах ... 387
Приложение 3. Ряды предпочтительных номиналов резисторов и конденсаторов ... 390
Предметный указатель ... 393
Литература ... 406

Посвящается
Владимиру Петровичу Миловзорову

Предисловие

Обучение высококвалифицированных специалистов в области конструкторской и технологической подготовки машиностроительного производства невозможно без освоения основ полупроводниковой аналоговой и цифровой схемотехники. В современном машиностроительном производстве широко применяются самые разнообразные технические средства, базирующиеся на электронике. Это универсальные металлорежущие станки и станки с ЧПУ, промышленные контроллеры, роботы-манипуляторы, гибкие производственные модули и системы, автоматические транспортные и складские системы. Наконец, современное производство немыслимо без повсеместного использования современной компьютерной техники. Фундаментом для изучения всего многообразия технических средств, используемых в современном автоматизированном машиностроительном производстве, является дисциплина «Основы электроники».

В результате изучения дисциплины «Основы электроники» студент должен:

знать

— основные типы и области применения электронных приборов и устройств;

— параметры современных полупроводниковых устройств (усилителей, генераторов, вторичных источников питания, цифровых преобразователей, микропроцессорных управляющих и измерительных комплексов);

уметь

— разрабатывать принципиальные электрические схемы;

— проектировать типовые электрические и электронные устройства;

владеть

— навыками работы с электротехнической аппаратурой и электронными устройствами.

Базовые знания, полученные при изучении дисциплины «Основы электроники», особенно ее раздела «Микро-

процессорная техника», необходимы для освоения таких дисциплин, как «Управление системами и процессами», «Аппаратные и программные средства систем управления», «Автоматизация производственных процессов». Материал, изложенный в учебнике, охватывает все основные направления современной полупроводниковой схемотехники, включая основы полупроводниковых приборов, аналоговую и цифровую схемотехнику. Особое внимание уделяется основам микропроцессорной техники, без которой невозможно представить современное автоматизированное машиностроительное производство.

Изложение материала построено таким образом, чтобы ранее изложенный материал становился основой для рассмотрения последующего. Сначала дается общее представление о физических основах полупроводниковой электроники и рассматриваются дискретные полупроводниковые приборы. Затем изучаются базовые транзисторные каскады, являющиеся основой для самых сложных аналоговых и цифровых устройств. Далее излагается более сложный материал — аналоговая и цифровая полупроводниковая схемотехника и, в заключение, описывается устройство микропроцессоров и программируемых логических интегральных схем.

Материал учебника разделен на пять глав. Первая глава посвящена основам полупроводниковой схемотехники. В ней рассмотрена работа диодов, биполярных и полевых транзисторов, тиристоров, основные схемы включения транзисторов, а также изложены основы микроэлектроники. Вторая глава освещает особенности аналоговой схемотехники. Здесь дано описание различных усилительных каскадов, аналоговых устройств на основе операционных усилителей, цифроаналоговых и аналого-цифровых устройств, а также схемотехники силовых устройств и вторичных источников электропитания. Третья глава начинается с изложения основ цифровой схемотехники — двоичной арифметики и булевой алгебры — после чего рассматриваются основные логические элементы, комбинационные и последовательностные устройства. Четвертая, самая большая по объему, глава посвящена вопросам схемотехники программируемых цифровых устройств, речь идет о полупроводниковых запоминающих устройствах, микропроцессорах и их программировании, интерфейсных устройствах. В конце четвертой главы рассмотрены различные направ-

ления программируемой матричной логики, включая программируемые логические интегральные схемы, а также средства автоматизированного проектирования электронных устройств. В пятой главе рассматривается архитектура современных микропроцессоров. Этот материал важен для правильного понимания принципов работы современных систем, основанных на микропроцессорах.

В конце каждой главы имеются контрольные вопросы и задания, которые помогут студентам проверить усвоение материала.

В приложениях 1, 2, 3 приведены начальные сведения об электрических принципиальных схемах, в том числе условные обозначения дискретных и интегральных элементов, правила обозначения соединений элементов, сведения о номинальных значениях и допустимых отклонениях резисторов и конденсаторов. Этот материал будет полезен тем, кто по роду своей деятельности должен уметь «читать» принципиальные электрические схемы, разбираться в работе электронных устройств на уровне его схемы.

В учебнике приведен предметный указатель.

При изложении материала большое внимание уделено описанию физики происходящих процессов, хотя бы и в ущерб формульному описанию работы устройств. Приведены простые практические примеры, позволяющие глубже вникнуть в изучаемый вопрос.

Глава 1
ПОЛУПРОВОДНИКОВЫЕ ЭЛЕМЕНТЫ И ОСНОВЫ МИКРОЭЛЕКТРОНИКИ

В результате изучения главы 1 студент должен:

знать

- физику процессов, происходящих в полупроводниковых материалах;
- основные элементы полупроводниковой электроники и их характеристики;
- основные схемы включения транзисторов;
- основы технологии изготовления микросхем;

уметь

- разбираться в работе основных полупроводниковых приборов;
- выбирать схему включения транзистора в зависимости от требований к электрической схеме;

владеть

- методикой расчета схем включения транзисторов.

1.1. Физические основы полупроводниковой электроники

1.1.1. Виды электронных приборов

Электронными называют приборы, в которых ток создается движением электронов в вакууме, газе или полупроводнике.

В своем развитии электроника прошла несколько этапов. Первые электронные устройства выполнялись на электровакуумных приборах (так называемых катодных или электронных лампах). С середины XX в. широкое применение нашли полупроводниковые приборы (транзисторы, диоды, тиристоры), изготовляемые как отдельные, самостоятельные элементы, из которых собирались электронные устройства. В последней четверти XX в. основой

многих электронных устройств стали интегральные микросхемы, представляющие собой пластинку полупроводника с размещенным на ней множеством транзисторов и других элементов электрических цепей. Со времени их изобретения (США, 1959 г.) интегральные микросхемы постоянно совершенствуются и усложняются. В современных сверхбольших интегральных схемах счет идет уже на десятки и сотни миллионов транзисторов и других элементов.

В настоящее время для решения тех или иных задач (преобразование вида энергии, усиление сигналов, генерирование мощных излучений, управление электродвигателями, обработка цифровой информации и ее отображение и т.п.) используются все виды электронных приборов, но явное преимущество сохраняется за полупроводниковыми приборами и микросхемами.

1.1.2. Электрофизические свойства полупроводников

Все вещества образованы атомами, состоящими из положительно заряженных ядер и вращающихся вокруг них отрицательно заряженных электронов. Ядро включает электрически нейтральные частицы — нейтроны и положительно заряженные протоны. Количество протонов определяет заряд ядра. Отрицательный заряд электрона по величине равен положительному заряду протона. В нормальном состоянии число электронов, образующих электронную оболочку атома, равно числу протонов в ядре, и атом электрически нейтрален. Электроны вращаются вокруг ядра по орбитам, сгруппированным в слои. Каждому слою соответствует строго определенная энергия электрона W (так называемый разрешенный энергетический уровень). Количество электронов в слоях строго определено: в первом, ближайшем к ядру слое может находиться не более двух электронов, во втором — не более восьми и т.д. Электроны целиком заполненных слоев устойчивы к внешним воздействиям. «Не уместившиеся» во внутренних слоях электроны образуют незаполненный внешний слой, который легко отдает и принимает электроны. Эти электроны определяют валентность элемента при химических реакциях. Чем дальше от ядра расположена орбита электрона, тем большей энергией он обладает. Под воздействием энергии теплоты, света, радиации или каких-либо других внешних факторов электрон из валентной зоны может перейти на новую, более удаленную от ядра орбиту. Такой электрон

называется возбужденным, а при дальнейшем увеличении энергии, называемой работой выхода, электрон покидает поверхность вещества.

В кристалле происходит взаимодействие между соседними атомами, заключающееся в том, что на электроны атома воздействуют ядра соседних атомов. В результате разрешенные энергетические уровни электронов смещаются и расщепляются на несколько — по числу соседних атомов в кристаллической решетке. Эти уровни создают энергетические зоны. Совокупность энергетических уровней, соответствующих внешнему слою электронов, образует *валентную зону*. Разрешенные уровни энергии, которые остаются незанятыми, составляют *зону проводимости*, так как ее уровни могут занимать возбужденные электроны, обеспечивающие электропроводность вещества. Между валентной зоной и зоной проводимости может располагаться *запрещенная зона*.

Зонная структура лежит в основе разделения веществ на проводники, полупроводники и диэлектрики. На рис. 1.1 показано расположение энергетических зон для этих групп веществ. У проводников (металлов) валентная зона *1* и зона проводимости *2* перекрывают друг друга (рис. 1.1, *а*) и валентные электроны легко переходят в зону проводимости. У диэлектриков (рис. 1.1, *б*) ширина запрещенной зоны велика (более 6 эВ (электрон-вольт)), и для перехода валентных электронов в зону проводимости надо сообщить значительную энергию (такой процесс происходит при пробое изоляции). У полупроводников (рис. 1.1, *в*) запрещенная зона относительно мала и колеблется от 0,1 до 3,0 эВ.

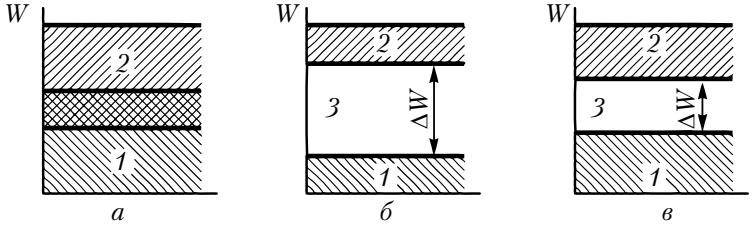


Рис. 1.1. **Энергетические зоны проводника (*а*), диэлектрика (*б*) и полупроводника (*в*):**

1 — валентная зона; *2* — зона проводимости; *3* — запрещенная зона

В кристаллической решетке четырехвалентного полупроводника (например, кремния) каждый атом связан с четырьмя соседними атомами с помощью двух валентных электронов — по одному от каждого атома. Такая связь называется *ковалентной*. При ее образовании электрон принадлежит уже не одному, а обоим связанным между собой атомам, т.е. является для них общим. В результате вокруг каждого ядра образуется восьмиэлектронная оболочка, устойчивая к внешним воздействиям. Так как все валентные электроны оказываются прочно связанными между собой, свободных электронов, способных обеспечить электропроводность, нет. Такую структуру имеют химически чистые полупроводники при температуре абсолютный нуль (рис. 1.2, *а*).

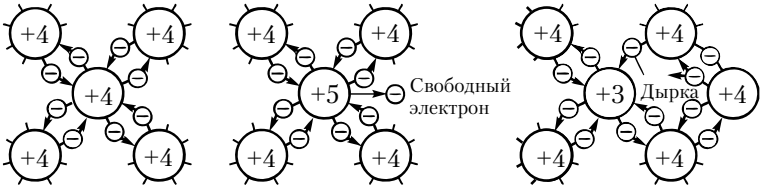


Рис. 1.2. **Кристаллическая структура чистого полупроводника (*а*), полупроводника с донорной (*б*) и акцепторной (*в*) примесью**

Под воздействием внешних факторов (например, при повышении температуры) отдельные электроны атомов кристаллической решетки приобретают энергию, достаточную для освобождения от ковалентных связей, и становятся свободными.

При освобождении электрона от ковалентной связи в кристаллической решетке возникает как бы свободное место, обладающее положительным зарядом. Такое место называется *дыркой*, а процесс образования пары «свободный электрон — дырка» — генерацией. В дырку может «перескочить» валентный электрон из ковалентной связи соседнего атома. В результате ковалентная связь в одном атоме восстановится (этот процесс называется рекомбинацией), а в соседнем разрушится, образуя новую дырку. Такое перемещение дырки по кристаллу равносильно перемещению положительного заряда.

При отсутствии внешнего электрического поля дырки перемещаются хаотически. Если же приложить к кристаллу разность потенциалов, то под действием созданного электрического поля движение дырок и электронов становится упорядоченным, и в кристалле возникает электрический ток. Таким образом, проводимость полупроводника обусловлена перемещением как отрицательно заряженных электронов, так и положительно заряженных дырок. Соответственно различают два типа проводимости — *электронную*, или *проводимость n-типа*, и *дырочную*, или *проводимость p-типа*.

Для создания полупроводниковых элементов широко применяют *примесные полупроводники*. С четырехвалентным германием и кремнием используют пятивалентные (мышьяк, сурьму, фосфор) и трехвалентные примеси (бор, алюминий, индий, галлий).

В случае пятивалентной примеси (рис. 1.2, *б*) четыре валентных электрона примесного атома совместно с четырьмя электронами соседних атомов основного вещества образуют ковалентные связи, а пятый валентный электрон оказывается «лишним». В результате даже при комнатной температуре «лишние» электроны легко освобождаются от своих атомов, переходя в зону проводимости. В таких полупроводниках электропроводность обеспечивается главным образом избытком свободных электронов. Их называют полупроводниками *n*-типа, а примеси — *донорными*.

За счет тепловой энергии в полупроводнике *n*-типа могут образовываться и отдельные дырки при генерации пар «свободный электрон — дырка». Поэтому электроны в полупроводнике *n*-типа называют основными, а дырки — неосновными носителями зарядов.

При введении трехвалентной примеси (рис. 1.2, *в*) в одной из ковалентных связей примесного атома и атома основного полупроводника отсутствует электрон, т.е. образуется дырка. Разрешенные энергетические уровни валентных зон примеси и основного полупроводника находятся рядом. Электрон валентной зоны атома основного полупроводника легко захватывается трехвалентным атомом примеси, в результате чего дырка образуется уже в атоме основного полупроводника и, таким образом, перемещается по кристаллической решетке. Дырки в таких полупроводниках ста-

новятся основными носителями зарядов, создавая эффект перемещения положительных зарядов. Трехвалентные примеси называют *акцепторными*, а полупроводники с такой примесью — полупроводниками *p*-типа. Неосновными носителями в этом случае выступает небольшое количество свободных электронов, образовавшихся в результате тепловой генерации пар «свободный электрон — дырка».

1.1.3. *P-n*-переход и его свойства

Тонкий слой полупроводника между двумя областями, одна из которых представляет полупроводник *p*-типа, а другая *n*-типа, называют *p-n*-переходом. Концентрации основных носителей заряда в *p*- и *n*-областях могут быть равны или существенно различаться. В первом случае *p-n*-переход называют симметричным, во втором — несимметричным. Чаще используют несимметричные переходы.

Пусть концентрация акцепторной примеси в *p*-области больше, чем концентрация донорной примеси в *n*-области (рис. 1.3, *а*). Соответственно, концентрация дырок (светлые кружки) в *p*-области будет больше, чем концентрация электронов (черные кружки) в *n*-области.

За счет диффузии дырки из *p*-области и электроны из *n*-области стремятся равномерно распределиться по всему объему. Если бы электроны и дырки были нейтральными, то диффузия в конечном итоге привела бы к полному выравниванию их концентраций по всему объему кристалла. Однако этого не происходит. Дырки, переходя из *p*-области в *n*-область, рекомбинируют с частью электронов, принадлежащих атомам донорной примеси. В результате оставшиеся без электронов положительно заряженные ионы донорной примеси образуют приграничный слой с положительным зарядом. В то же время уход этих дырок из *p*-области приводит к тому, что атомы акцепторной примеси, захватившие соседний электрон, образуют нескомпенсированный отрицательный заряд ионов в приграничной области. Аналогично происходит диффузионное перемещение электронов из *n*-области в *p*-область, приводящее к тому же эффекту. В результате на границе, разделяющей *n*-область и *p*-область, образуется узкий, в доли микрона, приграничный слой, одна сторона которого заряжена отрицательно (*p*-область), а другая — положительно (*n*-область).

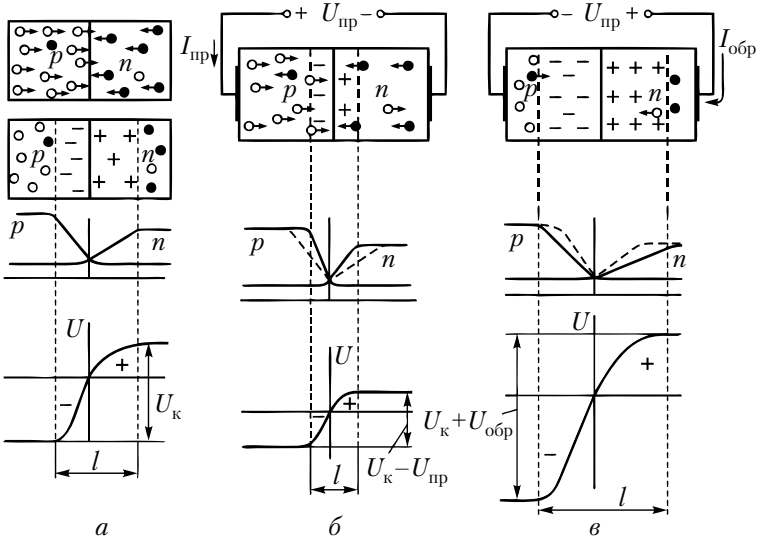


Рис. 1.3. ***P-n*-структура:**

а — в равновесном состоянии; *б* — при прямом внешнем напряжении; *в* — при обратном внешнем напряжении; *l* — ширина *p-n*-перехода

Разность потенциалов, образованную приграничными зарядами, называют *контактной разностью потенциалов* $U_к$ (см. рис. 1.3, *а*), или *потенциальным барьером*, преодолеть который носители не в состоянии. Дырки, подошедшие к границе со стороны *p*-области, отталкиваются назад положительным зарядом, а электроны, подошедшие из *n*-области, — отрицательным зарядом. Таким образом, образуется *p-n*-переход, представляющий собой слой полупроводника с пониженным содержанием носителей — так называемый обедненный слой, который имеет относительно высокое электрическое сопротивление.

Свойства *p-n*-структуры изменяются, если к ней приложить внешнее напряжение. Если внешнее напряжение противоположно по знаку контактной разности потенциалов (рис. 1.3, *б*), то дырки *p*-области, отталкиваясь от приложенного положительного потенциала внешнего источника, приближаются к границе между областями, компенсируют заряд части отрицательных ионов и сужают ширину *p-n*-перехода со стороны *p*-области. Аналогично этому, электроны *n*-области, отталкиваясь от отрицательного потен-

циала внешнего источника, компенсируют заряд части положительных ионов и сужают ширину *p-n*-перехода со стороны *n*-области. Потенциальный барьер сужается, через него начинают проникать дырки из *p*-области и электроны из *n*-области и через *p-n*-переход начинает течь ток.

С увеличением внешнего напряжения ток возрастает неограниченно, так как создается основными носителями, концентрация которых постоянно восполняется источником внешнего напряжения.

Полярность внешнего напряжения, приводящая к снижению потенциального барьера, называется прямой, или открывающей, а созданный ею ток — прямым. При подаче такого напряжения *p-n*-переход открыт.

Если к *p-n*-структуре приложить напряжение обратной полярности (рис. 1.3, *в*), то эффект будет противоположным. Под действием электрического поля источника дырки *p*-области смещаются к отрицательному потенциалу внешнего напряжения, а электроны *n*-области — к положительному потенциалу. Таким образом, основные носители зарядов отодвигаются внешним полем от границы, увеличивая ширину *p-n*-перехода, который оказывается почти свободным от носителей заряда. Электрическое сопротивление *p-n*-перехода при этом возрастает. Такая полярность внешнего напряжения называется обратной, запирающей. При подаче такого напряжения *p-n*-переход закрыт.

Тем не менее при обратном напряжении наблюдается протекание небольшого тока $I_{обр}$. Этот ток, в отличие от прямого, определяется носителями не примесной, а собственной проводимости, образующейся в результате генерации пар «свободный электрон—дырка» под воздействием температуры. Эти носители обозначены на рис. 1.3, *в* единственным электроном в *p*-области и единственной дыркой в *n*-области. Значение обратного тока практически не зависит от внешнего напряжения. Это объясняется тем, что в единицу времени количество генерируемых пар «свободных электрон—дырка» при неизменной температуре остается постоянным, и даже при $U_{обр}$ в доли вольта все носители участвуют в создании обратного тока.

При подаче обратного напряжения *p-n*-переход уподобляется конденсатору, пластинами которого являются *p*- и *n*-области, разделенные диэлектриком. Роль диэлектрика выполняет приграничная область, почти свободная от носителей заряда. Эту емкость *p-n*-перехода называют *барьерной*. Она тем больше, чем меньше ширина *p-n*-перехода и чем больше его площадь.

1.1.4. Основные технологические процессы изготовления *p-n*-переходов

Метод сплавления. Этот технологический процесс заключается в том, что в пластинку полупроводника одного типа проводимости вплавляют примесь, необходимую для образования полупроводника другого типа проводимости. Например, на пластинку германия *n*-типа помещают таблетку индия и нагревают ее до температуры плавления. При этом примесь расплавляется, и в ней частично растворяется материал полупроводника, создавая в приграничной зоне слой проводимости *p*-типа. Сплавные *p-n*-переходы имеют высокую надежность, работоспособны при больших обратных напряжениях и обладают малым собственным сопротивлением, обеспечивающим малое прямое падение напряжения на них.

Метод диффузии. В этом технологическом процессе *p*- и *n*-области получают в полупроводнике путем проникновения акцепторных или донорных примесей, содержащихся в атмосфере паров, куда помещают нагретую до высокой температуры пластинку полупроводника. Так как атомы примеси диффундируют внутрь пластины с поверхности, наибольшая концентрация примеси наблюдается в приповерхностной области и убывает с увеличением расстояния в глубь полупроводника.

Метод эпитаксиального наращивания. Процесс кристаллизации одного вещества на кристалле-подложке другого называется *эпитаксиальным наращиванием*. При этом кристаллическая решетка подложки определяет структуру решетки, в которой кристаллизуется наращиваемый слой. Полупроводниковые эпитаксиальные слои (пленки) могут быть получены различными способами: герметическим испарением в вакууме; кристаллизацией в расплавленном веществе, содержащем примесь; осаждением из парообразной формы. Изменяя тип примеси и условия наращивания, можно в широких пределах менять электрические свойства эпитаксиальной пленки.

Ионное легирование. Процесс заключается в бомбардировке ионами примеси нагретой полупроводниковой пластины, находящейся в вакууме. Ионы предварительно разгоняются до определенной скорости и, внедряясь в пластину полупроводника, играют роль донорных или акцепторных примесей.

Оксидное маскирование. Этот процесс используют для того, чтобы обеспечить проникновение примеси только в определенные участки пластины, защитив от них остальную ее поверхность. В полупроводниковых структурах на основе кремния в качестве маски используется диоксид кремния SiO_2, который является хорошим изолятором и обладает по сравнению с чистым кремнием значительно меньшей скоростью диффузии в него примесей. Для получения пленки оксида кремниевую пластину нагревают до 900—1200 °С в атмосфере кислорода. После охлаждения те участки полупроводника, которые должны подвергаться воздействию примесей, освобождают от пленки окисла травлением.

Фотолитография. Это процесс получения на поверхности пленки оксида необходимого рисунка расположения окон. Оксидную пленку покрывают фоторезистом (светочувствительным слоем) и экспонируют (засвечивают) ультрафиолетовыми лучами через маску, на которой выполнен рисунок в виде прозрачных и непрозрачных участков. Участки фоторезиста, подвергнувшиеся освещению, оказываются задубленными (нерастворимыми), а с неосвещенных участков фоторезист удаляют растворителем.

Травление пленки диоксида кремния с участков, не защищенных задубленным фоторезистом, производят плавиковой кислотой, в результате в оксидной пленке образуются окна, через которые производится диффузия, эпитаксиальное наращивание или ионное легирование.

1.2. Полупроводниковые диоды

Полупроводниковым диодом называют полупроводниковый прибор с двумя выводами (приставка «ди-» означает два) и одним *p-n*-переходом.

По функциональному назначению, принципу образования *p-n*-перехода и использованию тех или иных его свойств диоды делятся на выпрямительные, стабилитроны, импульсные, диоды Шоттки, фотодиоды, светодиоды и т.п. Один из вариантов конструкции и условное обозначение диода приведены на рис. 1.4, *а, б*. Вывод от *p*-области называют *анодом*, а от *n*-области — *катодом*.

Вольт-амперная характеристика полупроводникового диода, т.е. зависимость протекающего через диод тока от приложенного к нему напряжения, определяется вольт-амперной характеристикой p-n-перехода (рис. 1.4, *в*). При подаче к диоду прямого напряжения диод открыт и пропускает прямой ток, при этом падение напряжения на диоде составляет десятые доли вольт. При подаче обратного напряжения диод заперт, и если оно не достигает значения $U_{\text{обр.max}}$, то через диод протекает пренебрежительно малый обратный ток I_0 ($I_{\text{пр}}$ и $I_{\text{обр}}$ на рис. 1.4, *в* показаны в разных масштабах). При подаче обратного напряжения, превышающего значение $U_{\text{обр.max}}$, наступает пробой p-n-перехода, при котором обратный ток резко возрастает.

Различают два вида пробоя: электрический (обратимый) и тепловой (необратимый, выводящий полупроводниковый диод из строя). Под воздействием сильного электрического поля электроны освобождаются от ковалентных связей и получают энергию, достаточную для преодоления высокого потенциального барьера. Двигаясь с большой скоростью в p-n-переходе, электроны сталкиваются с нейтральными атомами и ионизируют их, в результате чего появляются новые свободные электроны и дырки. Этот

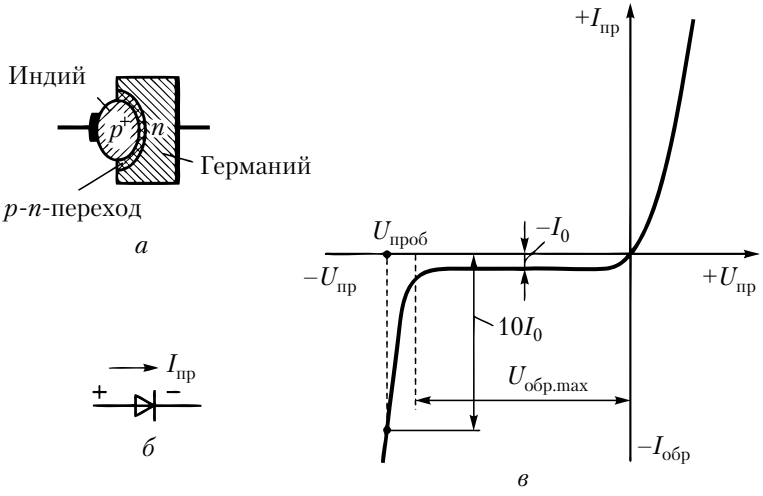


Рис. 1.4. **Полупроводниковый диод:**
а — вариант конструкции; *б* — условное обозначение; *в* – вольт-амперная характеристика

лавинообразный процесс приводит к резкому увеличению обратного тока, т.е. к электрическому пробою p-n-перехода. Если обратный ток ограничивать, то после снятия высокого обратного напряжения p-n-переход восстанавливается. Если же обратный ток не ограничивать, то электрический пробой перейдет в тепловой, при котором за счет разогрева p-n-перехода происходит энергичная генерация пар «свободный электрон—дырка», приводящая к резкому увеличению обратного тока. Увеличение тока приводит к еще большему разогреву p-n-перехода и, значит, к дальнейшей генерации носителей. Процесс нарастает лавинообразно и приводит к выводу диода из строя.

1.2.1. Выпрямительные диоды

Эти диоды предназначены для преобразования (выпрямления) переменного тока в постоянный. К их быстродействию, емкости p-n-перехода и стабильности параметров высоких требований не предъявляют. Их выполняют на сплавных и диффузионных несимметричных p-n-переходах. Выпрямительные диоды характеризуются малым сопротивлением в прямом направлении и позволяют пропускать большие токи (до десятков и сотен ампер) при допустимых обратных напряжениях до 1000 В. Для этого площадь p-n-перехода выполняется относительно большой и, следовательно, емкость p-n-перехода достаточно велика (десятки пикофарад). Поэтому переходные процессы в этих диодах протекают относительно долго. (Под длительностью переходного процесса понимают время перехода из открытого состояния диода в запертое и наоборот при перемене полярности приложенного напряжения.)

Основными параметрами выпрямительных диодов являются:

- допустимое обратное напряжение $U_{обр}$, которое диод может выдержать в течение длительного времени без нарушения работоспособности;
- средний прямой ток $I_{пр.ср}$ — наибольшее допустимое значение постоянного тока, протекающего длительно в прямом направлении;
- максимально допустимый импульсный прямой ток $I_{пр}$ при указанной в паспорте наибольшей длительности импульса;
- средний обратный ток $I_{обр.ср}$ — среднее за период значение обратного тока;

• среднее прямое напряжение $U_{\text{пр.ср}}$ — падение напряжения на открытом диоде;

• средняя рассеиваемая мощность $P_{\text{ср.д}}$ — средняя за период мощность, выделяющаяся в диоде при выпрямлении переменного тока;

• дифференциальное сопротивление $r_{\text{диф}} = \Delta U_{\text{пр.ср}} / \Delta I_{\text{пр.ср}}$.

1.2.2. Стабилитроны

Стабилитроны — это диоды, использующие участок вольт-амперной характеристики p-n-перехода, соответствующий обратному электрическому пробою (рис. 1.5, *а*). Стабилитрону, как показывает само название, свойственна стабильность, т.е. неизменность падения напряжения на нем при изменениях в несколько раз тока, протекающего через него. Благодаря этому свойству, стабилитроны широко применяются в качестве источников опорного напряжения, которое должно оставаться неизменным при каких-либо изменениях других параметров схемы. Применяются они и как стабилизаторы напряжения при небольших мощностях нагрузки.

Во избежание теплового пробоя последовательно со стабилитроном включают резистор R_0 (рис. 1.5, *б*), ограничивающий ток $I_{\text{ст}}$, который является обратным током для p-n-структуры стабилитрона. При изменениях входного напряжения $\Delta U_{\text{вх}}$ (рис. 1.5, *в*) меняются ток $\Delta I_{\text{ст}}$ и падение напряжения от этого тока на R_0. Значения тока $I_{\text{ст}}$ и его изменений определяются точками A, B и C пересечения вольт-амперной характеристики p-n-перехода и прямых, проведенных под углом $\text{arctg}R_0$ из точек $U_{\text{вх}}$ и его изменений, отложенных на оси $U_{\text{обр}}$. Точка A определяет значение $U_{\text{ст}}$ при среднем значении $U_{\text{вх}}$, а точки B и C — изменения $\Delta U_{\text{ст}}$ при изменениях $\Delta U_{\text{вх}}$.

Дифференциальное сопротивление стабилитрона на участке BC равно

$$r_{\text{диф}} = \Delta U_{\text{ст}} / \Delta I_{\text{ст}}. \qquad (1.1)$$

Если напряжение $U_{\text{вх}}$ может изменяться в обе стороны от своего среднего значения, то точку A выбирают на середине линейного участка вольт-амперной характеристики стабилитрона, причем

$$U_{\text{вх}} = U_{\text{ст}} + I_{\text{ст}} \cdot R_0. \qquad (1.2)$$

Перейдя от (1.2) к приращениям, запишем:

$$\Delta U_{вх} = \Delta U_{ст} + \Delta I_{ст} \cdot R_0,$$

а подставив $\Delta I_{ст}$ из (1.1), получим:

$$\Delta U_{вх} = \Delta U_{ст} + \Delta U_{ст} \cdot R_0 / r_{диф},$$

откуда

$$\Delta U_{ст} = \frac{\Delta U_{вх}}{1 + \dfrac{R_0}{r_{диф}}}.$$

При $R_0 >> r_{диф}$ получим, что $\Delta U_{ст} << \Delta U_{вх}$ и стабилизация тем лучше, чем больше отношение $R_0 / r_{диф}$.

Основными параметрами стабилитронов являются:

- напряжение стабилизации $U_{ст}$;
- минимальный ток стабилизации $I_{ст.min}$, при котором наступает устойчивый электрический пробой p-n-перехода;
- максимальный ток стабилизации $I_{ст.max}$, при котором мощность, рассеиваемая на стабилитроне, не превышает допустимого значения;
- дифференциальное сопротивление $r_{диф}$;



Рис. 1.5. **Стабилитрон:**
а — условное обозначение; *б* — схема включения; *в* — вольт-амперная характеристика

Содержание

• максимальная мощность рассеяния P_{max}, при которой еще не наступает тепловой пробой p-n-перехода;

• температурный коэффициент стабилизации $\alpha_{ст}$ — отношение относительного изменения напряжения стабилизации к абсолютному изменению температуры окружающей среды (выражается в %/град):

$$\alpha_{ст} = \Delta U_{ст} / (U_{ст} \cdot \Delta T).$$

Выпускаются кремниевые стабилитроны на напряжение стабилизации от 5 до 400 В и на мощность от 250 мВт до 50 Вт.

1.2.3. Диоды Шоттки

В основе выпрямляющего диода может использоваться не только переход между полупроводниками p- и n-типа, но и между полупроводником и металлом. Такие диоды называются *диодами Шоттки*.

Рассмотрим структуру металл—полупроводник n-типа. Если работа выхода электронов у металла выше, чем у полупроводника, то преобладающим будет перемещение электронов из полупроводника в металл (свободным электронам металла труднее приобрести энергию, равную работе выхода, чем электронам полупроводника). В результате металл заряжается отрицательно, а оставшиеся в полупроводнике ионы донорной примеси создают в его приграничном слое положительный потенциал (рис. 1.6, *а*). Такое распределение зарядов создает контактную разность

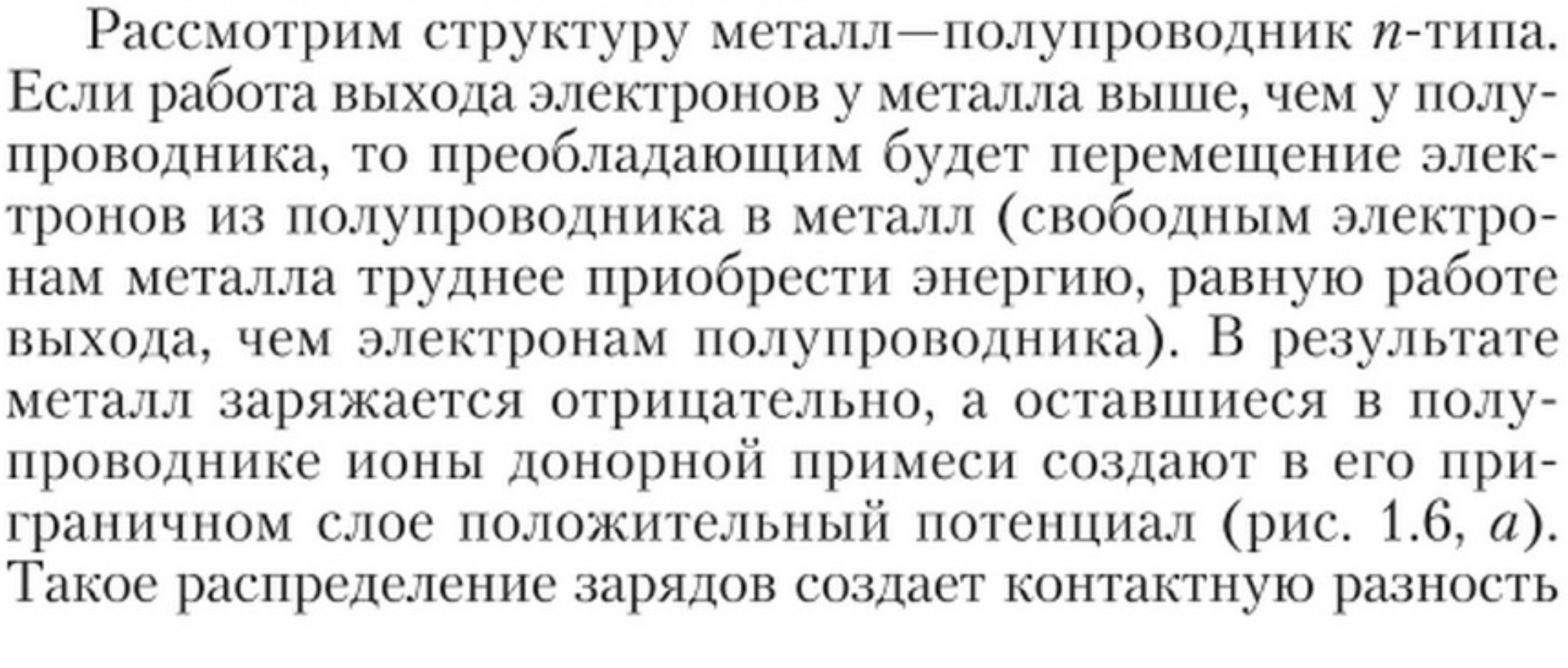


Рис. 1.6. **Металлополупроводниковый диод Шоттки:**
а — структура диода; *б* — условное обозначение; *в* — вольт-амперная характеристика;
1 — переход Шоттки; *2* — p-n-переход

потенциалов $U_к$ (потенциальный барьер), препятствующий дальнейшему перемещению электронов. При этом тонкий приграничный слой полупроводника обедняется носителями. Таким образом, в месте контакта металла и полупроводника возникает переход, аналогичный p-n-переходу. Если к такому переходу приложить обратное напряжение, совпадающее с $U_к$, то ширина обедненной области увеличится, а сопротивление перехода возрастет. Если приложить прямое напряжение, то оно будет противодействовать $U_к$, при этом переход сужается, потенциальный барьер уменьшается и через переход начинает течь ток. Вольт-амперные характеристики такого перехода и p-n-перехода оказываются аналогичными.

Основной отличительной особенностью характеристик диода Шоттки является значительно меньшее прямое падение напряжения по сравнению с диодами на основе p-n-перехода (рис. 1.6, *в*). Это объясняется тем, что в диоде Шоттки одно из веществ перехода — металл, и следовательно, его электрическое сопротивление (и соответствующее падение напряжения на нем) значительно меньше, чем у полупроводника.

Другая особенность диода Шоттки — отсутствие проникновения неосновных носителей заряда из металла в полупроводник (в рассматриваемом случае — дырок, которые для n-области являются неосновными). Это значительно повышает быстродействие диодов Шоттки по сравнению с обычными диодами, так как отпадает необходимость в рассасывании таких носителей при смене полярности внешнего напряжения.

Диоды Шоттки, у которых выпрямляющий переход представляет собой тонкую пленку молибдена или алюминия, нанесенную на пластинку кремния методом вакуумного напыления, обладают емкостью, не превышающей 0,01 пФ. Это обеспечивает чрезвычайно малое время их переключения (доли наносекунды) и предельно высокую частоту работы (десятки гигагерц). Мощные диоды позволяют пропускать токи в десятки ампер при обратных напряжениях до 500 В. Благодаря меньшему прямому напряжению (0,3 В вместо 0,7 В у диодов p-n-типа) они обеспечивают более высокий КПД. Условное обозначение металлополупроводникового диода Шоттки приведено на рис. 1.6, *б*.

1.2.4. Варикапы

Напомним, что при подаче обратного напряжения *p-n*-структура уподобляется конденсатору, пластинами которого являются *p*- и *n*-области, разделенные диэлектриком (переходом, почти свободным от носителей заряда). Образующаяся при этом барьерная емкость может быть использована в качестве конденсатора в электронной аппаратуре. *Варикапы* — это полупроводниковые диоды, работа которых основана на явлении барьерной емкости запертого *p-n*-перехода. Поскольку размеры области *p-n*-перехода зависят от значения приложенного к нему обратного напряжения, то и величина барьерной емкости изменяется вместе с этим напряжением.

Внешнее обратное напряжение, втягивая электроны вглубь *n*-области, а дырки — вглубь *p*-области, расширяет *p-n*-переход и изменяет барьерную емкость. Основной характеристикой варикапа является зависимость его емкости от значения обратного напряжения — вольт-фарадная характеристика. Основными параметрами варикапов являются номинальная емкость и диапазон ее изменения, а также допустимые обратное напряжение и мощность. Варикапы применяются для электрической настройки колебательных контуров в радиоаппаратуре.

1.2.5. Светодиоды

На основе явлений, происходящих в *p-n*-переходе при протекании через него прямого тока, можно получать полупроводниковые приборы, способные генерировать оптическое излучение. Такими приборами являются *полупроводниковые светодиоды*. Работа светодиодов основана на *инжекционной электролюминесценции*, т.е. генерации оптического излучения в *p-n*-переходе, находящемся под прямым внешним напряжением. Под воздействием внешней энергии электроны в атомах переходят в возбужденное состояние с более высоким уровнем энергии W_2, называемым метастабильным уровнем возбуждения. При возвращении этих электронов с метастабильного уровня W_2 на исходный W_1 происходит испускание фотонов с длиной волны, определяемой соотношением:

$$\lambda = 1{,}23\,(W_2 - W_1).$$

К преимуществам полупроводниковых светодиодов относятся высокий по сравнению с лампами накалива-

ния КПД, относительно узкий спектр излучения и хорошая диаграмма направленности, высокое быстродействие и малое напряжение питания. Все это обеспечивает удобство согласования с интегральными микросхемами, высокую надежность, долговечность и технологичность. Спектр излучения, а следовательно и его цвет, зависит от используемого полупроводникового материала. Светодиоды изготавливают не на основе кремния или германия, как большинство полупроводниковых приборов, а на основе арсенида-фосфида галлия. Яркость свечения пропорциональна прямому току светодиода. Тока в несколько миллиампер уже достаточно для отчетливой индикации. Светодиоды изготавливают как в виде отдельных индикаторов, так и в виде семисегментных или точечных матриц. Семисегментные матрицы состоят из семи светящихся полосок — сегментов, из которых можно синтезировать изображение любой цифры от 0 до 9 (такие матрицы используются, например, в электронных часах с цифровой индикацией). В точечных матрицах изображение формируется из светящихся точек. На основе точечных матриц можно синтезировать изображение уже не только цифры, но и любого индицируемого знака (буквы, специального символа и т.д.).

1.2.6. Фотодиоды

Простейший *фотодиод* представляет собой обычный полупроводниковый диод (см. рис. 1.4, *а*), в котором обеспечивается возможность воздействия оптического излучения на *p*-*n*-переход. В равновесном состоянии, когда поток излучения полностью отсутствует, концентрация носителей, распределение потенциала и энергетическая зонная диаграмма фотодиода полностью соответствуют обычной *p*-*n*-структуре.

При воздействии излучения в направлении, перпендикулярном плоскости *p*-*n*-перехода, в результате поглощения фотонов с энергией, большей, чем ширина запрещенной зоны, в *n*-области возникают электронно-дырочные пары. Эти электроны и дырки называют фотоносителями. При диффузии фотоносителей вглубь *n*-области основная доля электронов и дырок не успевает рекомбинировать и доходит до границы *p*-*n*-перехода. Здесь фотоносители разделяются электрическим полем *p*-*n*-перехода, причем дырки переходят в *p*-область, а электроны не могут преодо-

леть поле перехода и скапливаются у границы *p*-*n*-перехода и *n*-области.

Таким образом, ток через *p*-*n*-переход обусловлен дрейфом неосновных носителей — дырок. Дрейфовый ток фотоносителей называется *фототоком* $I_ф$. Фотоносители — дырки — заряжают *p*-область положительно относительно *n*-области, а фотоносители — электроны — заряжают *n*-область отрицательно по отношению к *p*-области. Возникающая разность потенциалов называется фотоЭДС — $E_ф$. Генерируемый ток в фотодиоде — обратный, он направлен от катода к аноду. Причем его величина тем больше, чем больше освещенность.

Фотодиоды могут использоваться для получения электрической энергии. Так, солнечные батареи изготавливают на основе фотодиодов с большой площадью *p*-*n*-перехода.

1.2.7. Оптроны

Светодиоды и фотодиоды часто используются в паре. При этом они помещаются в один корпус таким образом, чтобы светочувствительная площадка фотодиода располагалась напротив излучающей площадки светодиода. Полупроводниковые приборы, использующие пары «светодиод—фотодиод», называются оптронами (рис. 1.7). Они широко используются в электронной аппаратуре для гальванической развязки входных и выходных цепей.

Входные и выходные цепи в таких приборах оказываются электрически никак не связанными, поскольку передача сигнала осуществляется через оптическое излучение.

Использование оптронов в электронно-вычислительных устройствах является одним из основных методов повышения помехоустойчивости аппаратуры.

Основной носитель помех в радиоэлектронной аппаратуре — корпус. Корпус используется как один из полюсов электропитания, поэтому подключение к нему разных силовых устройств приводит к наведению кратковременных импульсных помех при коммутациях сильноточных цепей. В то же время для передачи информации чисто электрическим путем между устройствами — источником и приемником информации — должна быть электрическая связь по корпусу. Если к этому же

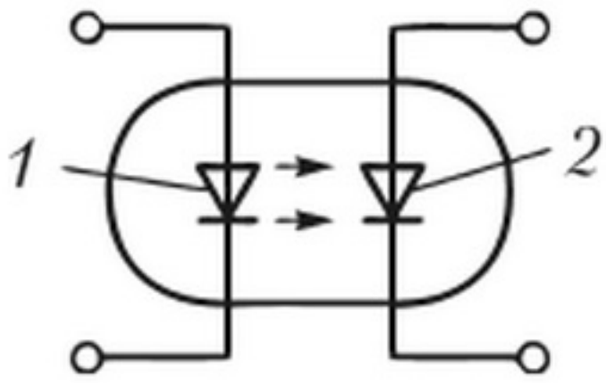


Рис. 1.7. **Оптрон:**
1 — светодиод;
2 — фотодиод

корпусу подключены силовые цепи, то помехи, вызванные коммутациями в этих цепях, приводят к сбоям в работе других устройств, подключенных к корпусу.

Передача информации с помощью оптронов позволяет развязать электрические цепи питания источника и приемника информации, так как носителем информации является электрически нейтральное оптическое излучение. Таким образом, устройства могут иметь разные корпуса, т.е. оказываются гальванически развязанными и не подверженными воздействию помех.

Кроме защиты от воздействия помех, гальваническая развязка на основе оптронов позволяет решить еще одну задачу — совместную работу устройств, находящихся под разными потенциалами. Любая, даже небольшая, разность потенциалов не позволяет чисто электрически соединять разные устройства, поскольку это приведет к выходу их из строя. Передача сигнала в оптроне возможна, даже если цепи светодиода и фотодиода находятся под разными (в некоторых оптронах до 500 В) напряжениями. Таким образом, устройства, информационно связанные с помощью оптронов, могут находиться под разными потенциалами.

1.3. Биполярные транзисторы

Биполярный транзистор — это полупроводниковый прибор, состоящий из трех чередующихся областей полупроводника с различным типом проводимости (*p-n-p* или *n-p-n*) с выводом от каждой области.

На рис. 1.8, *а* показан *n-p-n*-транзистор, а на рис. 1.9, *а* — *p-n-p*-транзистор. Области и выводы от них называются: эмиттер Э (англ. *emit* — испускать, извергать), база Б и коллектор К (англ. *collect* — собирать).

Рассмотрим работу транзистора *n-p-n*-типа. Чередующиеся области образуют два *p-n*-перехода база—эмиттер (БЭ) и база—коллектор (БК). К переходу БЭ прикладывают прямое напряжение $E_{БЭ}$, под действием которого электроны *n*-области эмиттера устремляются в базу, создавая ток эмиттера. Концентрацию примесей в эмиттере делают во много раз больше, чем в базе, а саму базу по возможности тоньше. Поэтому лишь незначительная часть (1—5%) испущенных эмиттером электронов рекомбинирует с дыр-

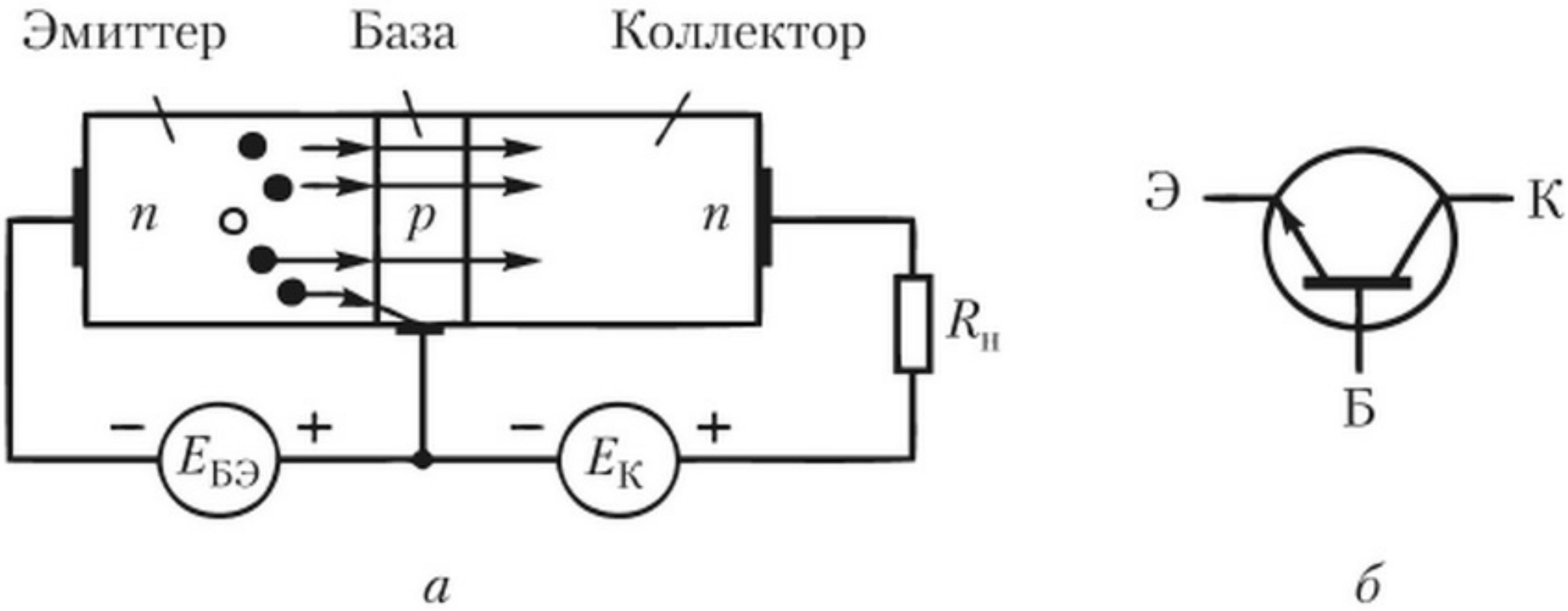


Рис. 1.8. **Биполярный транзистор *n-p-n*-типа:**
а — принцип действия; *б* — условное обозначение

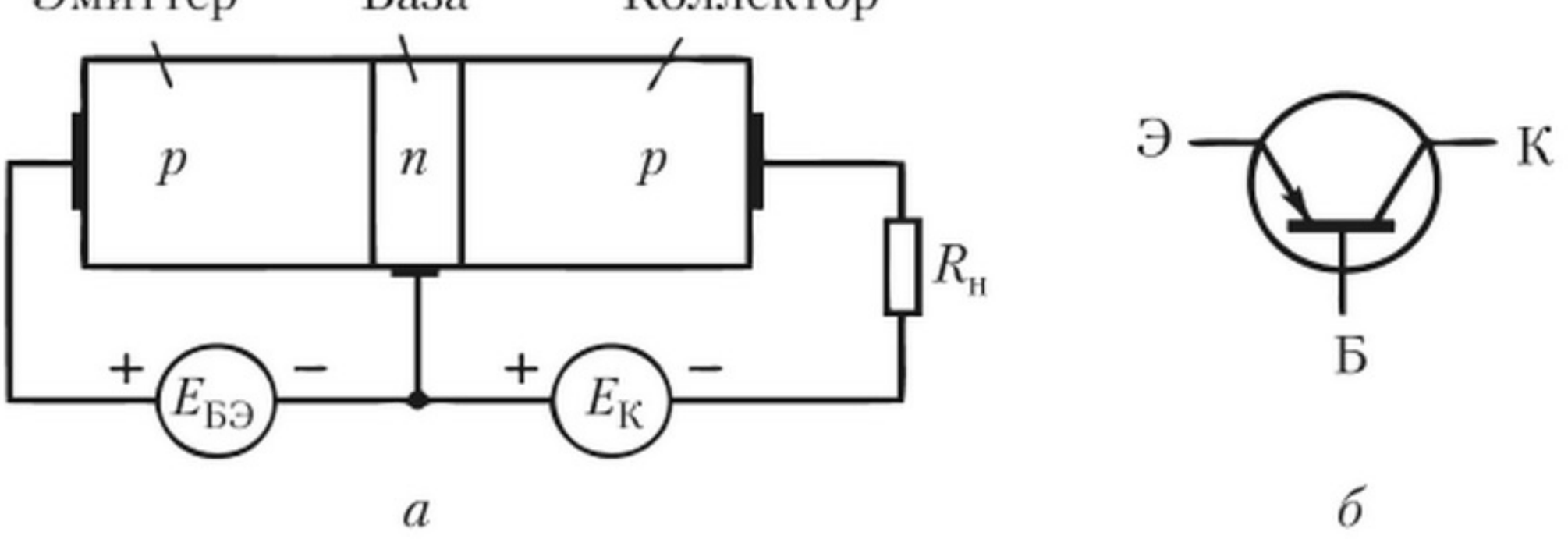


Рис. 1.9. **Биполярный транзистор *p-n-p*-типа:**
а — принцип действия; *б* — условное обозначение

ками базы. Бóльшая же часть электронов, миновав узкую (доли микрона) область базы, «собирается» коллекторным напряжением $E_К$, представляющим обратное напряжение для перехода БК, и, устремляясь к плюсу внешнего источника $E_К$, создает коллекторный ток, протекающий по нагрузке $R_н$.

Электроны, рекомбинировавшие с дырками базы, составляют ток базы $I_Б$. Ток коллектора, таким образом, определяется током эмиттера за вычетом тока базы:

$$I_К = I_Э - I_Б = \alpha I_Э,$$

где $\alpha = 0{,}95 \div 0{,}98$ — *коэффициент передачи тока эмиттера.*

Аналогично работает транзистор *p-n-p*-типа, отличаясь лишь тем, что его эмиттер испускает в базу не электроны, а дырки, поэтому полярности прикладываемых к нему прямого $U_{ЭБ}$ и обратного $E_К$ напряжений должны быть проти-

воположны транзистору *n-p-n*-типа (сравните рис. 1.8, *а* и рис. 1.9, *а*).

На условном обозначении транзисторов стрелка ставится на эмиттере и направлена всегда от *p*-области к *n*-области. На рис. 1.8, *б* приведено условное обозначение транзистора *n-p-n*, а на рис. 1.9, *б* — *p-n-p*. Кружок вокруг транзистора означает, что транзистор изготовлен в самостоятельном корпусе, а отсутствие кружка — что транзистор выполнен заодно с другими элементами на пластинке полупроводника интегральной микросхемы.

Стрелку эмиттера удобно рассматривать как указатель полярности прямого напряжения, приложенного между базой и эмиттером, которое «открывает» (подобно выпрямительному диоду) транзистор.

При использовании транзистора в электронных устройствах нужны два вывода для входного сигнала и два — для выходного. Так как у транзистора всего лишь три вывода, один из них должен быть общим, принадлежащим одновременно и к входной, и к выходной цепи. Возможны три варианта схем включения транзисторов — с общей базой, общим эмиттером и с общим коллектором.

1.3.1. Схема с общей базой

Схема включения транзистора с общей базой (ОБ) показана на рис. 1.10. Входным сигналом для схемы с ОБ является напряжение, поданное между эмиттером и базой $U_{вх} = U_{ЭБ}$; выходным — напряжение, выделяемое на нагрузке $U_{вых} = I_К R_н$; входным током — ток эмиттера $I_{вх} = I_Э$; выходным током — ток коллектора $I_{вых} = I_К$. Входное напряжение $U_{ЭБ}$ является управляющим для транзистора, поэтому

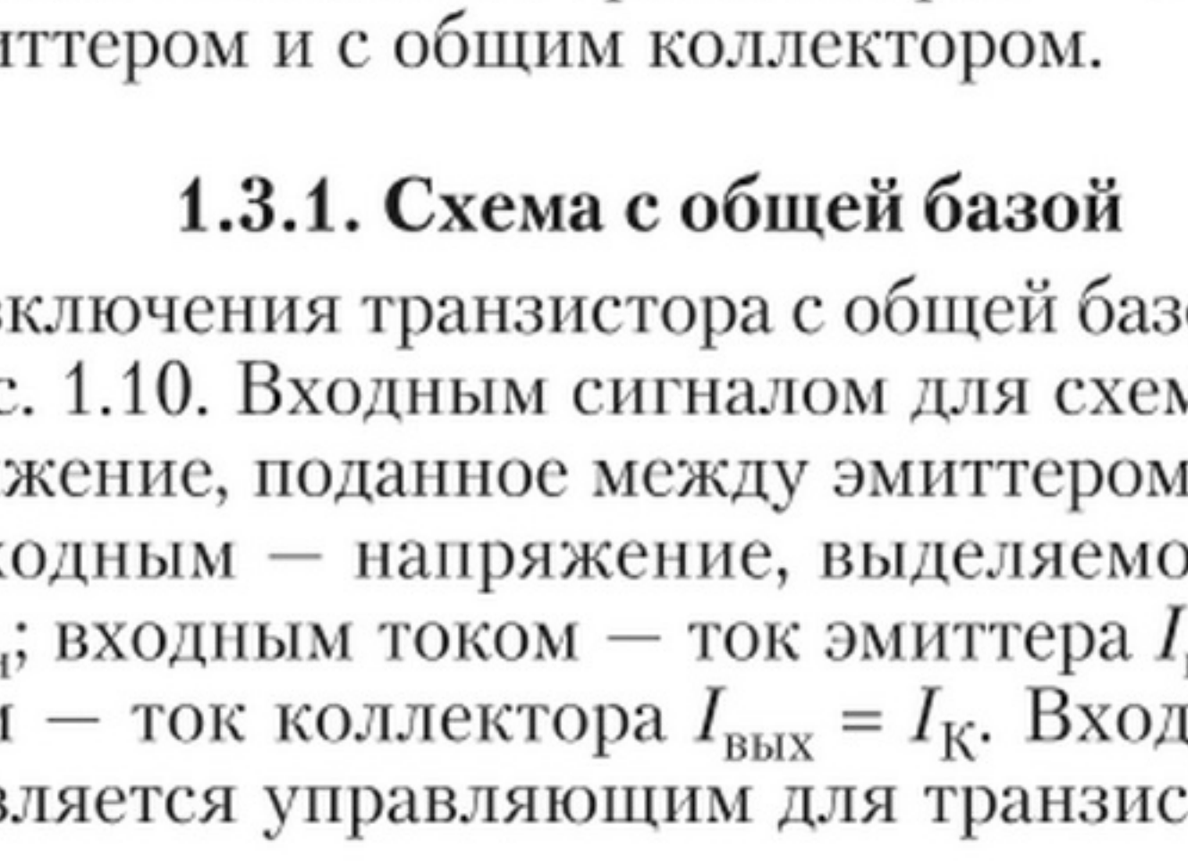


Рис. 1.10. **Схема включения транзистора с общей базой**

небольшое его изменение (на доли вольт) приводит к изменению тока эмиттера в очень широких пределах — практически от нуля до максимального. Максимальный ток определяется назначением транзистора (маломощные, средней мощности и большой мощности) и соответствующей конструкцией. Так как напряжение $U_{КБ}$ является обратным, величина напряжения внешнего источника $E_К$ может в десятки раз превышать значение напряжения $U_{ЭБ}$. Падение напряжения, выделяемого на нагрузке, будет тем больше, чем больше ток коллектора, при этом на самом транзисторе будет падать лишь небольшое напряжение $U_{КБ}$, которое будет тем меньше, чем больше ток коллектора:

$$U_{КБ} = E_К - U_{вых} = E_К - I_К \cdot R_н.$$

Таким образом, изменение на доли вольт входного напряжения приводит к изменению напряжения на нагрузке, чуть меньшего, чем напряжение $E_К$. Это положение определяет усилительные свойства транзистора.

Для оценки работы транзистора и его усилительных свойств в различных схемах включения рассматривают приращения входных и вызванные ими приращения выходных величин. Рассматривая транзистор как усилитель, принято характеризовать его свойства *коэффициентами усиления* и значением *входного сопротивления*. Различают три вида коэффициентов усиления:

- коэффициент усиления по току $K_I = \Delta I_{вых} / \Delta I_{вх}$;
- коэффициент усиления по напряжению $K_U = \Delta U_{вых} / \Delta U_{вх}$;
- коэффициент усиления по мощности $K_P = K_I \cdot K_U$.

Величина входного сопротивления определяется как отношение изменения входного напряжения к изменению входного тока: $R_{вх} = \Delta U_{вх} / \Delta I_{вх}$. Входное сопротивление любого усилителя приводит к искажению входного сигнала. Любой реальный источник сигнала обладает некоторым внутренним сопротивлением, и при подключении его к усилителю образуется делитель напряжения, состоящий из внутреннего сопротивления источника и входного сопротивления усилителя. Поэтому чем выше входное сопротивление усилителя, тем большая часть сигнала будет выделяться на этом сопротивлении и усиливаться и тем меньшая его часть будет падать на внутреннем сопротивлении самого источника.

Содержание

Так как коэффициент передачи тока эмиттера определяется как $\alpha = I_К / I_Э$, то с учетом того, что для схемы с ОБ ток эмиттера является входным током, а ток коллектора — выходным, коэффициент усиления по току будет равен:

$$К_{ІБ} = \Delta I_{вых} / \Delta I_{вх} = \Delta I_К / \Delta I_Э = \alpha = 0{,}95...0{,}99.$$

(Индекс «Б» в обозначении $К_{ІБ}$ показывает, что коэффициент характеризует работу схемы с ОБ.)

Так как входным напряжением является прямое для эмиттерного перехода напряжение $U_{ЭБ}$, а входным током — ток эмиттера, то входное сопротивление определится как

$$R_{вхБ} = \Delta U_{вх} / \Delta I_{вх} = \Delta U_{ЭБ} / \Delta I_Э$$

и составляет обычно единицы—десятки ом. Очевидно, что чем мощнее транзистор, тем больше будет ток эмиттера и тем меньше его входное сопротивление.

Коэффициент усиления по напряжению в схеме с ОБ равен:

$$K_{UБ} = \frac{\Delta U_{вых}}{\Delta U_{вх}} \alpha \frac{\Delta I_{вых} \cdot R_н}{\Delta I_{вх} \cdot R_{вхБ}} \approx \frac{R_н}{R_{вхБ}} \frac{R_н}{R_{вхБ}},$$

т.е. определяется соотношением сопротивления нагрузки и входного сопротивления. Если $R_н$ — килоомы, коэффициент $K_{UБ}$ может достигать 1000.

Коэффициент усиления по мощности определяется как произведение коэффициентов усиления по току и напряжению:

$$K_{PБ} = K_{IБ} K_{UБ} \approx \frac{R_н}{R_{вхБ}} \frac{R_н}{R_{вхБ}}.$$

Таким образом, $K_{PБ}$ тоже определяется соотношением сопротивлений.

Так как коэффициент усиления схемы с ОБ по току $K_{ІБ}$ оказывается меньше единицы, она применения не нашла.

1.3.2. Схема с общим эмиттером

Схема с общим эмиттером (ОЭ) представлена на рис. 1.11. Транзистор *n-p-n* в этой схеме работает так же, как и в схеме с ОБ. Заметим лишь, что общепринятое направление токов (от $+E_К$ — источника напряжения), обозначен-

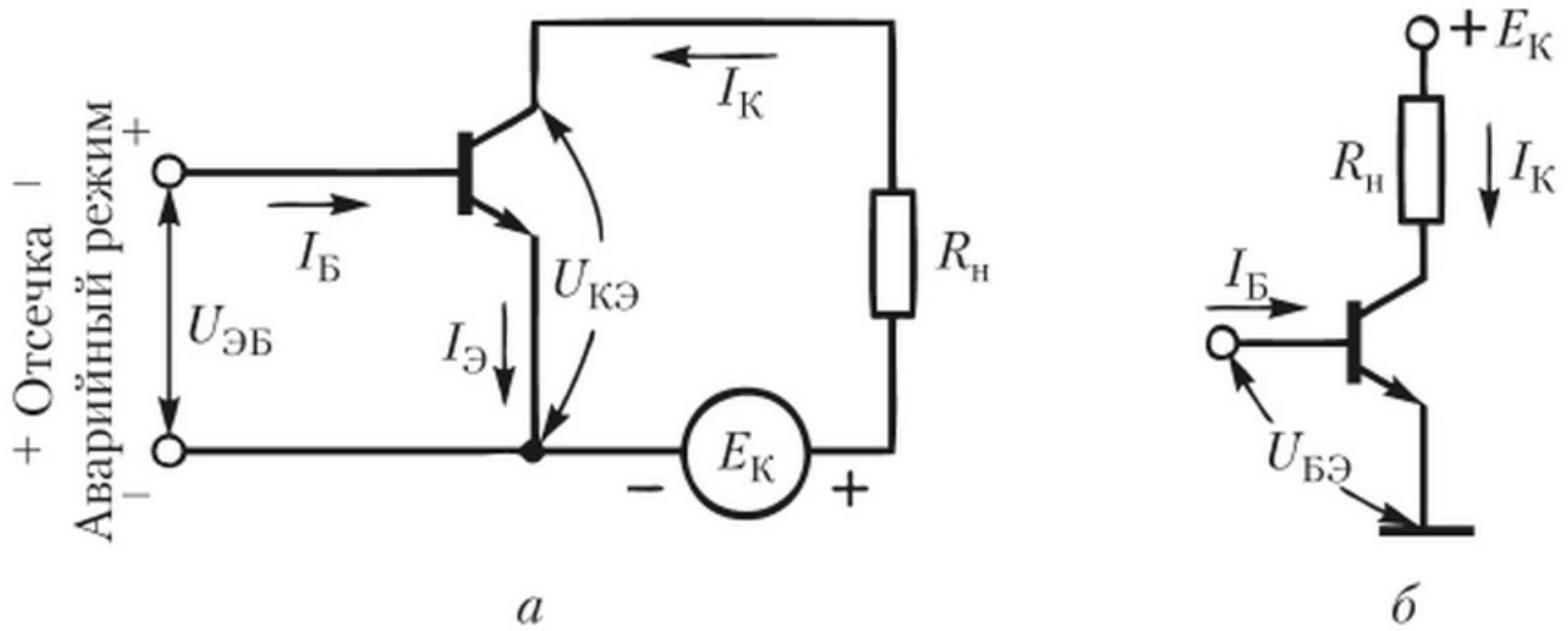


Рис. 1.11. **Схема включения транзистора с общим эмиттером (*а*); типовое изображение в схемах (*б*)**

ное на рис. 1.11, *а*, противоположно направлению движения электронов. Характерным признаком схемы с ОЭ является то, что нагрузка располагается в коллекторной цепи (рис. 1.11, *б*).

Так же как и для схемы с ОБ, входным сигналом в этой схеме является напряжение между базой и эмиттером, а выходными величинами — коллекторный ток $I_К$ и напряжение на нагрузке $U_{вых} = I_К \cdot R_н$. Транзистор в схеме с ОЭ характеризуется коэффициентом передачи тока

$$\beta = \Delta I_К / \Delta I_Б,$$

имеющим значения $\beta = 10...100$, который связан с коэффициентом α для схемы с ОБ соотношением:

$$\beta = \frac{\alpha}{1-\alpha}.$$

Оценим значения коэффициентов усиления схемы с ОЭ (их обозначают индексом «Э»).

Выходным током, как и в схеме с ОБ, является ток $I_К$, протекающий по нагрузке, а входным током (в отличие от схемы с ОБ) — ток базы $I_Б$; коэффициент усиления по току схемы с ОЭ равен

$$K_{IЭ} = \frac{\Delta I_{вых}}{\Delta I_{вх}} = \frac{\Delta I_К}{\Delta I_Б} = \beta$$

При $\alpha = 0{,}98$ $K_{IЭ} = 0{,}98/(1 - 0{,}98) \approx 50$, т.е. нескольким десяткам, что многократно превосходит аналогичный коэффициент у схемы с ОБ.

Входное сопротивление в схеме с ОЭ также значительно выше, чем в схеме с ОБ, так как в схеме с ОЭ входным током является ток базы, а в схеме с ОБ — во много раз больший ток эмиттера (а именно в $1/(1 - \alpha) \approx \beta$ раз):

$$R_{\text{вхЭ}} = \Delta U_{\text{вх}} / \Delta I_{\text{вх}} = \Delta U_{\text{БЭ}} / \Delta I_{\text{Б}}.$$

Величина входного сопротивления в схеме с ОЭ больше, чем в схеме с ОБ в $\approx \beta$ раз и составляет сотни ом.

Коэффициент усиления по напряжению в схеме с ОЭ соизмерим с таким же коэффициентом у схемы с ОБ:

$$K_{U\text{Э}} = \frac{\Delta U_{\text{вых}}}{\Delta U_{\text{вх}}} \quad \frac{\Delta I_{\text{К}} \cdot R_{\text{н}}}{\Delta I_{\text{Б}} \cdot R_{\text{вхЭ}}} \quad \frac{R_{\text{н}}}{R_{\text{вхЭ}}}. \qquad \approx \beta$$

По коэффициенту усиления по мощности схема с ОЭ за счет значительно большего коэффициента усиления по току также многократно превосходит схему с ОБ:

$$K_{P\text{Э}} = K_{I\text{Э}} \; K_{U\text{Э}} \qquad 2 \approx \frac{R_{\text{н}}}{R_{\text{вхЭ}}} \beta$$

и зависит от коэффициента передачи тока β и отношения сопротивления нагрузки к входному сопротивлению.

Благодаря отмеченным свойствам, схема с ОЭ нашла очень широкое применение.

1.3.3. Входные и выходные характеристики схемы с общим эмиттером

Работу схемы обычно описывают с помощью входных и выходных характеристик транзистора в той или иной схеме включения. Для схемы с ОЭ входная характеристика — это зависимость входного тока от напряжения на входе схемы, т.е. $I_{\text{Б}} = f\,(U_{\text{БЭ}})$ при фиксированных значениях напряжения коллектор — эмиттер ($U_{\text{КЭ}} = \text{const}$).

Выходные характеристики — это зависимости выходного тока, т.е. тока коллектора, от падения напряжения между коллектором и эмиттером транзистора $I_{\text{К}} = f\,(U_{\text{БЭ}})$ при токе базы $I_{\text{Б}} = \text{const}$.

Входная характеристика по существу повторяет вид характеристики диода при подаче прямого напряжения

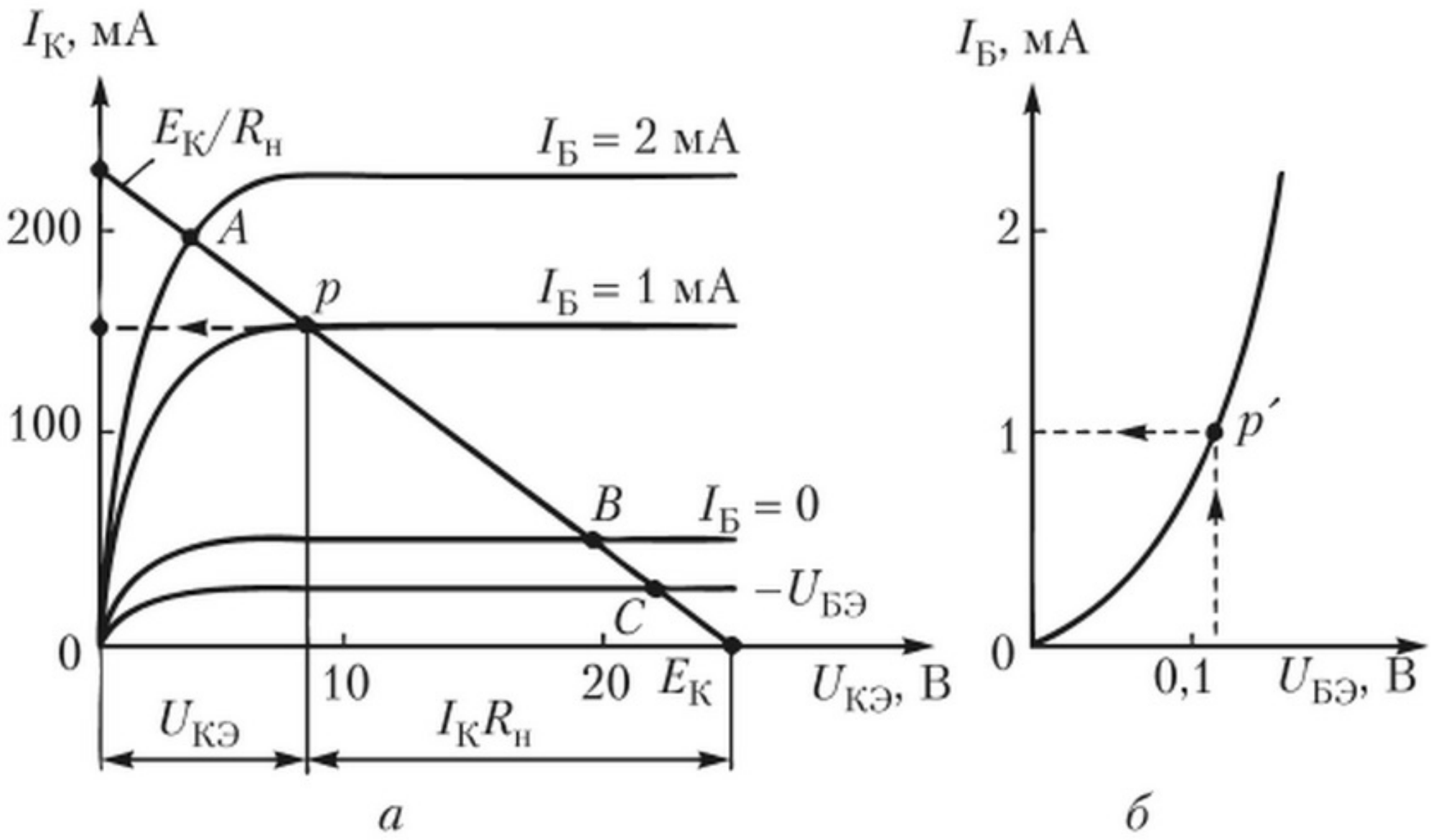


Рис. 1.12. **Выходные (*а*) и входная (*б*) характеристики транзистора в схеме с общим эмиттером**

(рис. 1.12, *б*). С ростом напряжения $U_{КЭ}$ входная характеристика будет незначительно смещаться вправо.

Вид выходных характеристик (рис. 1.12, *а*) резко различен в области малых (участок $0A$) и относительно больших значений $U_{КЭ}$. Напомним, что для нормальной работы транзистора необходимо, чтобы на переход база—эмиттер подавалось прямое напряжение, а на переход база—коллектор — обратное. Поэтому, пока $|U_{КЭ}| < U_{БЭ}$, напряжение на коллекторном переходе оказывается прямым, что резко уменьшает ток $I_К$. При $|U_{КЭ}| > U_{БЭ}$ напряжение на коллекторном переходе $U_{БК} = U_{КЭ} - U_{БЭ}$ становится обратным и, следовательно, мало влияет на величину коллекторного тока, который определяется в основном током эмиттера. При таком напряжении все носители, инжектированные эмиттером в базу и прошедшие через область базы, устремляются к внешнему источнику. При напряжении $U_{БЭ} < 0$ эмиттер носителей не инжектирует и ток базы $I_Б = 0$, однако в коллекторной цепи протекает ток $I_{К0}$ (самая нижняя выходная характеристика). Этот ток соответствует обратному току I_0 обычного p-n-перехода.

При работе транзистора изменяется его режим. Действительно, чем больше ток, протекающий через транзистор, тем больше падение напряжения на нагрузке, а следовательно, тем меньшее напряжение будет падать на самом транзисторе. Характеристики, представленные на рис. 1.12, *а*, *б*,

описывают лишь *статический режим* работы схемы. Для оценки динамики и влияния нагрузки на работу схемы используют графоаналитический метод расчета на основе входных и выходных характеристик. Рассмотрим этот метод на примере входных и выходных характеристик схемы с ОЭ.

Проведем прямую через точку $E_К$, отложенную на оси абсцисс, и точку $E_К / R_н$, отложенную на оси ординат выходных характеристик транзистора. Полученная прямая называется *нагрузочной*. Точка $E_К / R_н$ этой прямой соответствует такому току, который мог бы течь через нагрузку, если транзистор замкнуть накоротко. Точка $E_К$ соответствует другому крайнему случаю — цепь разомкнута, ток через нагрузку равен нулю, а напряжение $U_{КЭ}$ равно $E_К$. Точка p пересечения нагрузочной прямой со статической выходной характеристикой, соответствующей входному току $I_Б$, определит рабочий режим схемы, т.е. ток в нагрузке $I_К$, падение напряжения на ней $U_н = I_К \cdot R_н$ и падение напряжения $U_{КЭ}$ на самом транзисторе. На рис. 1.12, *а* точка p соответствует подаче в транзистор тока базы $I_Б = 1$ мА. Нетрудно видеть, что подача тока базы $I_Б = 2$ мА приводит к смещению рабочей точки в точку A и перераспределению напряжений между нагрузкой и транзистором.

Пример 1.1. Рассчитайте схему с ОЭ и $R_н = 110$ Ом при входном напряжении $U_{БЭ} = +0{,}1$ В, напряжении питания $E_К = +25$ В, используя характеристики транзистора.

Решение. Найдем отношение $E_К / R_н = 25/110 = 228$ мА и, отложив найденную точку на оси $I_К$ и значение $E_К = +25$ В на оси $U_{КЭ}$, проведем нагрузочную прямую.

По входной характеристике для напряжения $U_{БЭ} = 0{,}1$ В определим входной ток $I_Б = 1$ мА.

Точка пересечения p прямой с характеристикой, соответствующей $I_Б = 1$ мА, определит ток $I_К = 150$ мА.

Напряжение на нагрузке равно

$$U_н = I_К \cdot R_н = 0{,}15 \cdot 110 = 16{,}5 \text{ В}.$$

Напряжение между коллектором и эмиттером транзистора

$$U_{КЭ} = E_К - I_К \cdot R_н = 25 - 16{,}5 = 8{,}5 \text{ В}.$$

В заключение отметим, что режим, соответствующий точке A, называют *режимом насыщения* (при заданных значениях $R_н$ и $E_К$ ток $I_К$ в точке A достигает наибольшего

возможного значения). Режим, соответствующий точке *B* (входной сигнал равен нулю), а также точке *C* (входной сигнал отрицателен и запирает транзистор), называют *режимом отсечки*. Все промежуточные состояния транзистора с нагрузкой между точками *A* и *B* относятся к *активному режиму* его работы.

1.3.4. Схема с общим коллектором

В схеме с общим коллектором (ОК) нагрузка $R_н$ включена не в цепь коллектора, а в цепь эмиттера. Входным в этой схеме является напряжение между базой и корпусом, а выходным — между эмиттером и корпусом (рис. 1.13).

В отличие от схем с ОБ и ОЭ, в которых потенциал эмиттера был привязан к корпусу, в схеме с ОК потенциал эмиттера привязан к напряжению на нагрузке. Чтобы транзистор мог работать в активном режиме, необходимо, чтобы входное напряжение в этой схеме было выше напряжения на нагрузке на величину напряжения на $U_{БЭ}$:

$$U_{вх} = U_{вых} + U_{БЭ}.$$

В связи с этим значения входных напряжений в схеме с ОК оказываются в сотни раз больше, чем в схемах с ОБ и ОЭ.

Другой особенностью схемы с ОК является отсутствие усиления по напряжению. Как видно из схемы, $U_{вых}$ отли-

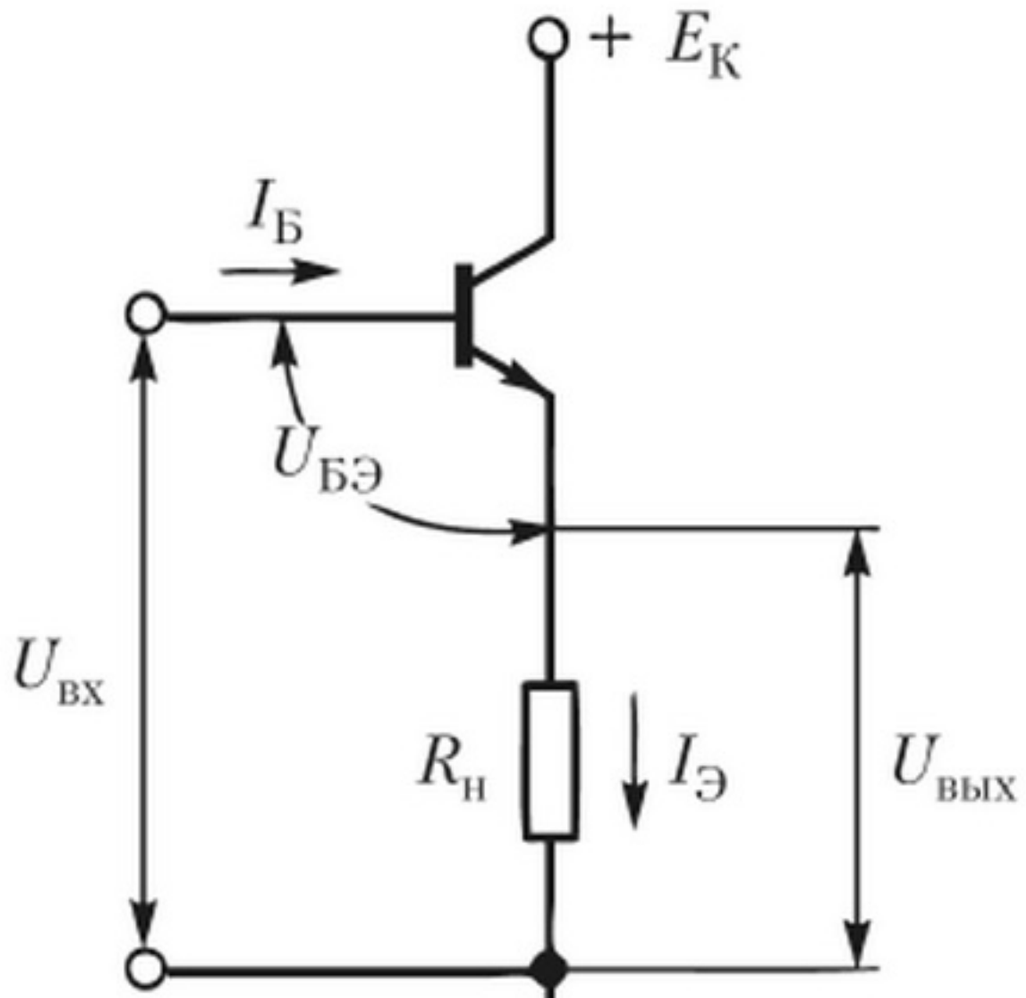


Рис. 1.13. **Схема включения транзистора с общим коллектором**

Содержание

чается от $U_{вх}$ на падение напряжения $U_{БЭ}$, которое при открытом транзисторе составляет доли вольт. Если входное напряжение увеличится на небольшую величину $\Delta U_{вх}$, то в первый момент произойдет увеличение управляющего напряжения $U_{БЭ}$, что приведет к увеличению тока, текущего через транзистор. Но с ростом тока увеличится и напряжение на нагрузке, а это приведет к уменьшению управляющего напряжения $U_{БЭ} = U_{вх} - U_{вых}$. Изменение входного напряжения $\Delta U_{вх}$ будет скомпенсировано аналогичным изменением выходного напряжения. Получается, что выходное напряжение будет в точности отслеживать все изменения входного. Поэтому схема с ОК получила название «эмиттерный повторитель». Коэффициент усиления по напряжению схемы с ОК $K_{UK} \approx 1$.

Оценим усилительные свойства схемы. Входным током по-прежнему является ток базы $I_Б$. Поэтому коэффициент усиления по току с учетом того, что $\beta = \Delta I_К / \Delta I_Б$, равен

$$K_{IK} = \frac{\Delta I_Э}{\Delta I_Б} = \frac{\Delta I_К + \Delta I_Б}{\Delta I_Б} = \frac{\Delta I_К}{\Delta I_Б} + 1 = \beta + 1,$$

т.е. примерно такой же, как и в схеме с ОЭ.

Оценим величину входного сопротивления схемы с ОК. Входное напряжение для схемы складывается из небольшого падения напряжения на база—эмиттерном переходе и падения напряжения на нагрузке, а входным током является ток базы. Поэтому

$$R_{вх} = \frac{\Delta U_{вх}}{\Delta I_{вх}} = \frac{\Delta U_{БЭ} + \Delta I_Э \cdot R_н}{\Delta I_Б}.$$

Поскольку напряжение на $U_{БЭ}$ значительно меньше напряжения на нагрузке, им можно пренебречь. Тогда, учитывая взаимосвязь между током эмиттера и током базы $I_Э \approx \beta I_Б$, величина входного сопротивления запишется как

$$R_{вх} \approx \frac{\beta \cdot \Delta I_Б \cdot R_н}{\Delta I_Б} \approx \beta R_н.$$

Таким образом, входное сопротивление схемы с ОК многократно превосходит входное сопротивление схем с ОЭ и ОБ и составляет десятки килоом.

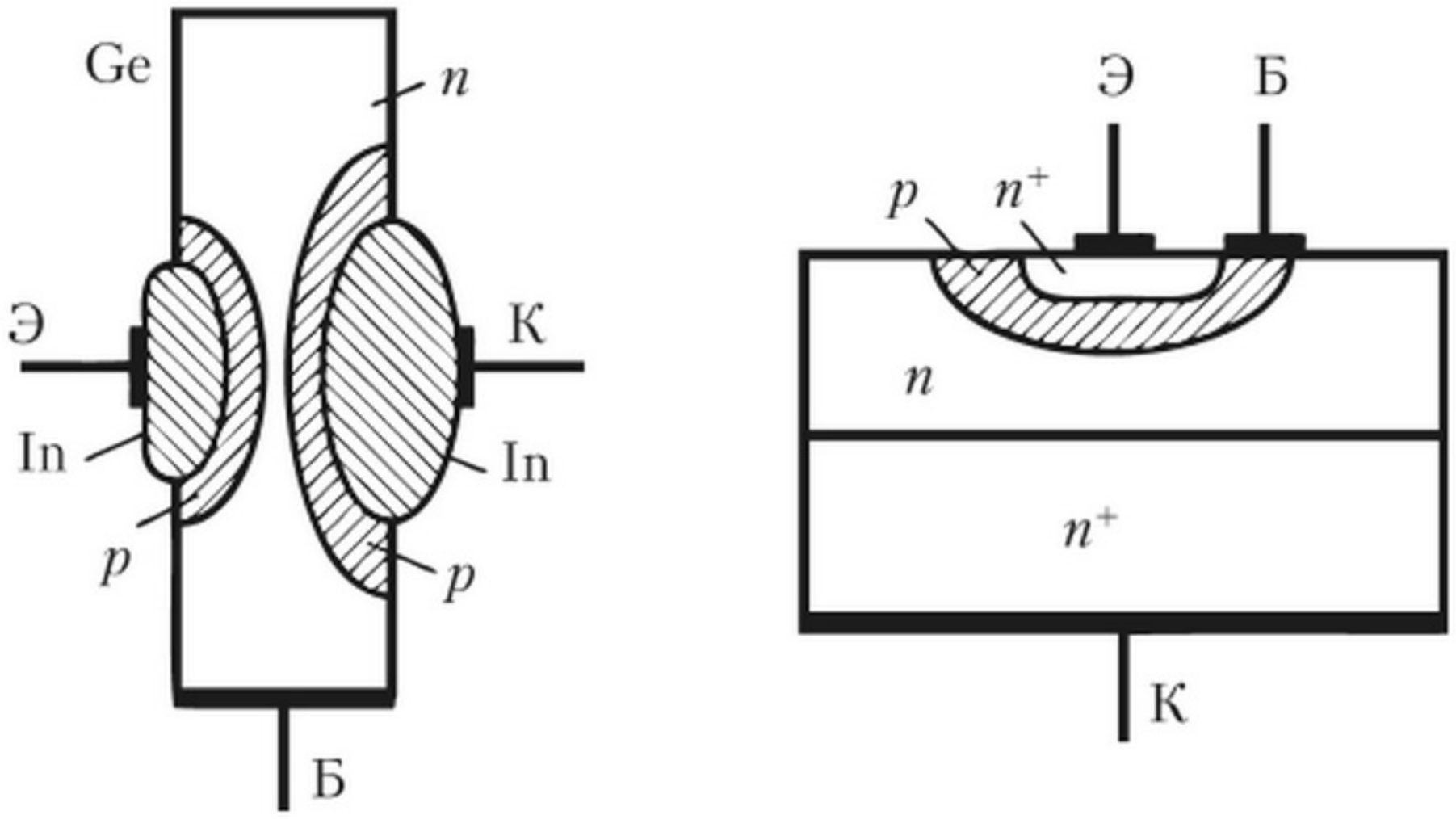
Благодаря отмеченным свойствам, эмиттерный повторитель используют в качестве выходного каскада устройств для усиления сигнала по мощности, когда усиление его по напряжению уже достигнуто предыдущими каскадами. Схема с ОК обеспечивает усиление по мощности

$$K_{PK} = K_{IK} \cdot K_{UK} \approx (\beta + 1)1 = \beta,$$

т.е. в десятки раз.

1.3.5. Технология изготовления биполярных транзисторов

При изготовлении биполярных транзисторов используют различные технологии выполнения *p-n*-переходов. Вот как, например, может быть изготовлен транзистор *методом вплавления* (рис. 1.14, *а*). Базой транзистора служит пластинка германия (или кремния) *n*-типа, на которую с двух сторон наплавляют капли акцепторной примеси, например индия. В приграничных слоях между германием и индием образуются *p*-области, представляющие эмиттер и коллектор, расстояние между которыми (толщина базы) должно быть очень маленьким (не больше нескольких микрометров). Кроме того, концентрация атомов примеси в области базы должна быть во много раз ниже, чем в области эмиттера. Это условие очень важно для работы транзистора.



а *б*

Рис. 1.14. **Устройство биполярного транзистора:**
а — сплавного; *б* — планарно-диффузионного

Более совершенным является *диффузионный метод* изготовления транзисторов, при котором в пластинке кремния *n*-типа (рис. 1.14, *б*) с помощью фотолитографии вытравливают окна и подвергают ее воздействию борной кислоты. Бор диффундирует в кремний на глубину нескольких микрометров, создавая в окне слой *p*-типа — базу будущей транзисторной структуры. Затем подобным же образом с помощью пятиокиси фосфора производят диффузию донорной примеси, образуя во внутренней части окна эмиттерную область проводимости n^+-типа с повышенной концентрацией примеси.

Коллектором в такой *n-p-n*-структуре служит исходная пластинка кремния *n*-типа, а ее нижний слой с повышенной концентрацией доноров (n^+) обеспечивает пониженное электрическое сопротивление коллектора, что важно для снижения нагрева транзистора протекающим через него током, особенно у мощных транзисторов.

Транзистор является основным элементом в интегральных микросхемах. Даже диоды, если они нужны в соответствии с электрической схемой, выполняют на основе транзисторов (используют либо база—эмиттерный переход, либо база—коллекторный, либо оба перехода вместе). Особенности технологии выполнения транзисторов в интегральной схемотехнике будут рассмотрены в разделе «Основы микроэлектроники».

1.4. Полевые транзисторы

Полевой транзистор — *это полупроводниковый прибор, имеющий три электрода: исток, сток и затвор*. Между истоком и стоком в кристалле полупроводника, из которого выполнен полевой транзистор, расположен *канал*, через который течет ток транзистора. Канал выполняется из полупроводника одного типа — *n* или *p*. Управление током, текущим через канал, осуществляется путем изменения проводимости канала, которая зависит от напряжения между затвором и истоком. В отличие от биполярных транзисторов, в которых ток транзистора от эмиттера к коллектору течет последовательно через два *p-n*-перехода, в полевых транзисторах ток течет через канал, который образуется в полупроводнике одного типа проводимости, и через *p-n*-переходы не течет. Так как направление тока в полевом

транзисторе — от истока — через канал — к стоку, а управление током осуществляется напряжением между затвором и истоком, то исток соответствует эмиттеру биполярного транзистора: сток — коллектору, а затвор — базе.

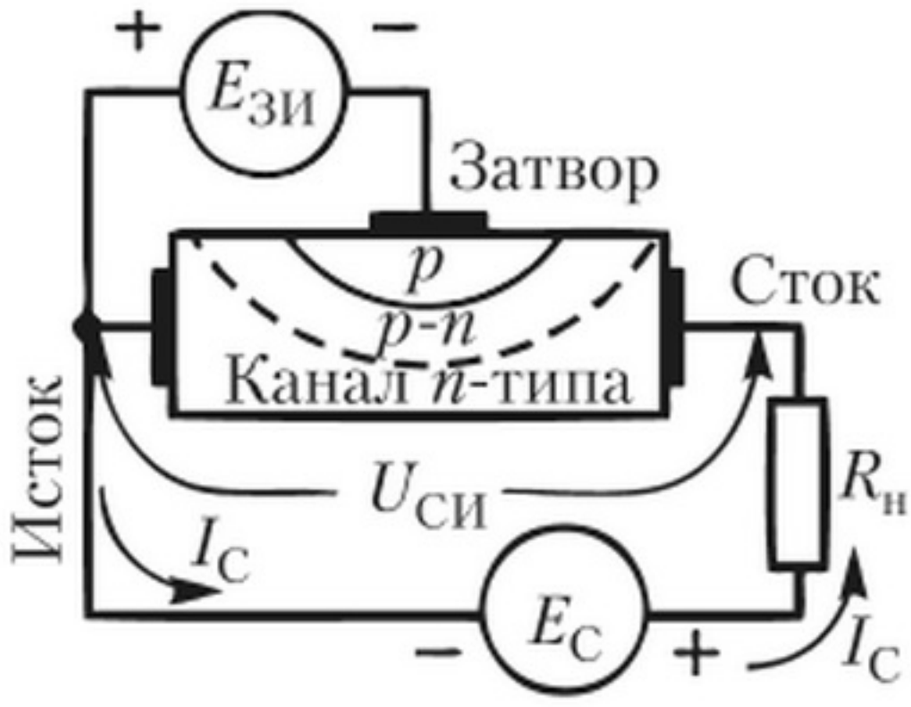
Изменение проводимости канала может осуществляться двумя способами. В зависимости от этого полевые транзисторы делятся на два основных вида: транзисторы *с управляющим p-n-переходом* и транзисторы *с изолированным затвором*.

1.4.1. Полевые транзисторы с управляющим *p-n*-переходом

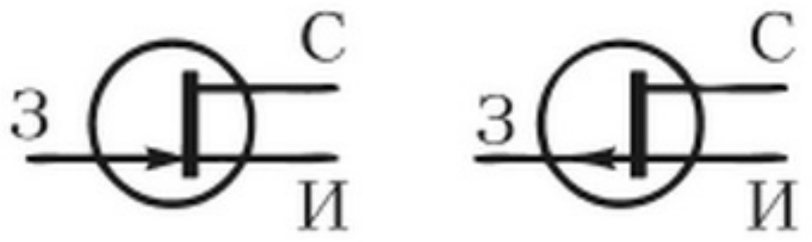
В полевых транзисторах с управляющим *p-n*-переходом управление током транзистора достигается путем изменения сечения канала за счет изменения области, занимаемой этим переходом. Управляющий *p-n*-переход образуется между каналом и затвором, которые выполняются из полупроводников противоположных типов проводимости. Так, если канал образован полупроводником *n*-типа (рис. 1.15, *а*), то затвор — полупроводником *p*-типа. Напряжение между затвором и истоком всегда подается обратной полярности, т.е. запирающей *p-n*-переход. Напомним, что при подаче напряжения обратной полярности область, занимаемая *p-n*-переходом, расширяется. При этом расширяется и область, обедненная носителями заряда, а значит, сужается область канала, через которую может течь ток. Причем, чем больше значение запирающего напряжения, тем шире область, занимаемая *p-n*-переходом, и тем меньше сечение и проводимость канала. Условные обозначения транзисторов с разными каналами показаны на рис. 1.15, *б*, *в*.

Так же, как и для биполярных транзисторов, для описания работы полевых транзисторов используют выходные характеристики. Выходная характеристика полевого транзистора — это зависимость тока стока I_C от напряжения между стоком и истоком при фиксированном напряжении между затвором и истоком. Выходные характеристики полевого транзистора с управляющим *p-n*-переходом показаны на рис. 1.15, *г*. Как видно, они очень похожи на выходные характеристики биполярного транзистора.

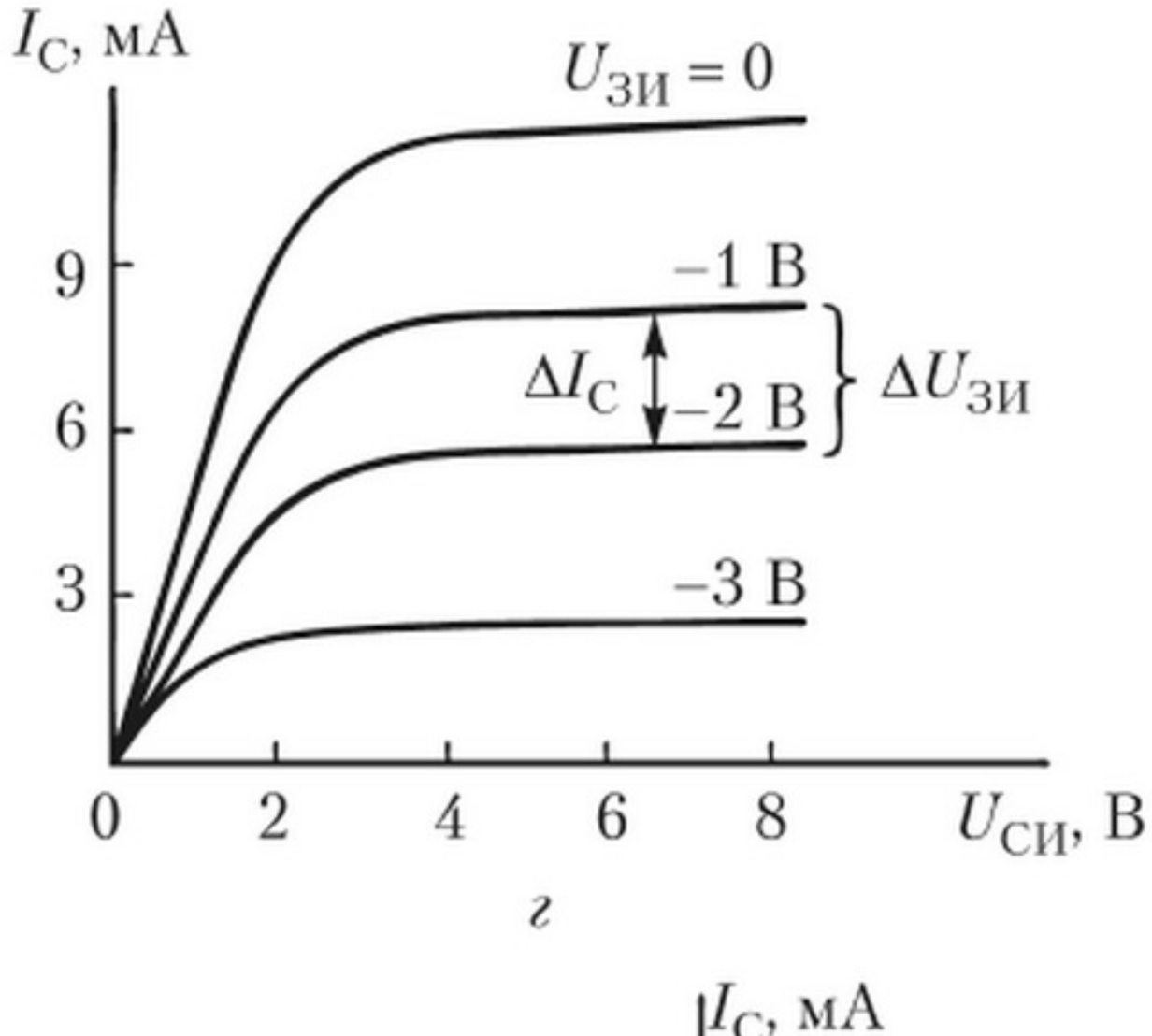
В отличие от биполярного, работа полевого транзистора может также описываться непосредственной зависимостью выходного параметра — тока стока от входного — управляющего напряжения между затвором и истоком. Эти харак-



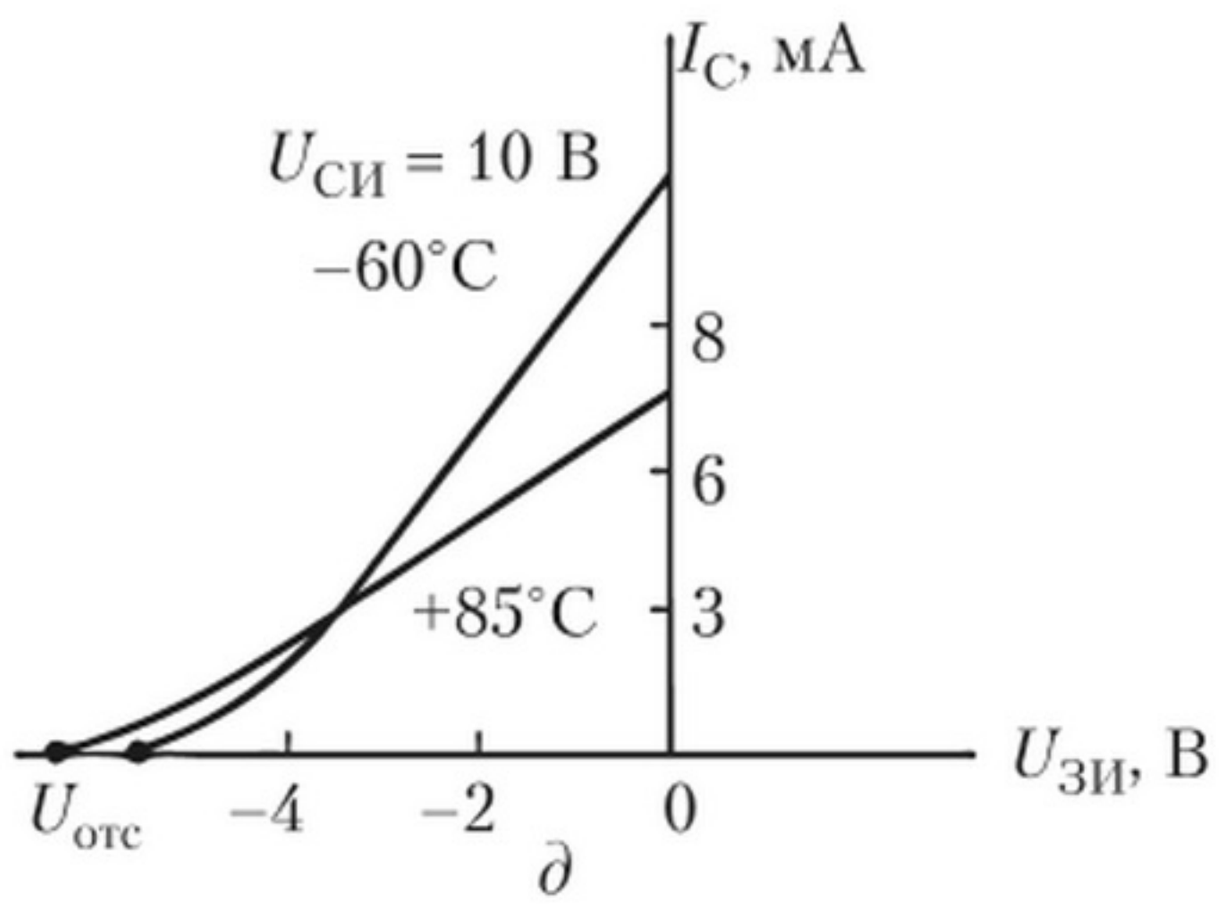
а



б *в*



г



д

Рис. 1.15. **Полевые транзисторы с управляющим *p-n*-переходом (*а*); условное обозначение транзисторов с *n*-каналом (*б*) и *p*-каналом (*в*); выходные характеристики (*г*); стокозатворные (передаточные) характеристики (*д*)**

Содержание

теристики называются *передаточными*, или *стокозатворными* (рис. 1.15, *д*). В зависимости от температуры, эти характеристики несколько изменяются. Напряжение $U_{ЗИ}$, при котором канал полностью перекрывается ($I_C \approx 0$), называется напряжением *отсечки* $U_{отс}$. Управляющее действие затвора характеризуют крутизной, которая может быть определена по выходным характеристикам (см. рис. 1.15, *г*):

$$S = \Delta I_C / \Delta U_{ЗИ}, \text{ при } U_{СИ} = \text{const}.$$

Так как управляющий p-n-переход всегда заперт, у полевых транзисторов практически отсутствует входной ток. Благодаря этому они имеют очень высокое входное сопротивление и практически не потребляют мощности от источника управляющего сигнала. Это свойство относится не только к транзисторам с управляющим p-n-переходом, но и ко всем полевым транзисторам, что выгодно отличает их от биполярных.

1.4.2. Полевые транзисторы с изолированным затвором

Если в полевых транзисторах с управляющим p-n-переходом затвор имеет электрический контакт с каналом, то в полевых транзисторах с изолированным затвором такой контакт отсутствует. В этих транзисторах (рис. 1.16, *а*, *б*) затвор представляет собой тонкую пленку металла, изолированного от полупроводника. В зависимости от вида изоляции различают МДП и МОП-транзисторы. Аббревиатура «МДП» расшифровывается как «металл — диэлектрик — полупроводник», а «МОП» — как «металл — оксид — полупроводник». В последнем случае под «оксидом» понимается оксид кремния, который является высококачественным диэлектриком.

Исток и сток формируют в виде сильно легированных областей полупроводника. За счет этого области истока и стока имеют высокую концентрацию носителей, что отмечено на рисунке знаком «+». Как МДП-, так и МОП-транзисторы могут быть выполнены с каналом p- и n-типов. Канал в этой группе транзисторов может быть *встроенным* (т.е. созданным при изготовлении) и *индуцированным* (т.е. наводящимся под влиянием напряжения, приложенного к затвору).

Полевой транзистор с встроенным каналом. На рис. 1.16, *а* изображен МДП-транзистор с встроенным каналом n-типа (тонким слоем полупроводника n-типа), соединяющим

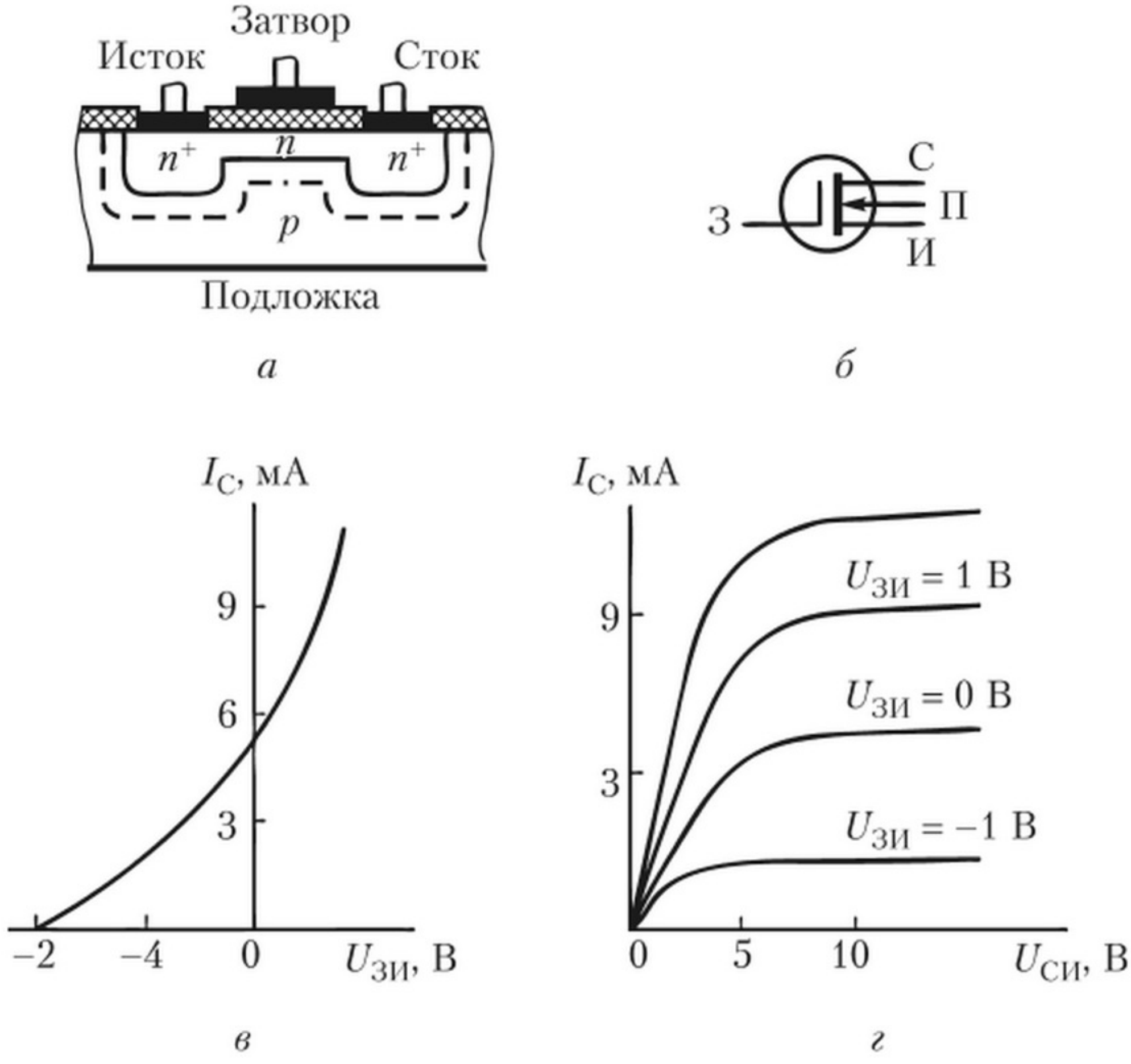


Рис. 1.16. **Полевой транзистор с изолированным затвором и встроенным каналом (*а*); его условное обозначение (*б*); передаточная (*в*) и выходные (*г*) характеристики**

исток и сток (n^+-области). Эти области образованы в подложке — полупроводнике p-типа. Строго говоря, в МДП- и МОП-транзисторах не три, а четыре электрода, включая подложку. Однако часто подложку электрически соединяют с истоком (или стоком), образуя три вывода.

В зависимости от полярности напряжения $U_{ЗИ}$, приложенного к затвору относительно истока, в канале может изменяться концентрация основных носителей (в рассматриваемом случае — электронов). При отрицательном напряжении на затворе $U_{ЗИ}$ электроны выталкиваются из области канала в области n^+, канал обедняется носителями и ток I_C снижается. Положительное напряжение на затворе втягивает электроны из областей n^+ в канал и ток I_C через канал возрастает. Таким образом, в отличие от полевого транзистора с p-n-переходом в этом полевом

транзисторе управляющее напряжение может быть как отрицательным, так и положительным, что отражено на его передаточной (рис. 1.16, *в*) и выходных (рис. 1.16, *г*) характеристиках.

Полевой транзистор с индуцированным каналом. Этот вид транзистора отличается от предыдущего тем, что при отсутствии напряжения на затворе канал отсутствует (рис. 1.17, *а*), так как *n*-области истока и стока образуют с *p*-подложкой два *p*-*n*-перехода, включенные навстречу друг другу, и, значит, при любой полярности напряжения $U_{СИ}$ один из переходов заперт.

Если же на затвор подать напряжение больше порогового $U_{ЗИ} > U_{ЗИпор}$, то созданное им электрическое поле вытягивает электроны из n^+-областей (и в какой-то мере из подложки), образуя тонкий слой *n*-типа в приповерхностной области *p*-подложки (рис. 1.17, *б*). Этот слой соединяет исток и сток, являясь каналом *n*-типа. От подложки канал изолирован возникшим обедненным слоем.

Таким образом, полевые транзисторы с индуцированным *n*-каналом (*n*-МОП-транзисторы), в отличие от рассмотренных ранее полевых транзисторов, управляются только положительным сигналом $U_{ЗИ}$ (рис. 1.17, *г*). Значение порогового напряжения у них 0,2÷0,1 В.

Значительно больше пороговое напряжение у *p*-МОП-транзистора, принцип работы которого аналогичен *n*-МОП-транзистору. Но в связи с тем, что носителями в нем служат дырки, а не электроны, полярность всех напряжений у этого транзистора противоположна *n*-МОП-транзистору. Значение порогового напряжения этого типа транзисторов составляет 2÷4 В (рис. 1.17, *д*).

Как и биполярные, полевые транзисторы можно включать по схеме с общим затвором (ОЗ), общим истоком (ОИ) и общим стоком (ОС). Как правило, используют схему с ОИ, так как она, подобно схеме с ОЭ биполярных транзисторов, позволяет получить значительные коэффициенты усиления по току, напряжению и мощности одновременно.

Преимущества полевых транзисторов:

1) высокое входное сопротивление в схеме с ОИ;

2) малый уровень собственных шумов, так как перенос тока осуществляют только основные для канала носители и, следовательно, нет рекомбинационного шума;

3) высокая устойчивость против температурных и радиоактивных воздействий;

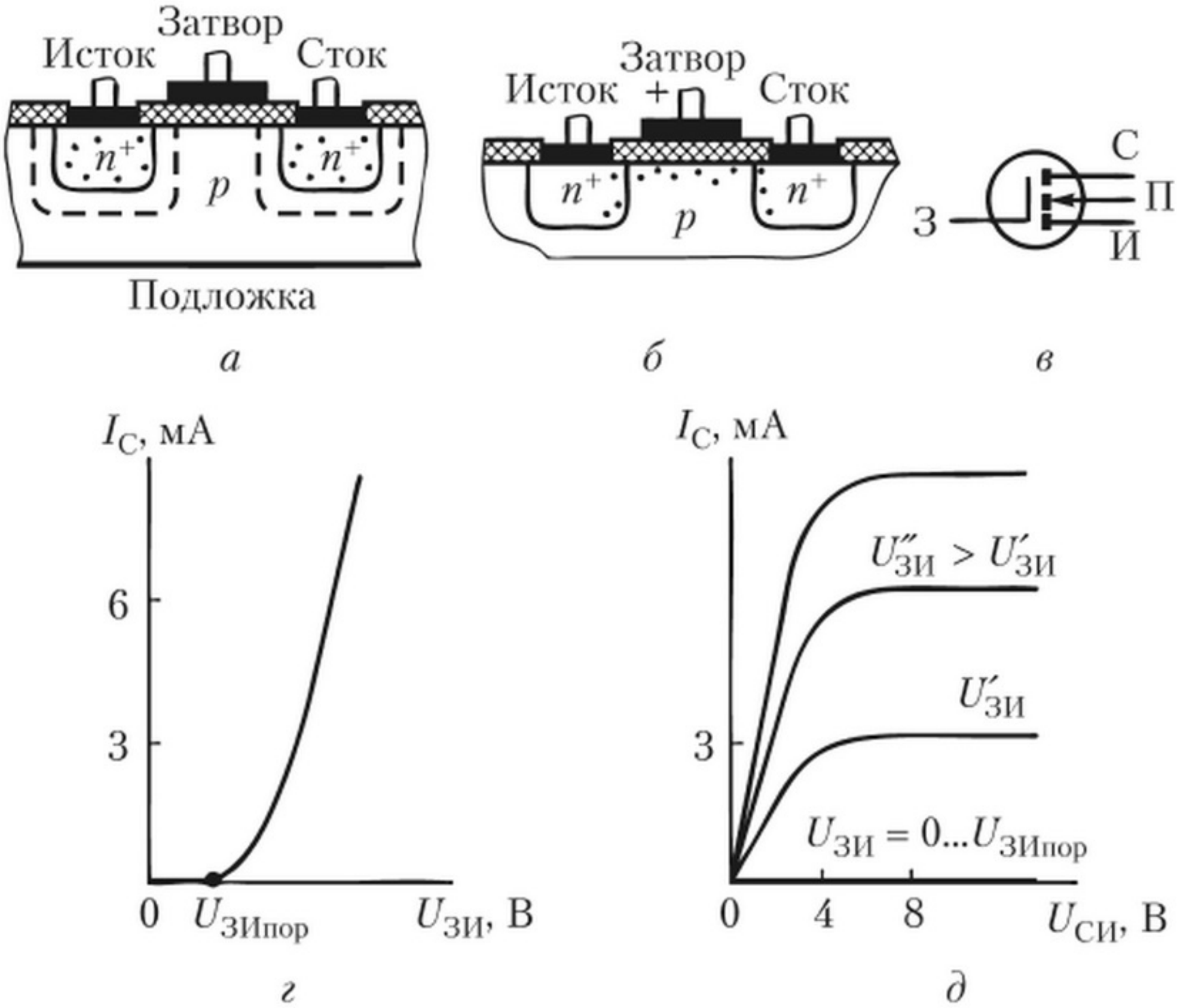


Рис. 1.17. **Полевой транзистор с индуцированным каналом в исходном состоянии (*а*) и при приложенном напряжении на затворе (*б*); его условное обозначение (*в*); передаточная (*г*) и выходные (*д*) характеристики**

4) высокая плотность расположения элементов при изготовлении интегральных схем.

Отметим также интересную особенность полевых транзисторов: в принципе исток и сток в транзисторах равноправны, т.е. в зависимости от приложенной полярности напряжения исток и сток могут меняться местами. На этом свойстве основано использование полевых транзисторов в качестве электронных ключей вместо обычных контактных переключателей.

Полевые транзисторы широко используются в усилителях, генераторах и другой радиоэлектронной аппаратуре, а МОП-транзисторы являются основой для разработки всех современных средств вычислительной техники, включая микропроцессоры, микроконтроллеры, полупроводниковую память.

Сравнивая условные обозначения транзисторов (см. рис. 1.8, *б*; 1.9, *б*; 1.15, *б*, *в*; 1.16, *б*; 1.17, *в*), подчерк-

нем, что стрелка в них *всегда направлена от p-области к n-области*, что позволяет легко установить, например, тип канала полевого транзистора.

1.5. Тиристоры

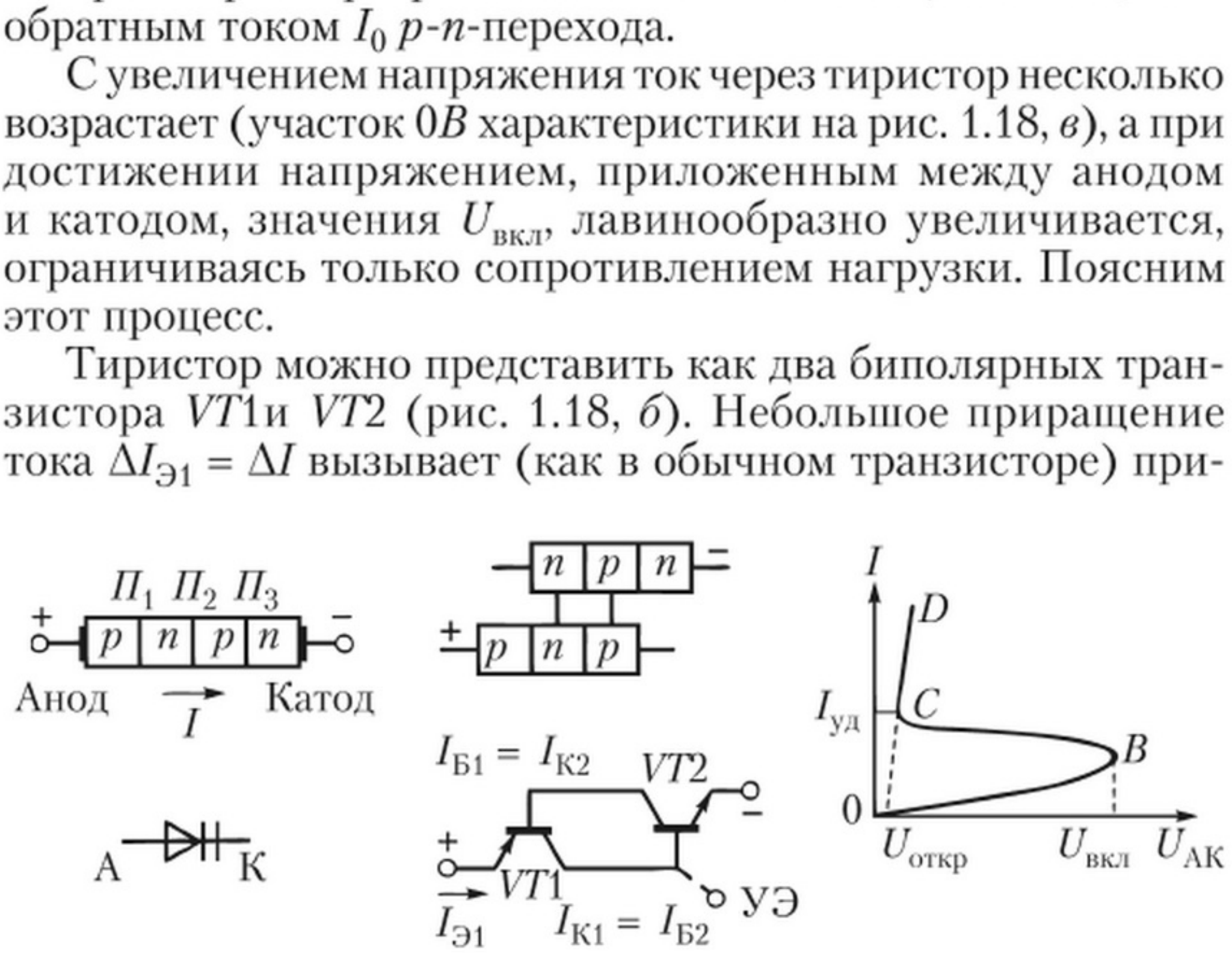
Тиристоры — это полупроводниковые приборы с тремя или более p-n-переходами, которые имеют два устойчивых состояния и применяются как мощные электронные ключи.

Диодные тиристоры (*динисторы*) имеют два вывода от крайних чередующихся *p*- и *n*-областей (рис. 1.18, *а*).

Вывод, соединенный с крайней *p*-областью, называется *анодом*, а с крайней *n*-областью — *катодом*. Внешнее напряжение U является прямым по отношению к переходам $П_1$ и $П_3$ и обратным для перехода $П_2$, поэтому переходы $П_1$ и $П_3$ открыты (подобно открытым диодам), а переход $П_2$ заперт. В результате напряжение U почти целиком приложено к $П_2$ и через тиристор протекает небольшой ток, являющийся обратным током I_0 *p-n*-перехода.

С увеличением напряжения ток через тиристор несколько возрастает (участок $0B$ характеристики на рис. 1.18, *в*), а при достижении напряжением, приложенным между анодом и катодом, значения $U_{вкл}$, лавинообразно увеличивается, ограничиваясь только сопротивлением нагрузки. Поясним этот процесс.

Тиристор можно представить как два биполярных транзистора $VT1$и $VT2$ (рис. 1.18, *б*). Небольшое приращение тока $\Delta I_{Э1} = \Delta I$ вызывает (как в обычном транзисторе) при-



а *б* *в*

Рис. 1.18. **Тиристор диодный и его условное обозначение (*а*), эквивалентная схема (*б*) и вольт-амперная характеристика (*в*)**

Содержание

ращение тока коллектора $\Delta I_{К1}$, который, поступая в базу транзистора $VT2$, вызывает приращение его коллекторного тока:

$$\Delta I_{К2} = \Delta I_{Б2}\beta_2 = \Delta I_{К1}\beta_2,$$

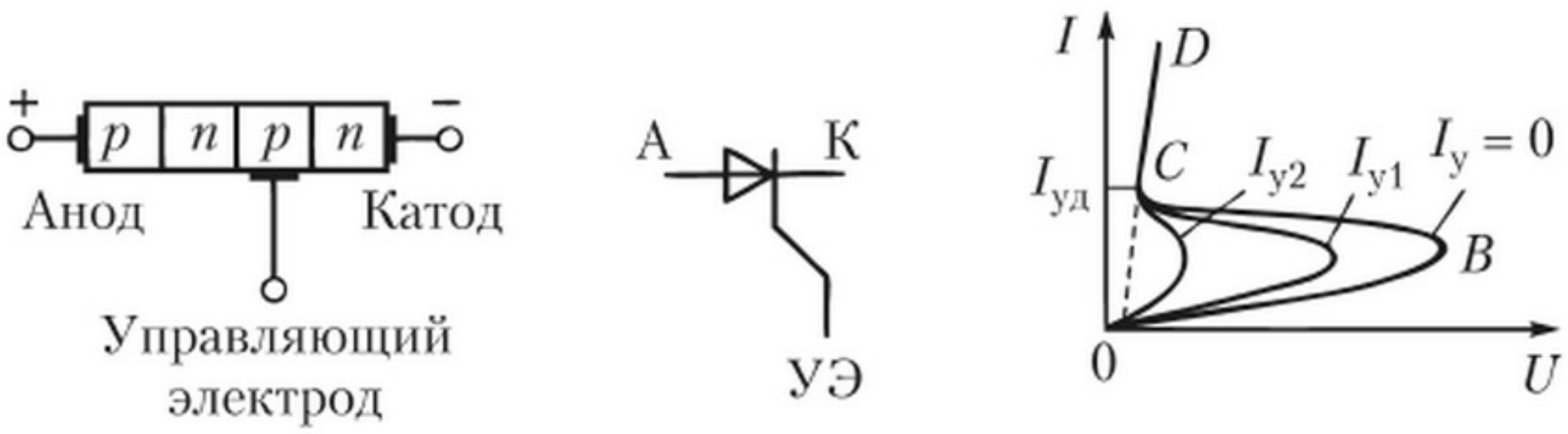
где β_2 — коэффициент передачи тока $VT2$. Но ток коллектора второго транзистора, как показано по схеме, является базовым для первого транзистора ($\Delta I_{К2} = \Delta I_{Б1}$), поэтому ток $I_{К1}$, в свою очередь, увеличивается:

$$\Delta I_{К1} = \Delta I_{Б1}\beta_1 = \Delta I_{К2}\,\beta_1 = \Delta I_{К1}\,\beta_2\,\beta_1 \text{ и т.д.}$$

Этот процесс соответствует участку BC вольт-амперной характеристики с отрицательным сопротивлением и переводит тиристор в открытое состояние, когда он ведет себя как диод в прямом направлении.

Чтобы запереть (погасить) тиристор, необходимо каким-либо образом уменьшить ток I, протекающий через него, до значения, меньше удерживающего $I_{уд}$. Если напряжение U, питающее схему, переменное, то тиристор запирается в отрицательный полупериод, когда ток I достигает нуля, если же оно постоянное, то для запирания тиристора применяют так называемые *схемы гашения*.

Перевод тиристора из запертого состояния в открытое можно вызвать не только повышением анодного напряжения, но и кратковременным увеличением тока базы в одном из транзисторов его эквивалентной схемы. Для этого от одной из баз делают вывод *управляющий электрод* (УЭ) (рис. 1.19, *а*, *б*). Подавая импульс тока управления $I_у$, можно вызвать лавинообразное увеличение тока при $U < U_{вкл}$



а *б* *в*

Рис. 1.19. **Триодный (управляемый) тиристор:**
а — структура; *б* — условное обозначение; *в* — вольт-амперные характеристики

Содержание

(рис. 1.19, *в*). Такие тиристоры называют триодными (управляемыми) тиристорами.

Основные параметры тиристоров:

- наибольший прямой ток — до 2000 А;
- допустимое обратное напряжение — от 100 до 2400 В;
- допустимая частота переключений — до 2000 Гц.

Тиристоры нашли свое применение в силовой электронике и электротехнике — там, где требуется формирование мощных питающих напряжений постоянного или переменного тока, питающих напряжений с регулируемой частотой, специальной формы. В частности, на основе тиристоров разрабатываются устройства регулирования частотой вращения электродвигателей, в том числе в приводах станков.

1.6. Основы микроэлектроники

Микроэлектроника — это направление электроники, позволяющее с помощью комплекса технологических, конструктивных и схемотехнических средств создавать малогабаритные, высоконадежные и экономичные электронные устройства.

Микроэлектроника основана на применении *интегральных микросхем* (ИМС), в которых элементы неразделно связаны между собой и представляют единое целое. ИМС изготавливают на основе кристалла полупроводника, в качестве которого чаще всего используют кремний. В кристалле кремния создаются *p*-*n*-переходы, образующие как активные, так и пассивные элементы электрической схемы. Элементы микросхемы связываются между собой электрически с помощью тонких металлических перемычек. Такой кристалл называют чип (англ. *chip* — кристалл). Характеристикой сложности ИМС является уровень интеграции, оцениваемый числом транзисторов, которые могут быть реализованы на кристалле.

В зависимости от уровня интеграции ИМС делят на несколько категорий:

- малые ИМС — до 10 элементов (МИС);
- средние ИМС — от 10 до 100 элементов (СИС);
- большие ИМС — от 100 до 10^5 элементов (БИС);
- сверхбольшие ИМС — 10^5 и более элементов (СБИС).

В качестве элементов в микросхемах чаще всего выступают транзисторы, что в особенности касается цифро-

вых микросхем. Современные СБИС содержат несколько сотен миллионов транзисторов, причем степень интеграции постоянно повышается. Необходимо отметить, что четкой границы между БИС и СБИС не существует, и часто их объединяют в один класс БИС/СБИС. На сегодняшний день практическое использование находят все категории ИМС.

Кроме степени интеграции ИМС могут классифицироваться в зависимости от их функционального назначения на два больших класса: цифровые и аналоговые. Цифровые ИМС оперируют с входными напряжениями, дискретно меняющими свое значение, которое соответствует либо «1», либо «0». Аналоговые ИМС используются для преобразования непрерывно изменяющихся во времени сигналов.

Цифровые ИМС в зависимости от степени интеграции могут выполнять простейшие логические преобразования (МИС), образовывать целые узлы цифровых устройств, таких как малоразрядные регистры, счетчики, дешифраторы, сумматоры и т.п. (СИС). Цифровые БИС/СБИС способны выполнять функции уже не отдельного узла, а целой системы. К ним относятся все микропроцессорные ИМС, микросхемы памяти, ИМС программируемой логики, ИМС, реализующие стратегию «Система в кристалле».

Аналоговые ИМС выполняют разнообразные функции: усиление сигналов переменного и постоянного токов, генерирование колебаний различной формы, обеспечение других ИМС стабилизированным напряжением питания, цифроаналоговое и аналого-цифровое преобразования сигналов, фильтрацию сигналов, их модуляцию и демодуляцию и т.п.

1.6.1. Технология полупроводниковых ИМС

Изготовление полупроводниковых ИМС осуществляют, используя два основных технологических процесса: *диффузию примесей*, создающих в полупроводнике область с типом проводимости, противоположным исходному, и *эпитаксиальное наращивание* слоя кремния на кремниевую подложку, имеющую противоположный тип проводимости.

Все элементы схемы формируются в так называемых *островках*, образованных в кристалле и изолированных между собой. Металлические полоски, необходимые для соединения элементов в электрическую схему, напыляют

на поверхность пластины-кристалла. Для этого электроды всех элементов выводятся на поверхность пластины и размещаются в одной плоскости, в одном *плане*. Поэтому технология изготовления схем с помощью диффузии называется *планарно-диффузионной*, а с помощью эпитаксиального наращивания — *эпитаксиально-планарной*.

Исходным материалом для изготовления ИМС по планарно-диффузионной технологии является слабо легированная пластина кремния *p*-типа, на которую методом фотолитографии наносят защитный слой SiO_2 (рис. 1.20). Через окна в защитном слое производится диффузия примеси *n*-типа, в результате чего образуются островки, границы которых упираются снизу в защитный слой, что резко снижает возможность протекания токов утечки по поверхности. Между островками и подложкой образуется *p-n*-переход, к которому подключают напряжение таким образом, чтобы этот переход был заперт (т.е. минусом на *p*-подложке). В результате островки становятся изолированными друг от друга.

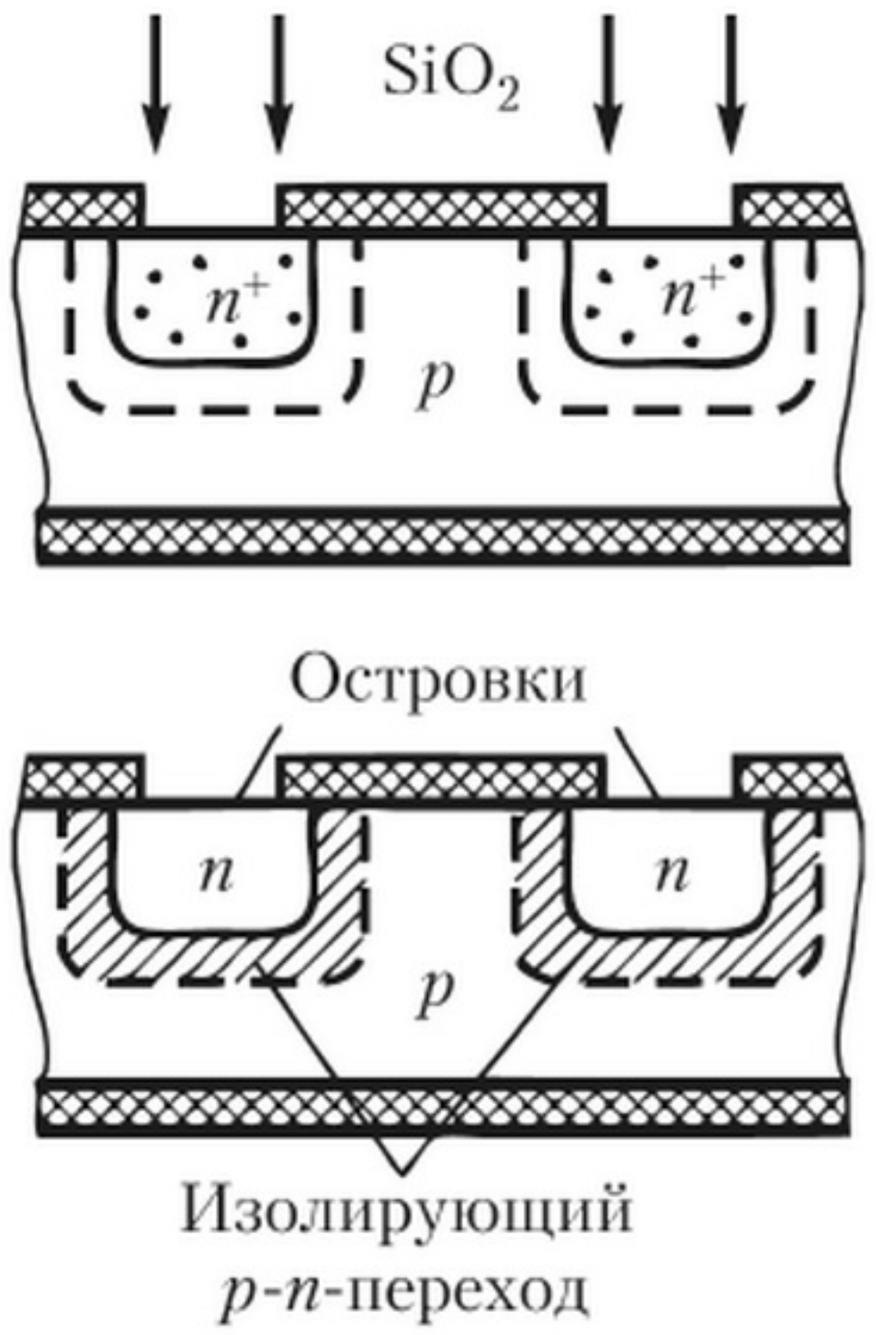


Рис. 1.20. **Изготовление островков по планарно-диффузионной технологии**

Исходным материалом при эпитаксиально-планарной технологии служит пластина кремния *n*-типа со слоем SiO_2 (рис. 1.21, *а*), в которой вытравливают продольные и поперечные канавки (рис. 1.21, *б*). Полученную фигур-

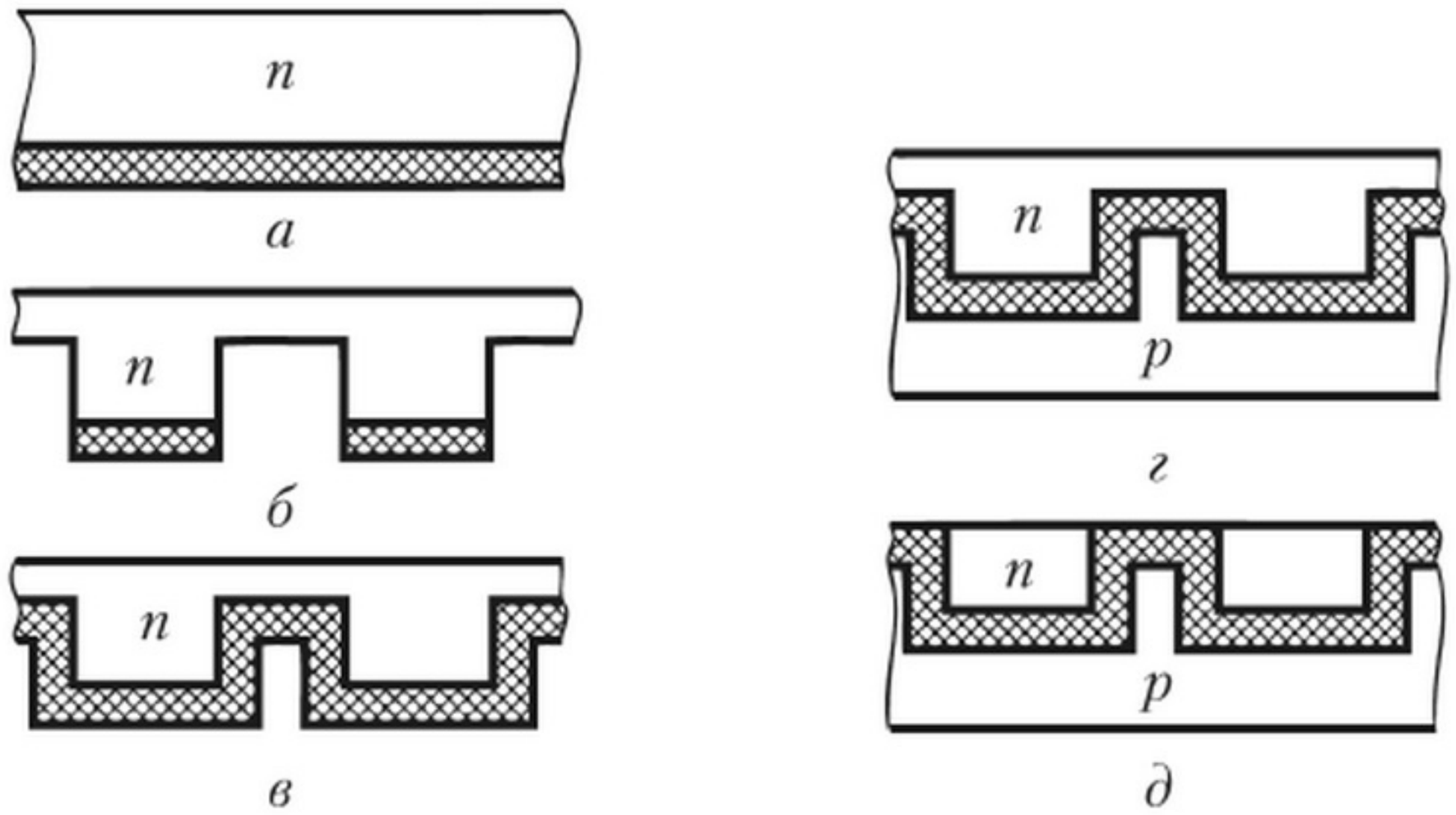


Рис. 1.21. **Изготовление островков по эпитаксиально-планарной технологии**

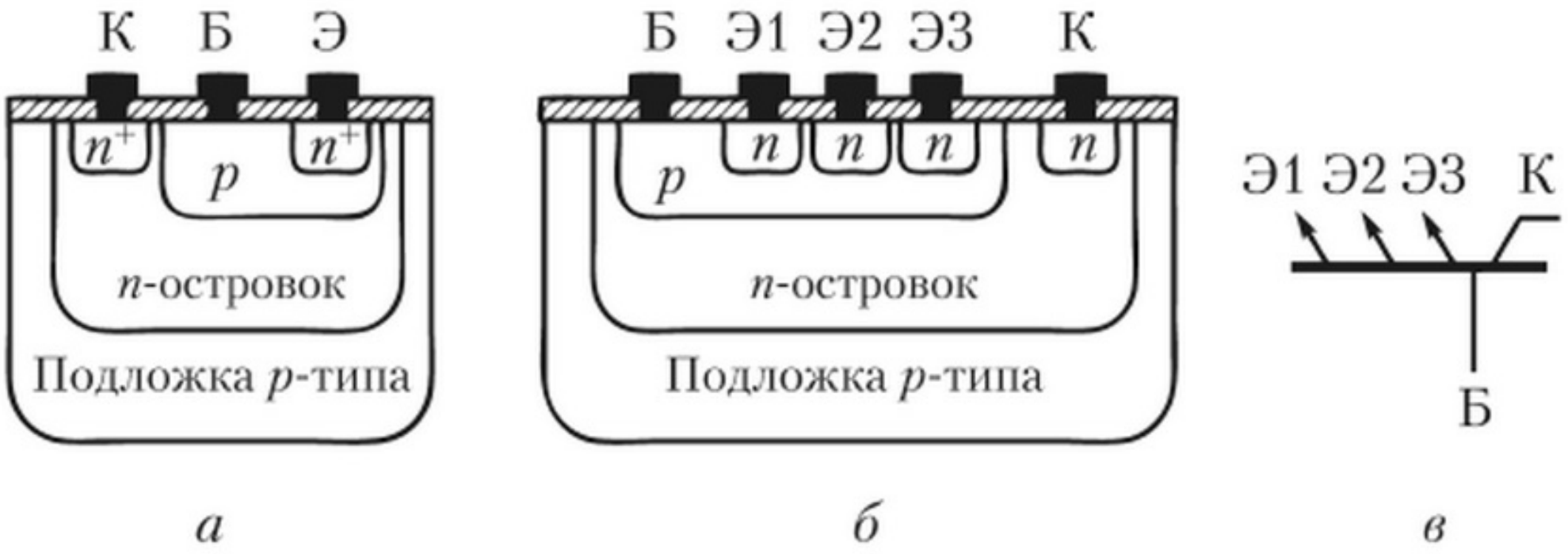
ную поверхность (в виде шахматной доски) снова окисляют, создавая изоляционный слой диоксида кремния (рис. 1.21, *в*). На этот слой эпитаксиально наращивают слой кремния собственной проводимости (рис. 1.21, *г*), а верхний слой кремния *n*-типа сошлифовывают. Полученные таким образом островки (рис. 1.21, *д*) надежно изолированы друг от друга фигурным слоем диэлектрика и емкость между ними существенно меньше, чем в предыдущем случае. Однако такая технология ИМС сложнее и стоимость их изготовления выше.

В полученных тем или иным способом островках формируют как активные, так и пассивные элементы методом диффузионной технологии или эпитаксиальным наращиванием.

1.6.2. Компоненты ИМС

Транзисторы ИМС получают последовательной диффузией донорных и акцепторных примесей в островки, созданные тем или иным способом (рис. 1.22, *а*). Характерным для них является расположение выводов в одной плоскости.

Для осуществления логических операций созданы *многоэмиттерные транзисторы* (рис. 1.22, *б*, *в*), применение которых основано на их свойстве оставаться открытыми, если хотя бы к одному из эмиттеров приложено относительно базы прямое напряжение. Запирание транзисторов



а *б* *в*

Рис. 1.22. **Транзисторы интегральных схем:**
а — биполярный транзистор; *б* — многоэмиттерный транзистор; *в* — условное обозначение многоэмиттерного транзистора

происходит тогда, когда на все эмиттеры поданы обратные напряжения.

Наряду с биполярными в ИМС широко применяют полевые МДП-транзисторы, особенно МОП-транзисторы с индуцированным каналом. В основе их изготовления, так же как и биполярных, лежит планарная технология. Так, при изготовлении островков по планарно-диффузионной технологии получается практически готовая заготовка для МОП-транзистора. Каждый из двух соседних островков (см. рис. 1.20) может быть стоком или истоком этого транзистора. Поэтому для их изготовления требуется меньшее по сравнению с эпитаксиально-планарной технологией количество операций.

Диоды ИМС специально не изготавливают, а в качестве их используют транзисторы, включаемые по одной из схем (рис. 1.23) в зависимости от требований, предъявляемых к диоду.

Так, на рис. 1.23, *а, б* в качестве диода используется p-n-переход база—эмиттер. Диод открыт при указанной на рисунке полярности приложенного напряжения и закрывается при противоположной полярности. Диоды, выполненные в соответствии с рис. 1.23, *а, б*, обеспечивают высокое быстродействие, но малый ток. Диоды, выполненные в соответствии с рис. 1.23, *в*, используют два параллельных p-n-перехода и, соответственно, больший ток, но меньшее быстродействие. Диоды, в соответствии с рис. 1.23, *г, д*, имеют наибольшее допустимое обратное напряжение, подобно тому, как в биполярных транзисторах наи-

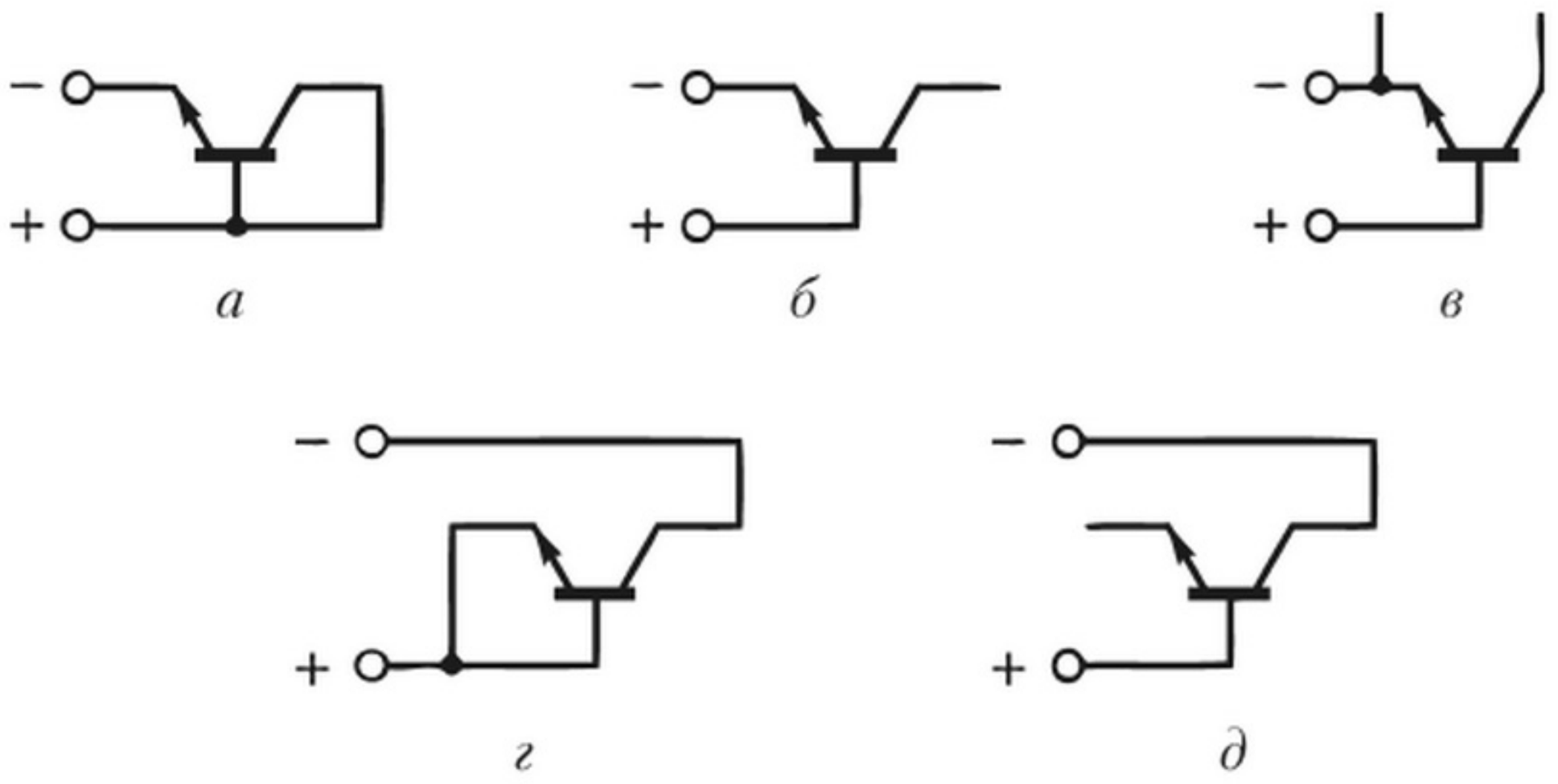


Рис. 1.23. **Варианты выполнения диодов на основе транзистора**

большее напряжение может быть приложено к переходу база—коллектор.

Резисторы ИМС получают диффузией примесей в отведенные для них островки одновременно с созданием эмиттерных и базовых областей транзисторов. В процессе эмиттерной диффузии создаются резисторы с относительно низким сопротивлением (так как в эмиттерной области концентрация носителей велика), а в процессе базовой диффузии — с относительно высоким сопротивлением, потому что в базовой области концентрация носителей значительно меньше. Значения диффузионных резисторов от 10 Ом до 50 кОм.

На рис. 1.24 изображен резистор, сформированный в процессе базовой диффузии.

Конденсаторы ИМС, так же как и диоды, специально не изготавливают. Для их формирования, так же как и в варикапах, используется барьерная емкость p-n-переходов, которые формируются в островках одновременно с формированием транзисторов. Возможны три варианта формирования конденсаторов. Наибольшую удельную емкость конденсатора обеспечивает использование перехода эмиттер—база (порядка 1500 пФ/мм^2), однако этот p-n-переход обладает наименьшим среди всех пробивным напряжением (единицы вольт). Использование перехода коллектор—база позволяет получить конденсатор, удельная емкость которого в 5—6 раз меньше, чем у конденсатора на основе перехода база—эмиттер, а пробивное напряжение

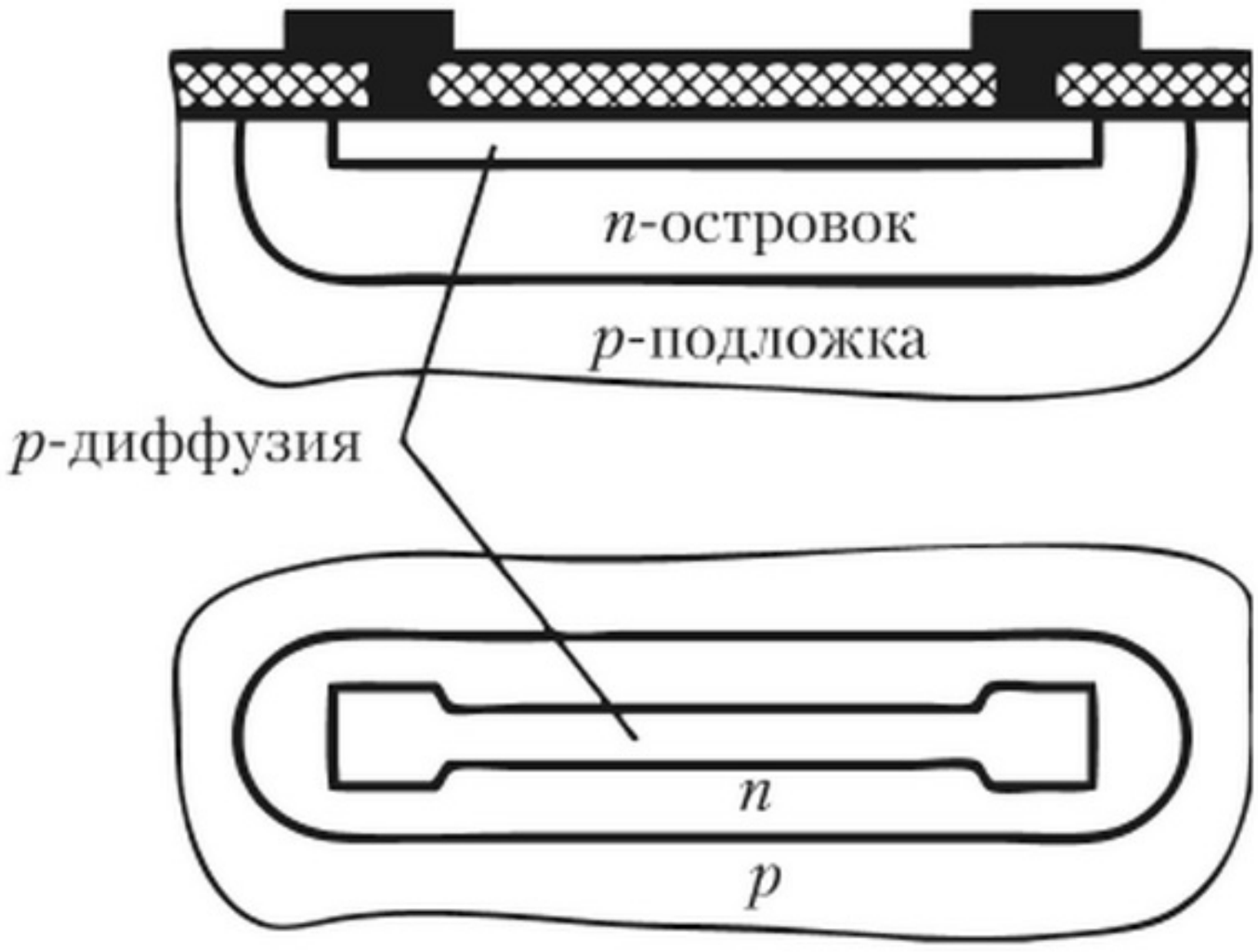


Рис. 1.24. **Резистор ИМС**

примерно во столько же раз больше. Последний вариант выполнения конденсатора заключается в использовании барьерной емкости, образуемой между подложкой кристалла и коллектором транзистора.

Поскольку барьерная емкость образуется только у запертого *p-n*-перехода, напряжение, приложенное к обкладкам конденсатора, должно быть запирающим, т.е. обратным для *p-n*-перехода, емкостью которого он образован.

Корпуса микросхем. Для защиты от воздействия внешних факторов и механических повреждений все микросхемы помещают в защитный корпус. ИМС размещаются, как правило, в монолитных корпусах с 14 или 16 выводами. Простейший и самый дешевый корпус — пластмассовый. Однако ввиду недостаточного теплоотвода в нем можно размещать лишь схемы невысокой степени интеграции с рассеиваемой мощностью до 200 мВт.

Микросхемы со средней и высокой степенью интеграции из-за большого числа активных элементов рассеивают большую мощность. Для их размещения необходимы корпуса, обеспечивающие хороший теплоотвод и защищающие их от перегрева. Поэтому для микросхем средней и высокой степени интеграции используют керамический и металло-керамический корпуса. Если необходимо более интенсивное охлаждение, могут использоваться радиаторы. Плата с размещенными на ней корпусами микросхем может также обдуваться вентилятором, расположенным внутри корпуса электронного устройства.

Поскольку БИС/СБИС значительно сложнее МИС и СИС, для их работы требуются гораздо большее число выводов и более сложные корпуса. Так, 16-разрядный микропроцессор Intel 8086 размещался в 40-контактном корпусе, а число контактов у микропроцессора Pentium 4 составляло уже 480. Для вывода электрических сигналов в корпусах современных СБИС используют специальные шариковые выводы, расположенные по периметру корпуса в несколько рядов. Количество контактов в таких корпусах находится в пределах от нескольких сот до двух тысяч. Причем новые модификации процессоров разрабатываются под серийно выпускаемые корпуса. Для подключения процессорных СБИС применяются специальные соединители — *сокеты*, к которым осуществляется механический прижим корпуса процессора. Для процессоров Sandy Bridge используется корпус и соответствующий сокет с 2011 контактами.

Современные СБИС рассеивают настолько большую мощность, что для их охлаждения используются специальные охлаждающие системы — кулеры, содержащие вентилятор, радиатор с теплоносителем и систему регулирования.

Контрольные вопросы и задания

1. Что такое электронная и дырочная проводимость?
2. Каким образом получают полупроводники *p*- и *n*-типов?
3. За счет чего в *p*-*n*-переходе образуется потенциальный барьер?
4. Что происходит в *p*-*n*-переходе при приложении к нему прямого и обратного напряжений?
5. Какие носители образуют прямой ток через диод и какие обратный? Объясните, почему.
6. Поясните разницу между электрическим пробоем и тепловым.
7. Объясните принцип работы стабилитрона.
8. Контакт каких материалов образует диод Шоттки?
9. Какие носители формируют ток в транзисторе *p*-*n*-*p*-типа и какие в транзисторе *n*-*p*-*n*-типа? Объясните, почему.
10. Что усиливает схема с общей базой: ток или напряжение? Объясните, почему.
11. Объясните, почему схема с общим эмиттером усиливает и ток, и напряжение.
12. В каком случае в схеме с общим эмиттером наступает режим насыщения, и в каком — отсечки?

Содержание

13. Почему схему с общим коллектором называют эмиттерным повторителем?

14. Почему у полевых транзисторов высокое входное сопротивление?

15. Почему у транзистора с индуцированным каналом управляющее напряжение однополярное, а с встроенным каналом — как положительное, так и отрицательное?

16. Чем объяснить лавинообразное нарастание тока через тиристор?

17. Чем динистор отличается от тиристора?

18. Какой вид транзисторов используется в биполярных интегральных микросхемах для реализации логики?

19. Какие из полевых транзисторов шире всего применяются в интегральных микросхемах?

20. Как в интегральных микросхемах получают диоды, резисторы и конденсаторы?

Содержание

Глава 2
АНАЛОГОВАЯ СХЕМОТЕХНИКА

В результате изучения главы 2 студент должен:

знать

- основные виды усилителей и функциональных преобразователей;
- основные схемы включения операционных усилителей;
- назначение усилительно-преобразовательных устройств в электронной технике;
- основные типы вторичных источников питания;

уметь

- выбирать необходимую схему усилительного каскада в зависимости от требований к принципиальной электрической схеме;
- синтезировать схемы включения операционных усилителей в зависимости от требований к устройству;

владеть

- методикой расчета усилительно-преобразовательных устройств на основе операционных усилителей;
- методикой расчета выпрямительных устройств.

2.1. Усилители

2.1.1. Усилители и их место в электронных устройствах

Полупроводниковые электронные устройства делятся на два больших класса: аналоговые и цифровые (дискретные). В основе классификации лежит возможность изменения в устройстве электрического сигнала, несущего информацию. Если информационный сигнал изменяется непрерывно и может принимать произвольные значения в широком диапазоне, устройство является аналоговым, если же сигнал изменяется дискретно и может принимать только два фиксированных значения, соответствующие двум цифрам двоичной системы счисления — нулю и единице, то устройство относится к цифровым, или дискрет-

ным. В аналоговых устройствах сам электрический сигнал и его параметры — уровень, частота и фаза электрического колебания — несут информацию о физической величине. В цифровых устройствах информация о величине закодирована цифровым кодом, состоящим из множества двоичных разрядов, каждый из которых может принимать только одно из двух фиксированных значений, которым соответствуют два уровня напряжения (обычно они обеспечиваются открытым либо закрытым состоянием транзистора, работающего в ключевом режиме).

Информацию о различных физических величинах и контролируемых процессах получают с помощью датчиков, называемых также измерительными преобразователями. Эти устройства осуществляют преобразование измеряемой величины в пропорциональный ей электрический сигнал. Очень часто эти сигналы небольшие, измеряемые тысячными долями вольт. После передачи по электрическим, радиочастотным или оптическим каналам связи сигналы приходят сильно ослабленными, и для нормальной работы приемников информации с этими сигналами требуется их предварительное усиление. Также невелик уровень электрических сигналов, считываемых с носителей информации во всевозможных магнитных и оптических запоминающих устройствах. Таким образом, для нормальной работы различных электронных устройств и систем необходимо усиление слабых сигналов. Это относится не только к чисто аналоговым устройствам, но и к цифровым, так как первичным информационным сигналом все равно остается маломощный электрический сигнал, который должен быть усилен. Поэтому усилители являются одними из основных узлов различной аппаратуры в устройствах автоматики, вычислительной и информационно-измерительной техники.

Усилитель — это устройство, увеличивающее интенсивность входного сигнала, используя энергию источника питания. В зависимости от назначения различают усилители напряжения и мощности, усилители сигнала переменного и постоянного тока, усилители, предназначенные для усиления сигналов в разных диапазонах частот.

В аналоговых электронных устройствах усилители не только обеспечивают простое усиление входного сигнала. На их основе выполняются самые разнообразные

устройства функциональной обработки сигналов, а также различные генераторы электрических сигналов.

2.1.2. Основные параметры усилителей

Основными параметрами усилителей являются коэффициенты усиления по напряжению K_U, по току K_I, по мощности K_P, а также его входное и выходное сопротивления. Для оценки влияния входного и выходного сопротивлений усилителя рассмотрим эквивалентную схему усилителя, к входной цепи которого подключен источник сигнала, а к выходной — нагрузка (рис. 2.1).

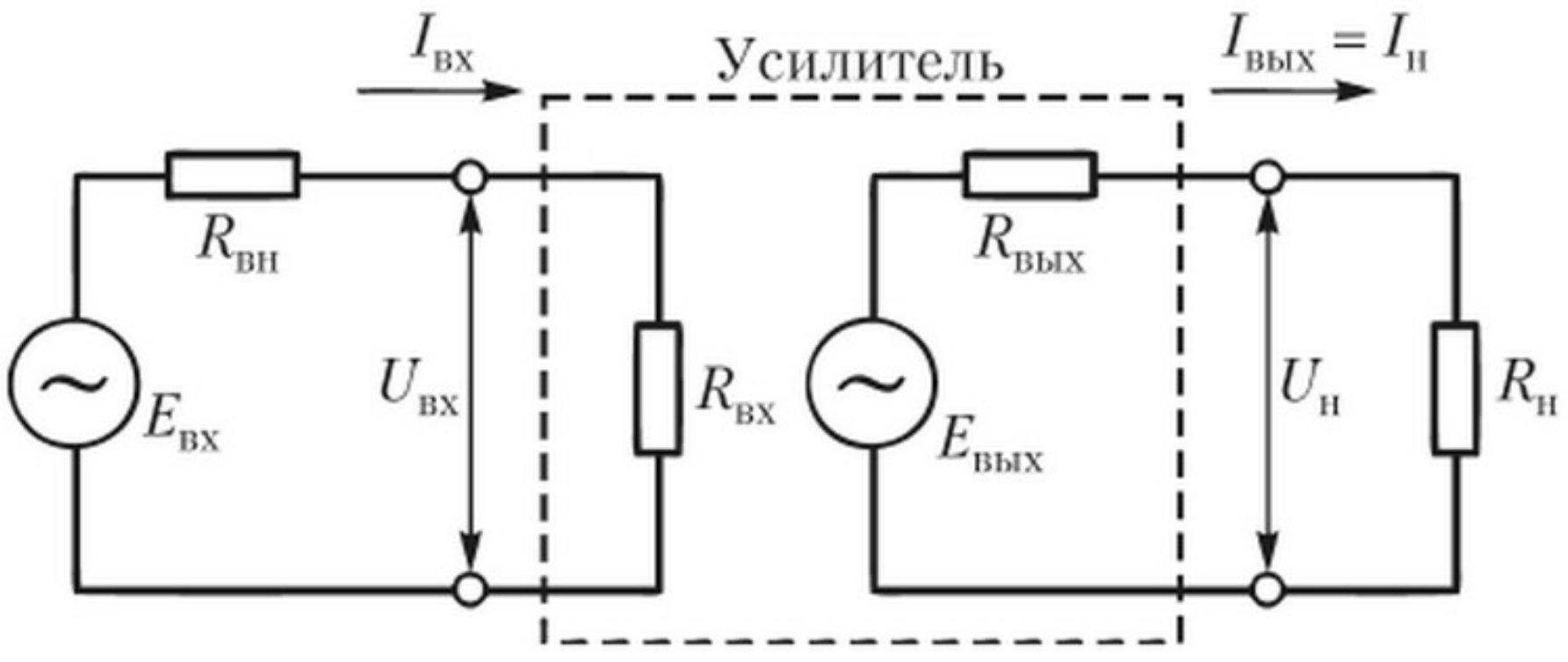


Рис. 2.1. **Эквивалентная схема усилителя**

Источник сигнала вырабатывает ЭДС $E_{вх}$ и имеет внутреннее сопротивление $R_{вн}$. Поскольку усилитель имеет входное сопротивление $R_{вх}$, во входной цепи течет ток

$$I_{вх} = \frac{E_{вх}}{R_{вн} + R_{вх}},$$

который создает на входе усилителя сигнал

$$U_{вх} = I_{вх} R_{вх} \quad E_{вх} \frac{R_{вх}}{R_{вн} + R_{вх}}.$$

Отсюда видно, что чем больше входное сопротивление усилителя, тем ближе значение ЭДС источника сигнала к напряжению, фактически приложенному к входу усилителя. По этой причине входное сопротивление усилителя стараются сделать по возможности большим, для чего во входных каскадах усилителей часто применяют полевые транзисторы.

На выходе усилитель создает сигнал $E_{вых} = U_{вх}K_U$, поступающий в нагрузку. Поскольку усилитель имеет выходное сопротивление, то ток в нагрузке равен

$$I_н = \frac{E_{вых}}{R_{вых} + R_н}.$$

Таким образом, напряжение на нагрузке будет отличаться от выходного напряжения усилителя и составит

$$U_н = I_н R_н \quad E_{вых} = \frac{R_н}{R_{вых} + R_н}.$$

Чем меньше выходное сопротивление усилителя, тем больше его КПД и мощность, отдаваемая нагрузке. Поэтому на выходе усилителей стремятся использовать каскады, обеспечивающие минимально возможное выходное сопротивление.

Частотные свойства усилителей определяет их *амплитудно-частотная характеристика*, т.е. зависимость коэффициента усиления от частоты сигнала.

По диапазону частот усиливаемых сигналов различают следующие классы усилителей:

- усилители низкой частоты — от десятков герц до десятков килогерц;
- усилители постоянного тока, которые по существу являются усилителями медленно меняющихся сигналов с диапазоном от нуля до сотен килогерц;
- избирательные, или полосовые, усилители, которые усиливают сигнал в очень узком диапазоне частот;
- широкополосные (импульсные) усилители с диапазоном от нескольких килогерц до нескольких мегагерц.

Для увеличения коэффициента усиления используют многокаскадные схемы, в которых выходной сигнал предыдущего каскада служит входным сигналом для последующего. Общий коэффициент усиления многокаскадного усилителя равен произведению коэффициентов усиления всех его каскадов.

Амплитудно-частотную характеристику усилителя рассмотрим на примере усилителя низкой частоты (рис. 2.2). Идеальная характеристика должна иметь одинаковый коэффициент усиления во всем диапазоне рабочих частот.

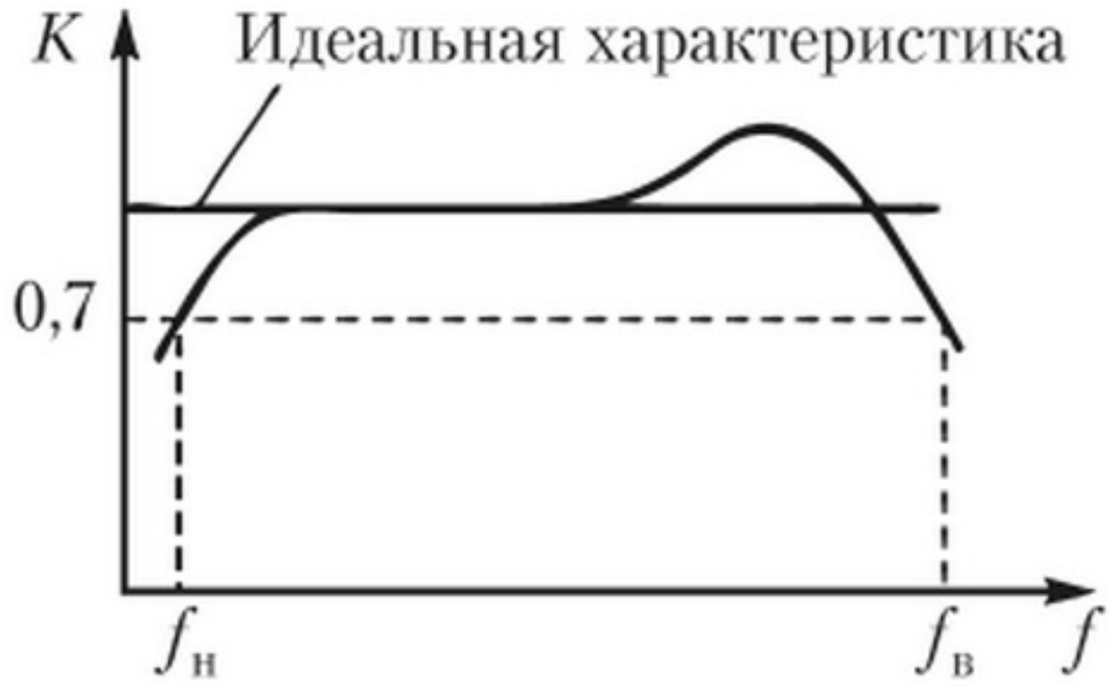


Рис. 2.2. **Амплитудно-частотная характеристика усилителя**

Реальная же характеристика имеет «завалы» на частотах, близких к границам диапазона рабочих частот, и «всплески» в середине диапазона.

Снижение K_U на низших частотах объясняется возрастанием емкостного сопротивления разделительных конденсаторов $X_C = 1/\omega C$ по мере снижения частоты сигнала и, следовательно, возрастанием потери напряжения на них.

Снижение K_U на высших частотах объясняется влиянием паразитных емкостей коллектор—база, коллектор—эмиттер и база—эмиттер, а также паразитных емкостей, которые возникают при монтаже. Эти емкости на высоких частотах приводят к закорачиванию транзисторов и снижению усиления сигнала. Всплески коэффициента усиления вызваны резонансными явлениями в усилителе.

Нижней $f_н$- и верхней $f_в$-граничными частотами, определяющими полосу частот усилителя, считают частоты, при которых коэффициент усиления снижается до 0,7 его значения на средних частотах.

Важным показателем качества усилителя является его способность сохранять форму кривой усиливаемого сигнала, изменяемую за счет *нелинейных искажений*. На рис. 2.3 в качестве примера показано, как искажается форма тока базы $I_Б$, т.е. входного тока транзистора, в схемах с ОЭ по сравнению с синусоидальной формой напряжения $U_{вх}$ за счет нелинейной входной характеристики транзистора. Максимальное значение тока в положительный полупериод I'_{max} больше, чем в отрицательный полупериод I''_{max}. В результате нелинейных искажений выходное напряжение содержит, кроме основной, еще и высшие гармонические составляющие. Степень искажения сигнала усилителем оценивается коэффициентом нелинейных искажений. Для уменьшения искажений важно, чтобы усилитель работал

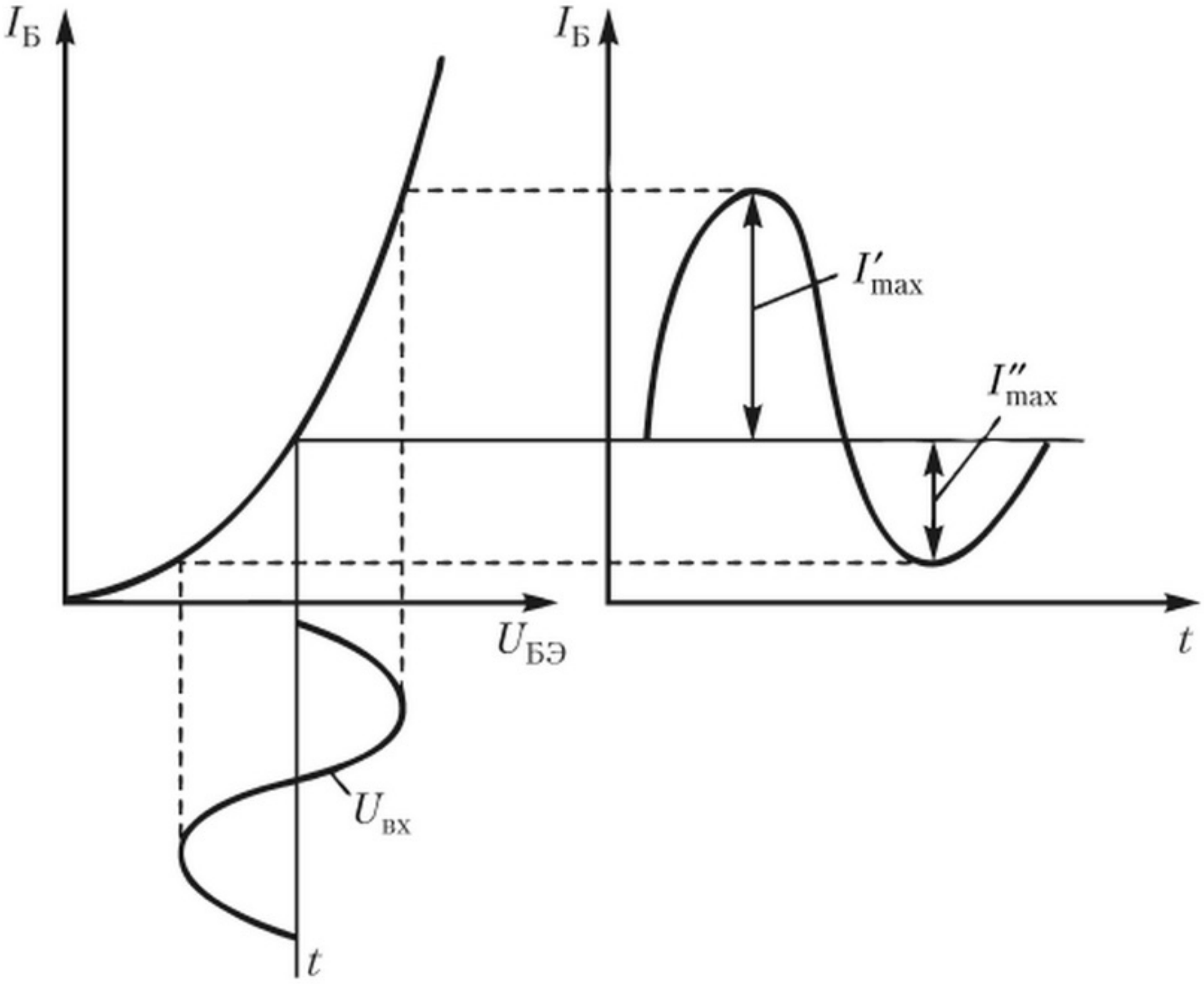


Рис. 2.3. **Возникновение нелинейных искажений**

на линейном участке входной характеристики и оставался на горизонтальных участках выходных характеристик транзисторов.

2.2. Каскады усилителей низкой частоты

2.2.1. Каскады на биполярных транзисторах

В усилителях на биполярных транзисторах используют, как правило, схему включения с общим эмиттером, обеспечивающую усиление как по напряжению, так и по току (рис. 2.4). В схеме резисторы $R'_Б$ и $R''_Б$, включенные между корпусом и точкой $+E_К$, образуют делитель для напряжения питания, которое фиксирует режим работы транзистора — его рабочую точку p' (рис. 2.5). Ток $I_д$, текущий через $R''_Б$, должен создавать падение напряжения, соответствующее напряжению между базой и эмиттером транзистора $U_{БЭр}$, поэтому

$$R''_Б = \frac{U_{БЭр}}{I_д},$$

где $I_д$ — ток делителя, образованного резисторами $R'_Б$ и $R''_Б$.

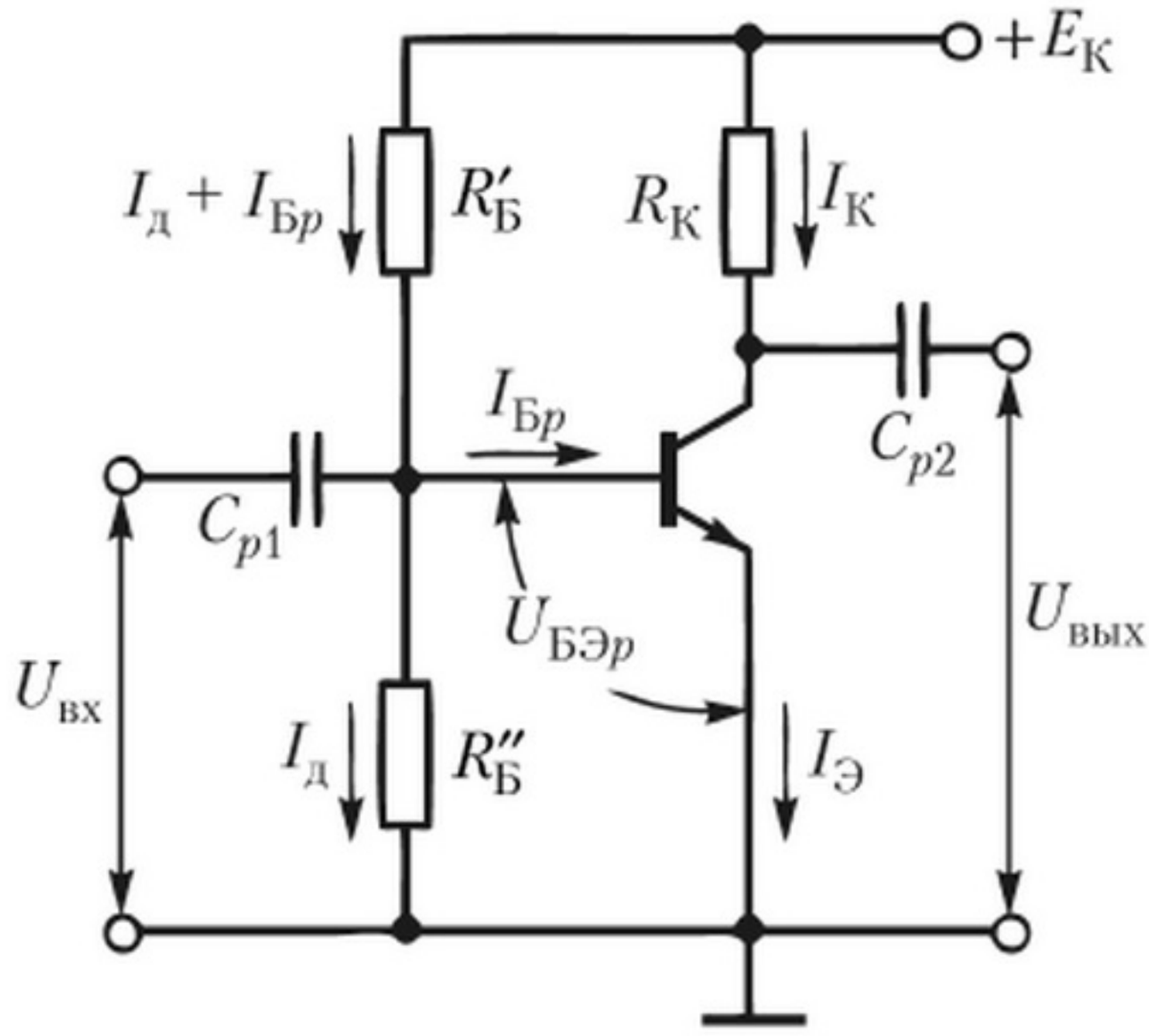


Рис. 2.4. **Усилительный каскад с общим эмиттером**

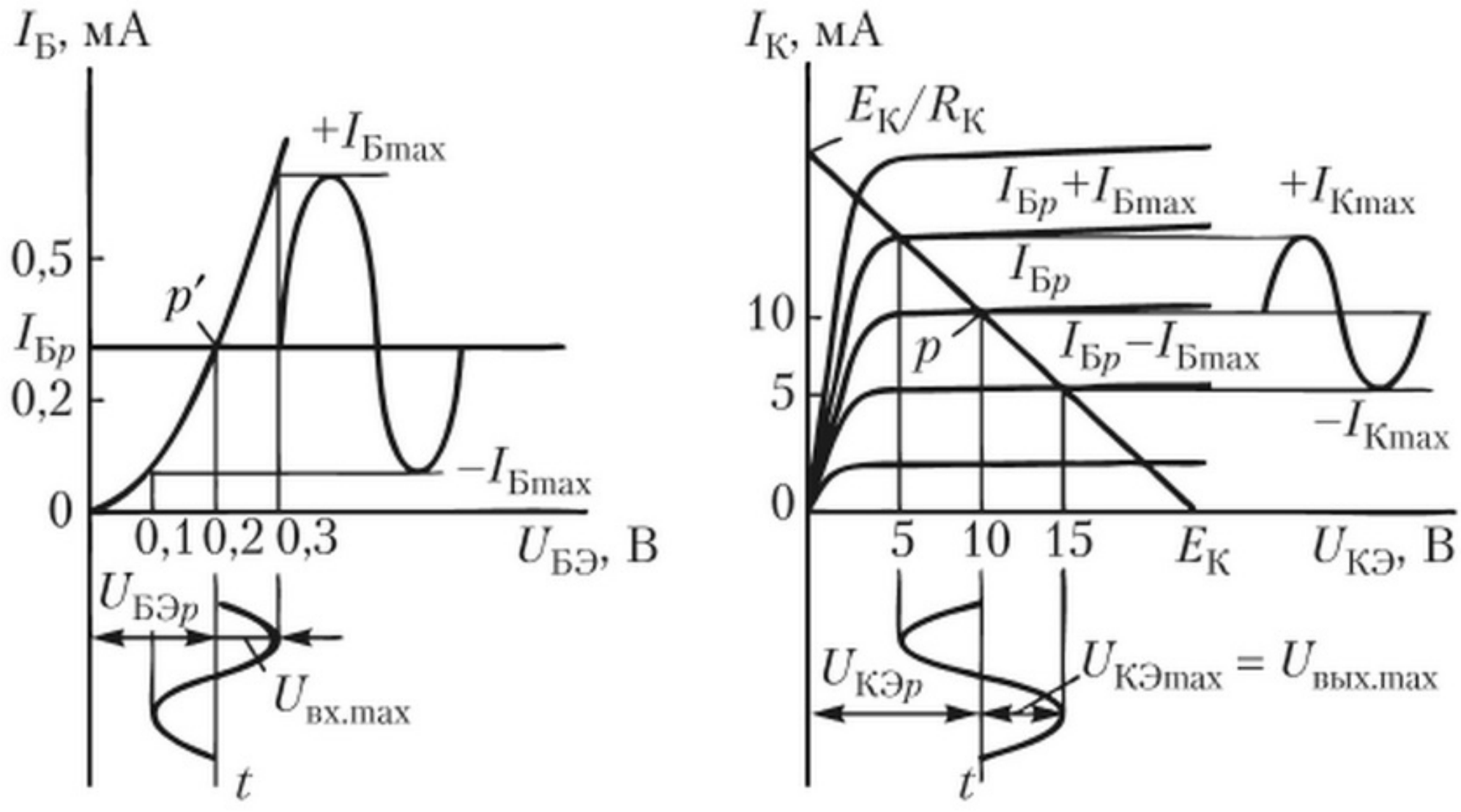


Рис. 2.5. **Влияние положения рабочей точки *p* на работу усилителя**

Через сопротивление $R'_Б$ течет ток I_Σ, равный сумме тока $I_д$ и тока, текущего в базу транзистора, поэтому

$$R'_Б = \frac{E_К - U_{БЭр}}{I_д + I_{Бр}}.$$

Ток делителя выбирают $I_д \approx (2...5)I_{Бр}$. Чем больше $I_д$, тем стабильнее работает каскад, так как изменения токов

коллектора $I_К$ и эмиттера $I_Э$, а значит, и тока базы $I_Б = I_К - I_Э$ не приведут к значительному изменению напряжения на $R'_Б$:

$$U_{R'_Б} = (I_д \quad I_Б) R'_Б.$$

Таким образом, напряжение $U_{БЭр}$ изменится незначительно. В то же время не следует выбирать ток делителя слишком большим, потому что это снижает КПД каскада из-за потерь энергии в делителе.

Допустим сначала, что $U_{вх} = 0$. Под действием напряжения $U_{БЭр}$ через открытый p-n-переход эмиттер—база протекает постоянный ток базы $I_{Бр}$. Разделительный конденсатор $C_{р1}$ не дает возможности постоянному току протекать через источник входного сигнала.

Транзистор открыт и находится в активной области. Его состояние определит точка p пересечения нагрузочной прямой, проведенной через точки $E_К$ и $E_К/R_К$, отсекаемые на осях (см. рис. 2.5), с характеристикой, соответствующей току $I_{Бр}$. Постоянный коллекторный ток $I_{Кр}$, соответствующий точке p, определит исходное напряжение между эмиттером и коллектором $U_{КЭр}$. Так как через разделительный конденсатор $C_{р2}$ постоянное напряжение не проходит, выходное напряжение $U_{вых} = 0$. Рассмотренное состояние схемы называют *режимом работы по постоянному току*.

Пусть теперь на вход схемы поступает сигнал в виде синусоидального напряжения с амплитудой $U_{вх.max}$. Этот сигнал уже пройдет через разделительный конденсатор и вызовет изменение управляющего напряжения $U_{БЭ}$. Под его действием произойдет изменение токов базы, эмиттера и коллектора. Изменение тока базы оценим по входной характеристике $I_Б = f(U_{БЭ})$ и определим амплитудные значения тока $+I_{Бmax}$ и $-I_{Бmax}$. Если бы входная характеристика была линейной, то изменения тока базы как в большую, так и в меньшую сторону были бы одинаковы, но из-за нелинейности характеристики амплитуда $+I_{Бmax}$ больше, чем $-I_{Бmax}$. Поскольку выходные характеристики $I_К = f(U_{КЭ})$ строят для разных токов базы, по ним можно определить токи коллектора, соответствующие токам $+I_{Бmax}$ и $-I_{Бmax}$. Изменения тока коллектора относительно среднего значения $I_{Кр}$ от $(I_{Кр} + I_{Кmax})$ до $(I_{Кр} - I_{Кmax})$ приведут к колебаниям напряжения на сопротивлении $R_К$ и, следовательно,

на коллекторе транзистора. Эти колебания легко оценить с помощью нагрузочной прямой. Действительно, рабочая точка p будет перемещаться по нагрузочной прямой между точками пересечения этой прямой с выходными характеристиками, соответствующими токам базы ($I_{Бp} + I_{Бmax}$) и ($I_{Бp} - I_{Бmax}$). Таким образом, колебания входного сигнала привели к пропорциональным колебаниям напряжения коллектор—эмиттер $U_{КЭ}$ с амплитудой $U_{КЭmax} = I_{Кmax}R_К$. Через конденсатор $C_{р2}$ эти колебания поступают на выход усилителя. Выходной сигнал, таким образом, равен

$$U_{вых} = U_{КЭmax} = I_{Кmax}R_К.$$

Этот режим называют режимом работы *по переменному току*.

Из приведенных на характеристиках построений видно, что $U_{вх.max} \approx 0{,}1$ В, $U_{вых.max} \approx 5$ В и, значит, коэффициент усиления по напряжению такого каскада

$$k = U_{вых.max}/U_{вх.max} = 5/0{,}1 = 50.$$

Следует обратить внимание, что положительному полупериоду входного напряжения (когда $U_{БЭp} + U_{вх.max}$) соответствует отрицательный полупериод выходного напряжения (т.е. $U_{КЭp} - U_{вых.max}$). Иначе говоря, между входным и выходным напряжениями существует сдвиг фаз, равный 180°. Для получения наименьших искажений усиливаемого сигнала рабочую точку p следует располагать на середине линейного участка входной характеристики.

2.2.2. Отрицательная обратная связь в усилителях

Транзисторы, будучи полупроводниковыми приборами, имеют два существенных недостатка. Первый связан с уже отмеченной принципиально нелинейной зависимостью тока коллектора от управляющего напряжения. Второй недостаток заключается в том, что характеристики транзистора сильно зависят от температуры. В частности, с ростом температуры возрастает неуправляемый ток коллектора $I_{К0}$, связанный с генерацией пар «электрон—дырка» (он увеличивается в два раза при повышении температуры на каждые 10 °С). В зависимости от температуры изменяется коэффициент передачи тока транзистора β. Это приводит к уходу рабочей точки p от своего первоначального положения, изменению коэффициента усиления и опять же к нелинейным искажениям выходного напряжения.

Существенное снижение нелинейных искажений обеспечивает использование в усилителях *отрицательной обратной связи*. При этом часть выходного сигнала подается обратно на вход, с тем чтобы противодействовать входному сигналу. Вследствие этого, естественно, уменьшается усиление. Однако с помощью отрицательной обратной связи можно добиться, чтобы усиление практически не зависело от нелинейной передаточной характеристики транзистора и в основном определялось соотношением омических сопротивлений.

Схема, приведенная на рис. 2.6, использует отрицательную обратную связь по току. Она обеспечивает компенсацию нелинейных искажений и термостабилизацию рабочей точки. Элементом, обеспечивающим эту связь, является резистор в эмиттерной цепи $R_Э$. Пусть увеличение коллекторного тока вызвано увеличением напряжения $U_{БЭ}$. Это приведет к увеличению падения напряжения на $R_Э$. Поскольку это напряжение приложено к эмиттеру транзистора, его рост приводит к уменьшению напряжения между базой и эмиттером транзистора и противодействует, таким образом, усилению. Следовательно, введение резистора $R_Э$ обеспечивает отрицательную обратную связь (ООС). Поскольку она вызвана протеканием эмиттерного тока, ее называют последовательной ООС по току.

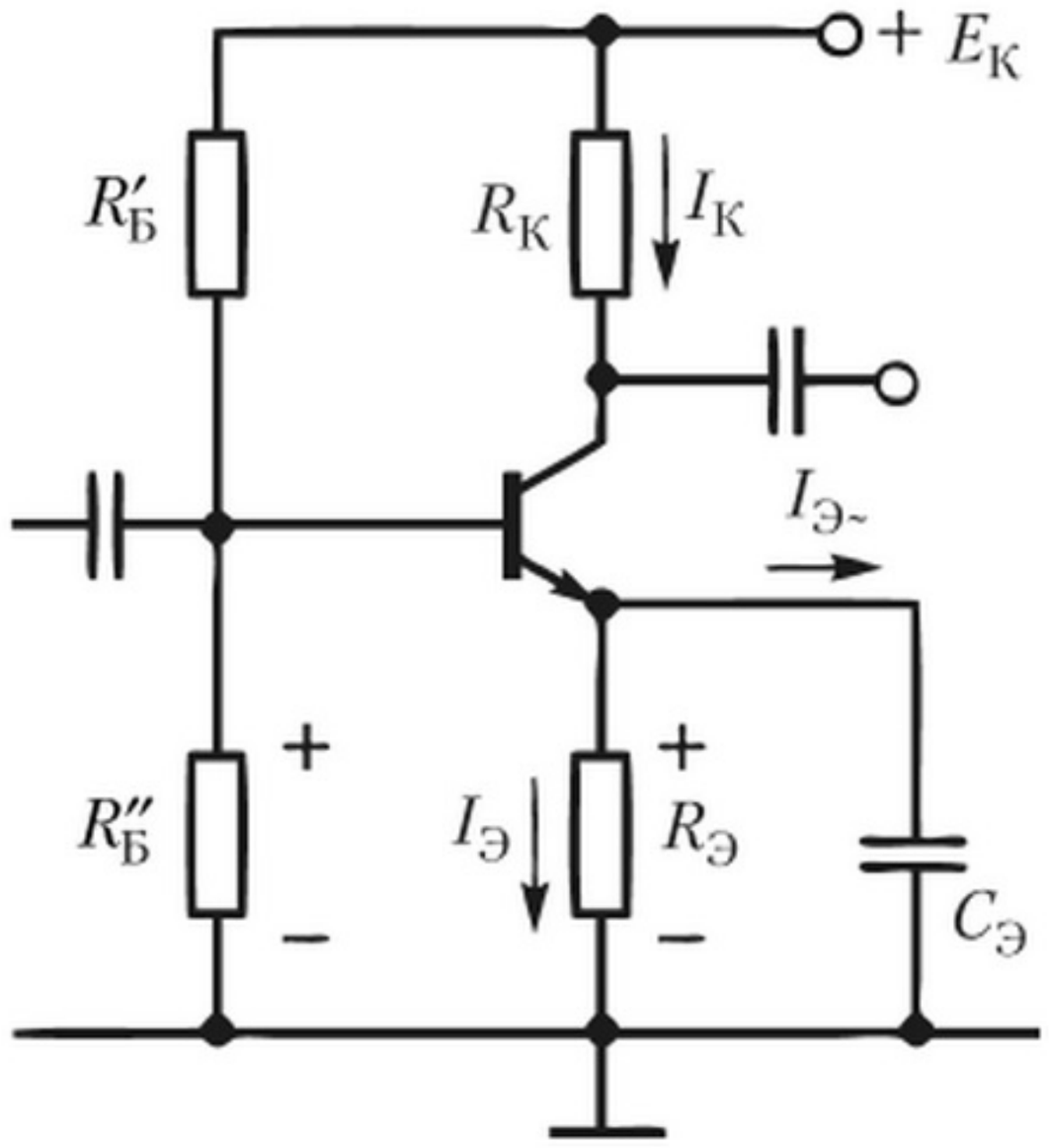


Рис. 2.6. **Схема с отрицательной обратной связью по току**

Содержание

Приближенно можно считать, что приращение напряжения на $R''_{Б}$ равно приращению напряжения на $R_{Э}$ (см. работу эмиттерного повторителя):

$$\Delta U_{R''_{Б}} = U_{R_{Э}}.$$

В связи с тем, что через $R_{К}$ протекает практически тот же ток, что и через $R_{Э}$, то и изменение падения напряжения $\Delta U_{R_{К}}$ на сопротивлении $R_{К}$ будет больше, чем соответствующее изменение $\Delta U_{R_{Э}}$ на сопротивлении $R_{Э}$ в $R_{К}/R_{Э}$ раз. Следовательно, коэффициент усиления по напряжению схемы с ООС приближенно определяется как

$$K_U = \frac{\Delta U_{R_{К}}}{\Delta U_{R''_{Б}}} \approx \frac{R_{К}}{R_{Э}}.$$

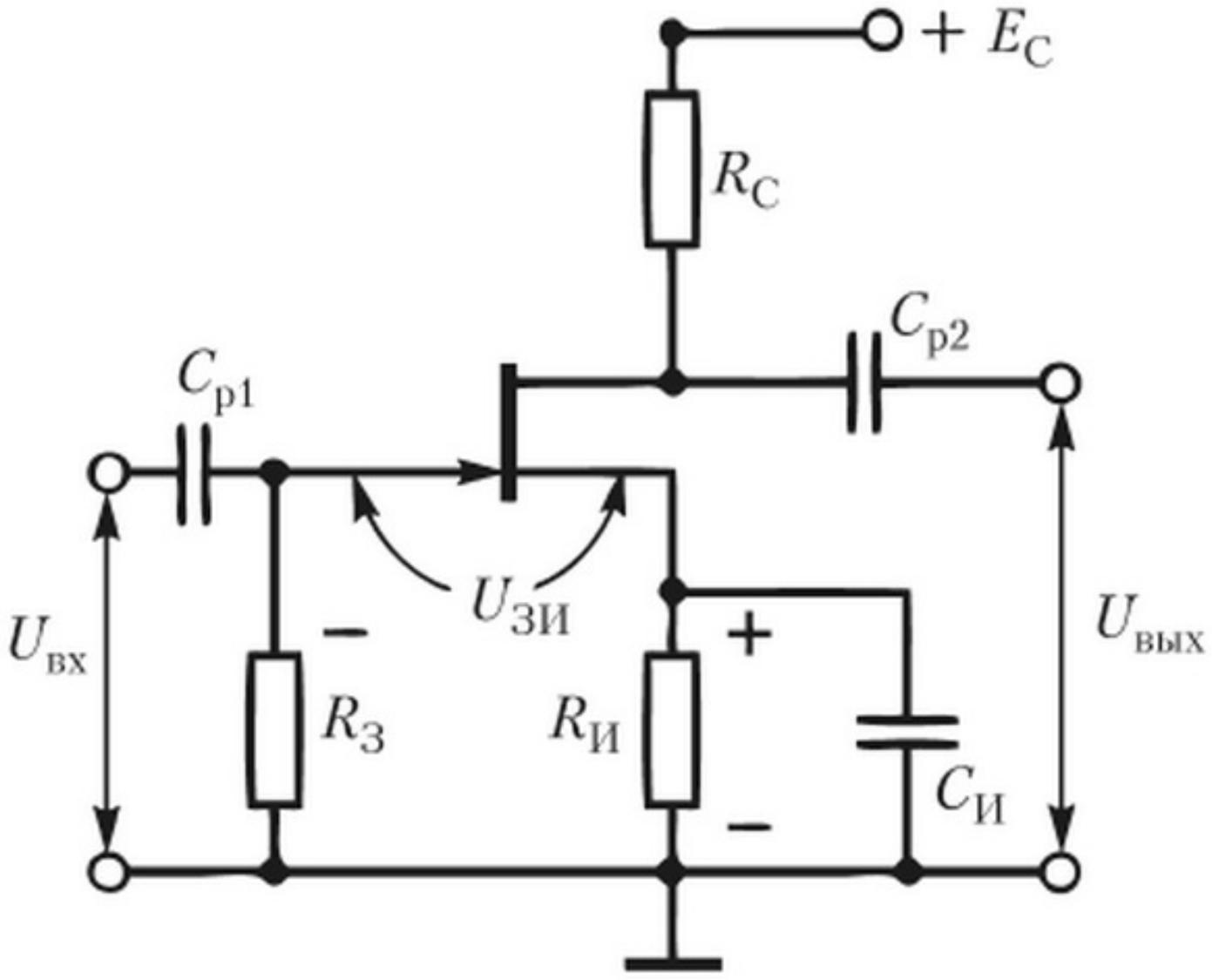
Как видим, коэффициент усиления уже не зависит от параметров транзистора, а определяется только соотношением омических сопротивлений $R_{К}$ и $R_{Э}$.

Положим теперь, что увеличение тока коллектора вызвано повышением температуры. Поскольку потенциал на базе транзистора при этом не изменяется, приращение $\Delta U_{R_{Э}}$ приведет к уменьшению управляющего напряжения $U_{БЭ}$ и снижению тока базы, а следовательно, и тока коллектора. Ток коллектора, таким образом, стабилизируется.

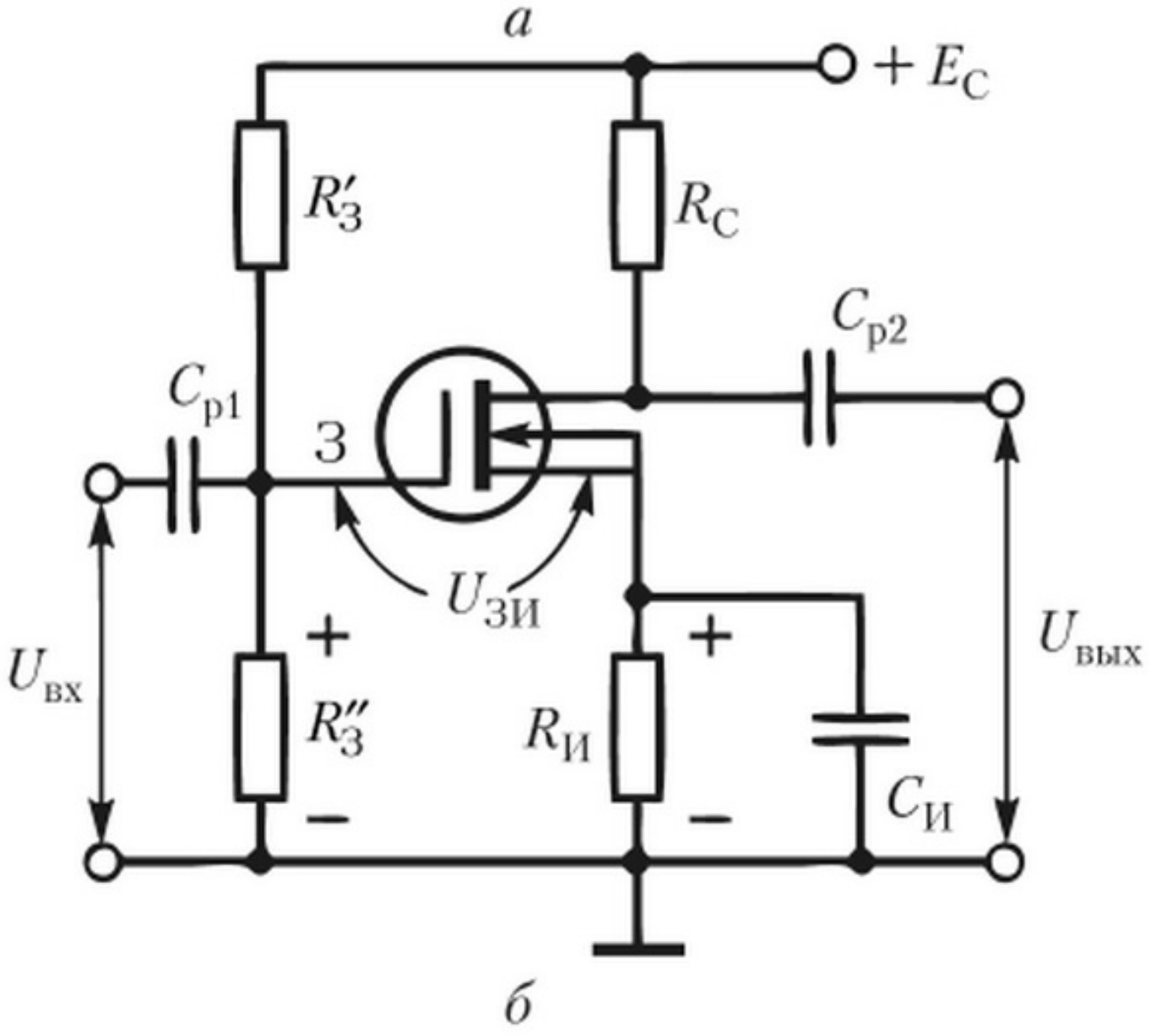
Для отвода от резистора $R_{Э}$ переменной составляющей тока эмиттера $I_{Э}$~ включают шунтирующий конденсатор $C_{Э}$ достаточно большой емкости (десятки микрофарад), сопротивление которого $X_C = 1/2\pi f C_{Э}$ для переменного тока составляет небольшую величину.

2.2.3. Усилительные каскады на полевых транзисторах

Из трех возможных схем включения полевых транзисторов (с общим затвором, общим истоком и общим стоком) наиболее распространены усилительные каскады с общим истоком (рис. 2.7), которые являются аналогом каскадов с общим эмиттером. Резистор $R_{С}$ играет роль нагрузки транзистора, разделительные конденсаторы $C_{р1}$ и $C_{р2}$ выполняют функции, не отличающиеся от их функций в каскадах на биполярных транзисторах.



а



б

Рис. 2.7. **Схемы усилительных каскадов на полевых транзисторах:**

а — на транзисторе с *p-n*-затвором; *б* — на транзисторе с индуцированным каналом

Входное напряжение в усилительных каскадах на полевых транзисторах с *p-n*-затвором (см. рис. 2.7, *а*) прикладывается между затвором и истоком, а на транзисторах с изолированным затвором (см. рис. 2.7, *б*) — между затвором и подложкой, которая обычно соединяется с истоком (или со стоком). Входной ток для транзисторов с *p-n*-затвором не превышает 10^{-8}А, а для транзисторов с изолирован-

ным затвором на несколько порядков меньше. У каскадов на транзисторах с p-n-затвором входное сопротивление на низких частотах составляет десятки мегаом, а у каскадов на МДП- (МОП-) транзисторах достигает $10^{12}...10^{15}$ Ом. Однако с повышением частоты входное сопротивление существенно уменьшается из-за протекания токов перезарядки паразитных емкостей затвор—исток и затвор—сток.

Подача синусоидального входного сигнала изменяет по гармоническому закону значение тока стока I_C согласно стокозатворным характеристикам транзисторов (см. рис. 1.15, *д* и 1.17, *г*) относительно исходной точки (подобной точкам p' и p на рис. 2.5). Изменения падения напряжения от тока I_C на резисторе R_C во много раз превосходят напряжения $U_{вх}$. Переменная составляющая этого падения напряжения через конденсатор C_{p2} поступает на выход каскада, как и в каскаде на биполярных транзисторах.

Что же касается смещения, определяющего положение исходной рабочей точки, то в каскадах на полевых транзисторах имеются свои особенности в отличие от биполярных. У транзисторов с p-n-затвором и с встроенным каналом смещение может быть обеспечено за счет падения напряжения на сопротивлении в цепи истока $R_И$ от начального тока I_{C0}, протекающего даже при $U_{ЗИ} = 0$. Для подачи потенциала смещения, отмеченного знаком «минус» у резистора $R_И$, к затвору достаточно подключить резистор $R_З$ (см. рис. 2.7, *а*). Так как ток затвора у полевых транзисторов ничтожно мал, падение напряжения от этого тока на резисторе $R_З$ практически равно нулю (даже если его сопротивление составляет десятки мегаом), и можно считать, что смещение $U_{ЗИ} \approx I_C R_И$.

У полевых транзисторов с индуцированным каналом обеспечить смещение рассмотренным способом нельзя, потому что при $U_{ЗИ} = 0$ они заперты. Поэтому в каскадах на полевых транзисторах с индуцированными каналами напряжение смещения на затвор подается с делителя напряжения на резисторах $R'_З$ и $R''_З$, подобно тому, как это выполняют в каскадах с биполярными транзисторами.

Температурные изменения тока стока (а значит, и смещения) в полевых транзисторах во много раз меньше изменений коллекторного тока у биполярных транзисторов. Поэтому обеспечение требуемой температурной стабильности не вызывает трудностей.

2.2.4. Дифференциальный каскад

В электронной технике часто требуются усилители сигналов, скорость изменения которых очень низка, например с датчиков медленно изменяющихся сигналов, таких как термопара. Такие усилители называют усилителями постоянного тока. Наиболее распространенной схемой, на базе которой они создаются, является *дифференциальный каскад* (рис. 2.8, *а*).



Рис. 2.8. **Дифференциальный усилительный каскад (*а*) и передаточная динамическая характеристика транзисторов (*б*)**

В дифференциальной схеме два входа, два выхода и два источника питания. Предположим, что параметры левого и правого плеч схемы одинаковы. Положительный потенциал источника питания $E_{см}$, поступающий через корпус и цепи входных сигналов на базы обоих транзисторов, открывает их в равной степени. Рабочие точки *p* на рис. 2.8, *б* транзисторов *VT*1и *VT*2 совпадают. Поэтому при отсутствии входных сигналов под действием источника питания $E_К$ по резисторам R_1 и R_2 протекают одинаковые коллекторные токи $I_1 = I_2 = I_{Кp}$. Так как $R_1 = R_2$, то эти токи создают на них одинаковые падения напряжения, а значит, потенциалы точек *a* и *b* одинаковы. Если нагрузочное сопротивление включить между точками *a* и *b*, то выходное напряжение на нем равно нулю. Так можно выполнить требования усилителей постоянного тока о равенстве нулю

$U_{вых}$ при отсутствии сигнала $U_{вх}$. Такое состояние схемы называют *режимом покоя*.

Если пренебречь базовыми токами, то по резистору R_3 протекает сумма токов $I_1 + I_2$, создающая на нем падение напряжения. Напряжение, задающее исходные рабочие точки транзисторов, будет определяться разностью $E_{см}$ и напряжения на R_3:

$$U_{БЭр} = E_{см} - (I_1 + I_2)R_3.$$

Такое включение R_3 создает последовательную отрицательную обратную связь по току, стабилизируя исходные рабочие точки транзисторов. Любые одновременные изменения, например, увеличение токов I_1 и I_2, возникающие под действием изменений напряжения источника питания, температуры и т.д., вызовут увеличение падения напряжения на R_3 и, следовательно, такое уменьшение $U_{БЭ}$, которое стремится вернуть (снизить) коллекторные токи к исходному значению, т.е. стабилизировать их суммарное значение:

$$I_1 + I_2 \approx \text{const}.$$

Таким же образом дифференциальный каскад реагирует на *синфазные сигнал и помеху*, т.е. входные сигналы, которые одновременно (без сдвига фаз) и одинаково ($U_{вх1} = U_{вх2}$) действуют на оба входа, стремясь одновременно изменить I_1 и I_2. Обратная связь тем выше, чем больше R_3.

Совершенно по-иному реагирует дифференциальный каскад, если сигналы на входах каскада *противофазные* (например, $U_{вх1} = +U_{вх}$, а $U_{вх2} = -U_{вх}$). В этом случае (см. рис. 2.8, *б*) ток I_1 возрастет на ΔI, а I_2 уменьшится на ΔI, но их сумма останется неизменной.

Поэтому обратная связь не стремится уменьшить изменения токов и каждое плечо ведет себя как обычная схема ОЭ, т.е. потенциал точки a и $U_{вых1}$ понизятся, а потенциал точки b и $U_{вых2}$ повысятся. Каскад реагирует только на *разность* входных сигналов, почему и называется дифференциальным.

Рассмотрим еще один случай, когда напряжение $U_{вх}$ действует только на один из входов, например $U_{вх1} > 0$, а $U_{вх2} = 0$. В первый момент ток I_1 возрастает, допустим, на $+\Delta I$, а I_2 остается неизменным, при этом возрастет и сумма токов $(I_1 + \Delta I) + I_2$, но вступившая в действие обратная связь приведет к тому, что ток первого транзистора станет равным $I_1 + \Delta I/2$, а ток второго: $I_2 - \Delta I/2$. И в этом случае потен-

циал точки a понизится, а точки b повысится, но изменения $U_{вых}$ будут в 2 раза меньше, чем в предыдущем случае.

Обычно усилители постоянного тока имеют несколько каскадов, причем дифференциальный каскад является первым и у него используется только один выход. Если принять, что используется выход 2, то подача положительного сигнала на вход 1 приводит к увеличению выходного сигнала, а подача положительного сигнала на вход 2 — к уменьшению. Поэтому вход 1 называют прямым, или *неинвертирующим*, а вход 2 — обратным, или *инвертирующим*.

2.3. Выходные каскады усилителей

2.3.1. Режимы работы выходных каскадов усилителей

Выходной каскад предназначен для отдачи заданной мощности в нагрузку, сопротивление которой тоже задано. Так как мощность поступает от источника питания усилителя через выходной каскад, его КПД должен быть высоким, иначе устройство будет неэкономичным, а габаритные размеры (поверхность охлаждения) раздутыми для отвода выделяющейся в каскаде теплоты. Если у входных каскадов нелинейность транзистора не оказывает влияния ввиду малости усиливаемых сигналов, то у выходных каскадов диапазон изменения сигнала большой, и нелинейность транзистора необходимо учитывать. С этой целью строят так называемую передаточную характеристику. *Передаточная характеристика — это зависимость выходного тока каскада (тока коллектора или эмиттера) от входного напряжения.* В ней учитываются нелинейность входной и выходной характеристик транзистора и изменения напряжения, падающего на самом транзисторе в зависимости от выходного тока.

На семействе статических выходных характеристик транзистора (рис. 2.9, *а*) по точкам $E_К$ и $E_К/R_н$, отложенным на осях координат, проводят нагрузочную прямую. Точки пересечения этой прямой с характеристиками, соответствующими разным токам базы $I_{Б1}$, ..., $I_{Бi}$, ..., $I_{Бn}$, определят ряд значений коллекторного тока $I_{К1}$, ..., $I_{Кi}$, ..., $I_{Кn}$. На входной характеристике транзистора (рис. 2.9, *б*) находят ряд значений напряжения $U_{БЭ1}$, ..., $U_{БЭi}$, ..., $U_{БЭn}$, которые необходимо подать для получения соответствующих базовых токов. Наконец, по парам значений $I_{Кi}$ и $U_{БЭi}$ строят передаточную характеристику каскада, которая связывает выходной пара-

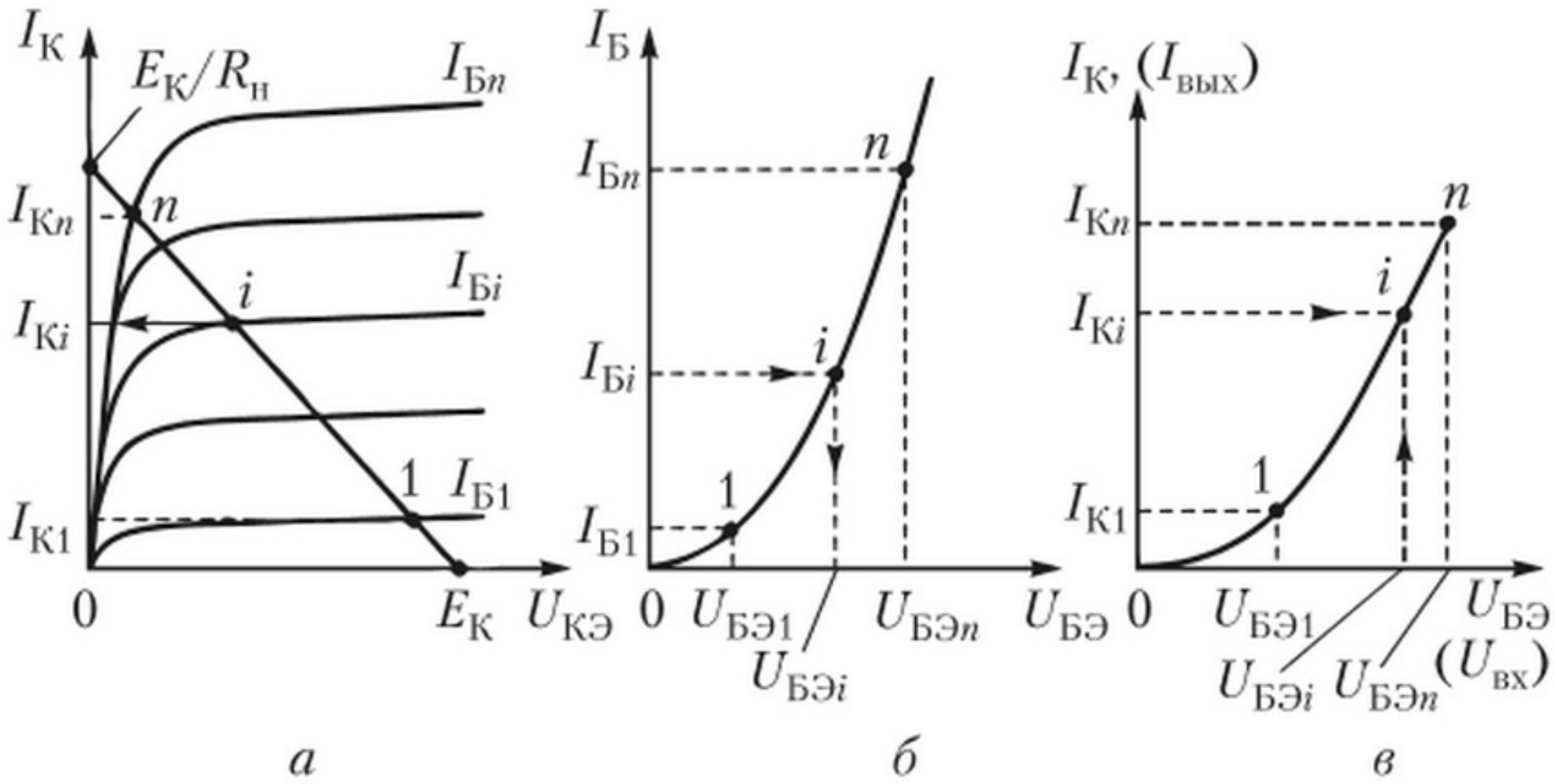


Рис. 2.9. **Построение передаточной характеристики (*в*) по выходной (*а*) и входной (*б*) характеристикам**

метр — ток на выходе каскада — с входным — напряжением сигнала на входе.

Возможны различные варианты выбора рабочего участка этой характеристики. Рассмотрим их подробнее.

Режим А — это режим, при котором исходная рабочая точка *p* (когда входной сигнал равен нулю) располагается примерно на середине линейного участка характеристики (рис. 2.10). В этом режиме в состоянии покоя через тран-

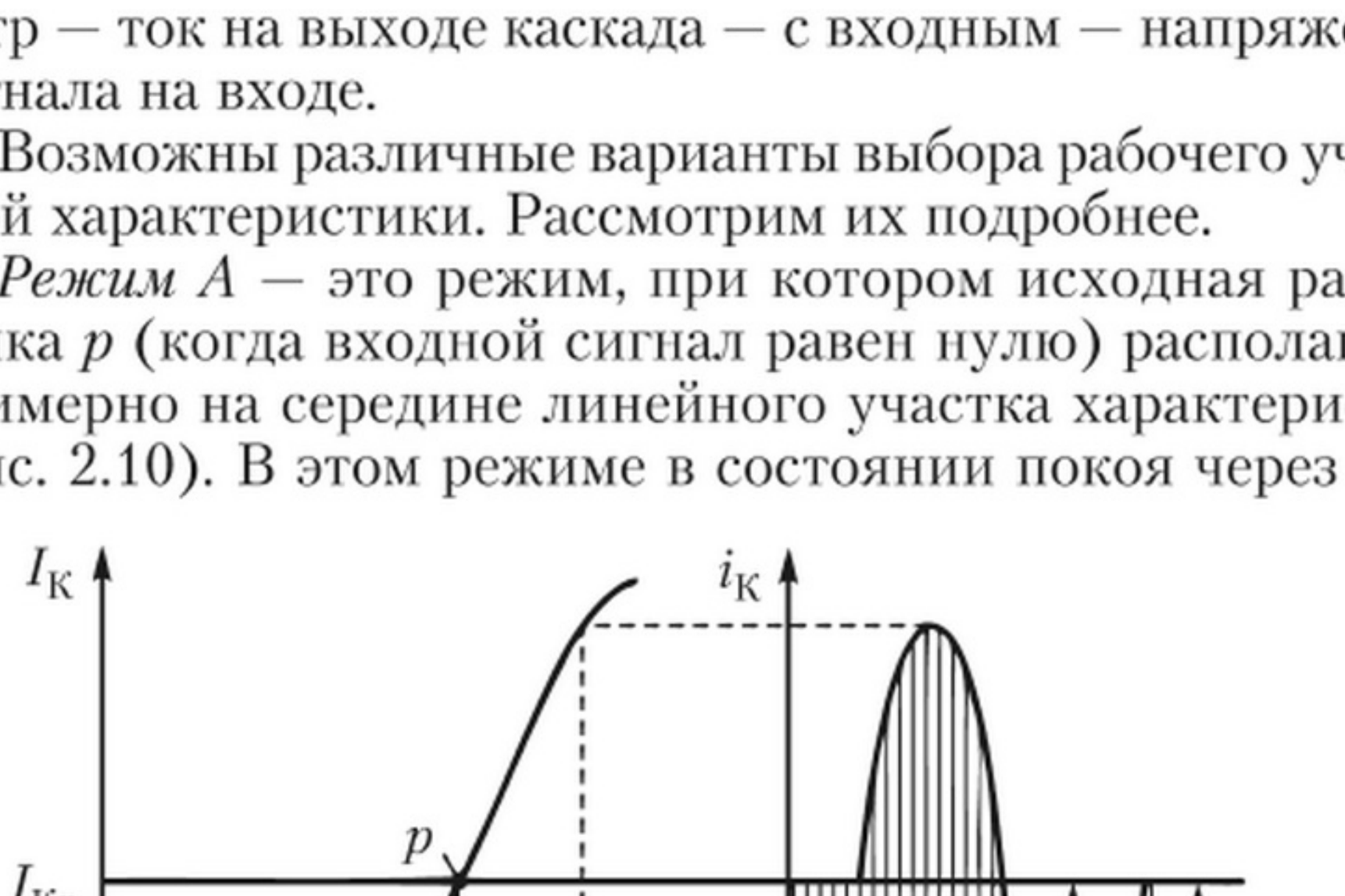
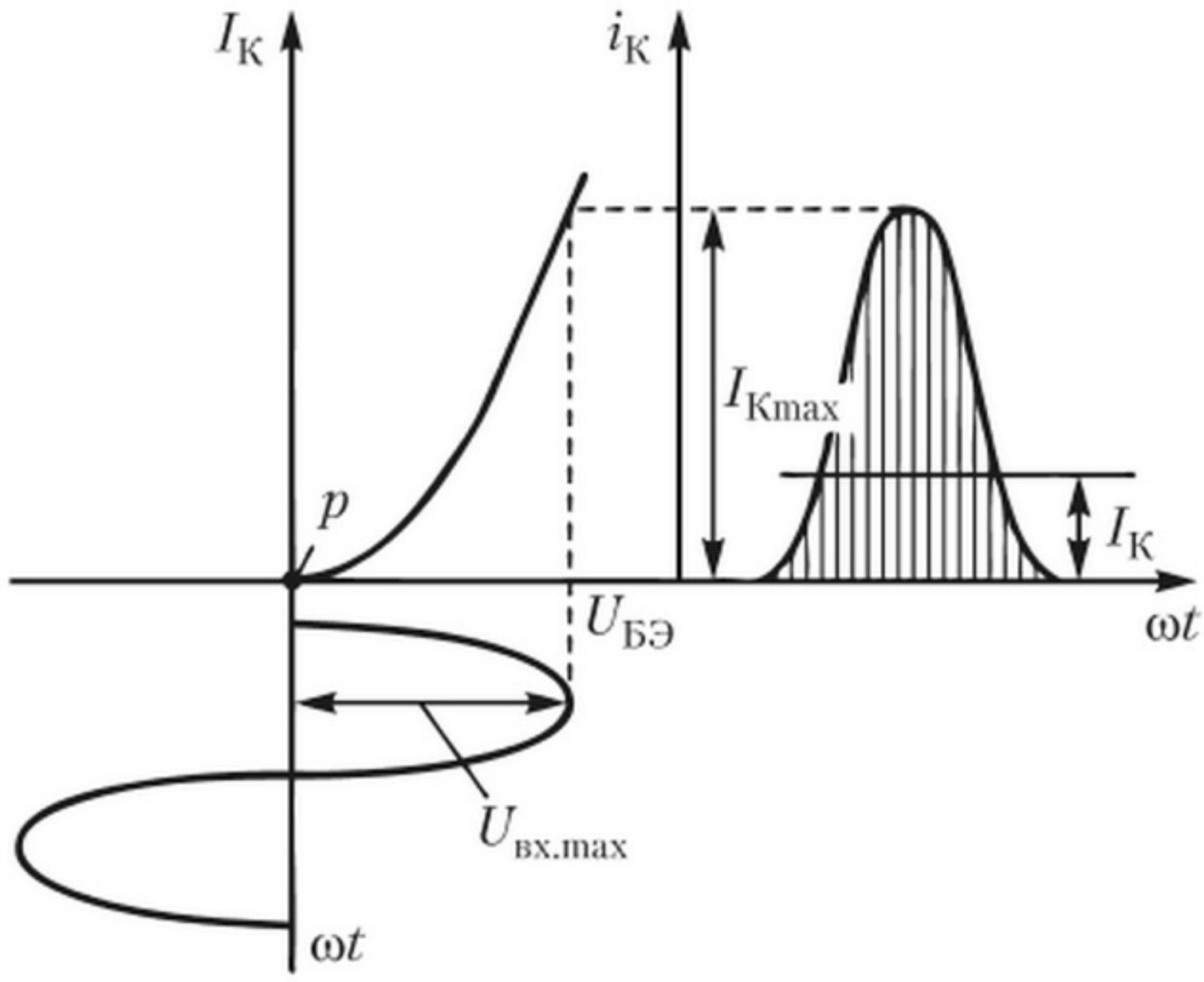
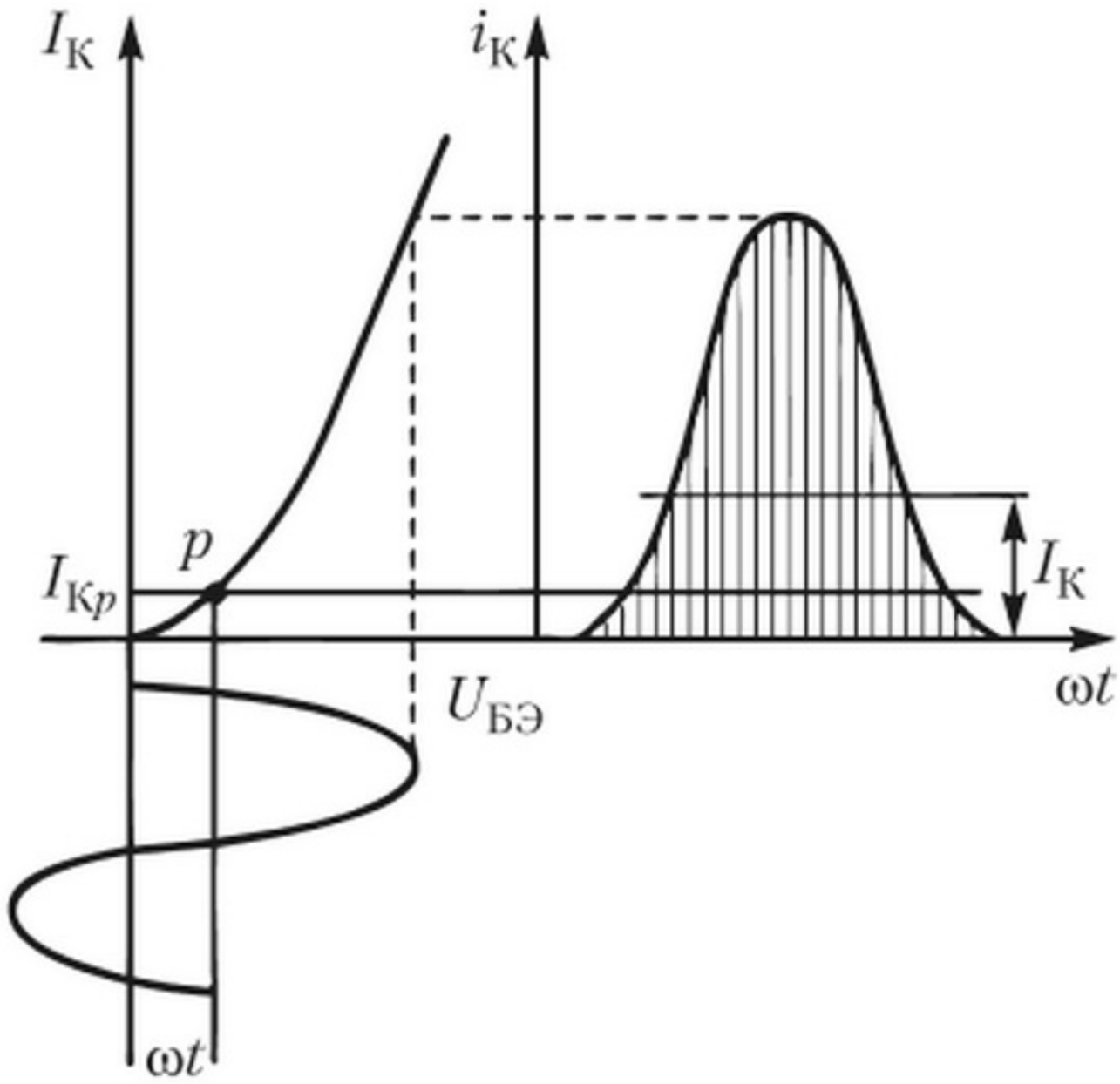


Рис. 2.10. **Режим *А* работы усилительного каскада**

Содержание



а



б

Рис. 2.11. **Режимы работы усилительного каскада:**
а — режим *B*; *б* — режим *AB*

зистор течет сравнительно большой постоянный ток $I_{Кp}$, а амплитуда переменной составляющей тока $I_{Кmax}$ меньше или равна этому току. При этом форма выходного сигнала повторяет форму входного и нелинейные искажения минимальны. Но КПД каскада составляет лишь 20—30%, потому

что полезная мощность определяется только переменной составляющей выходного тока, а потребляемая каскадом мощность — суммой переменной $I_{\text{Кmax}}$ и постоянной $I_{\text{К}p}$ составляющих выходного тока.

Режим B — это режим, при котором исходная рабочая точка совпадает с началом координат, т.е. в состоянии покоя выходной ток равен нулю (рис. 2.11, *а*).

При подаче на вход синусоидального сигнала ток в выходной цепи протекает лишь в течение половины периода и имеет форму импульсов. КПД каскада в этом режиме достигает 60—70%, так как постоянная составляющая $I_{\text{К}}$ коллекторного тока (определяемая по заштрихованной площади как среднее за период значение тока) значительно меньше, чем в режиме *A*. Однако форма усиливаемого сигнала слишком искажена.

Режим AB (рис. 2.11, *б*) занимает промежуточное положение. Такой режим позволяет уменьшить нелинейные искажения при применении двухтактных выходных каскадов.

2.3.2. Однотактные и двухтактные выходные каскады

Выходные каскады выполняют однотактными и двухтактными. В *однотактных* каскадах только один мощный усилительный транзистор, который работает как в положительный полупериод синусоиды, так и в отрицательный. В *двухтактных* каскадах — два мощных транзистора, которые работают по очереди.

Однотактный каскад. Схема однотактного выходного каскада аналогична схеме, изображенной на рис. 2.4. Нагрузка включается вместо резистора $R_{\text{К}}$, а разделительный конденсатор $C_{\text{р2}}$ отсутствует. Однотактный каскад, работающий в режиме *A*, обеспечивает наименьшие нелинейные искажения, но обладает рядом недостатков: низким КПД; невозможностью применения в режимах *B* и *AB* из-за больших нелинейных искажений в этих режимах. Из-за этих недостатков однотактные каскады применяют только при относительно небольших мощностях нагрузки.

Двухтактный каскад. Он позволяет избавиться от недостатков, присущих однотактному каскаду. Такие каскады выполняют на транзисторах, включенных по схемам с общим эмиттером или общим коллектором.

Обычно в предварительных каскадах усилителей обеспечивается необходимое усиление входного сигнала

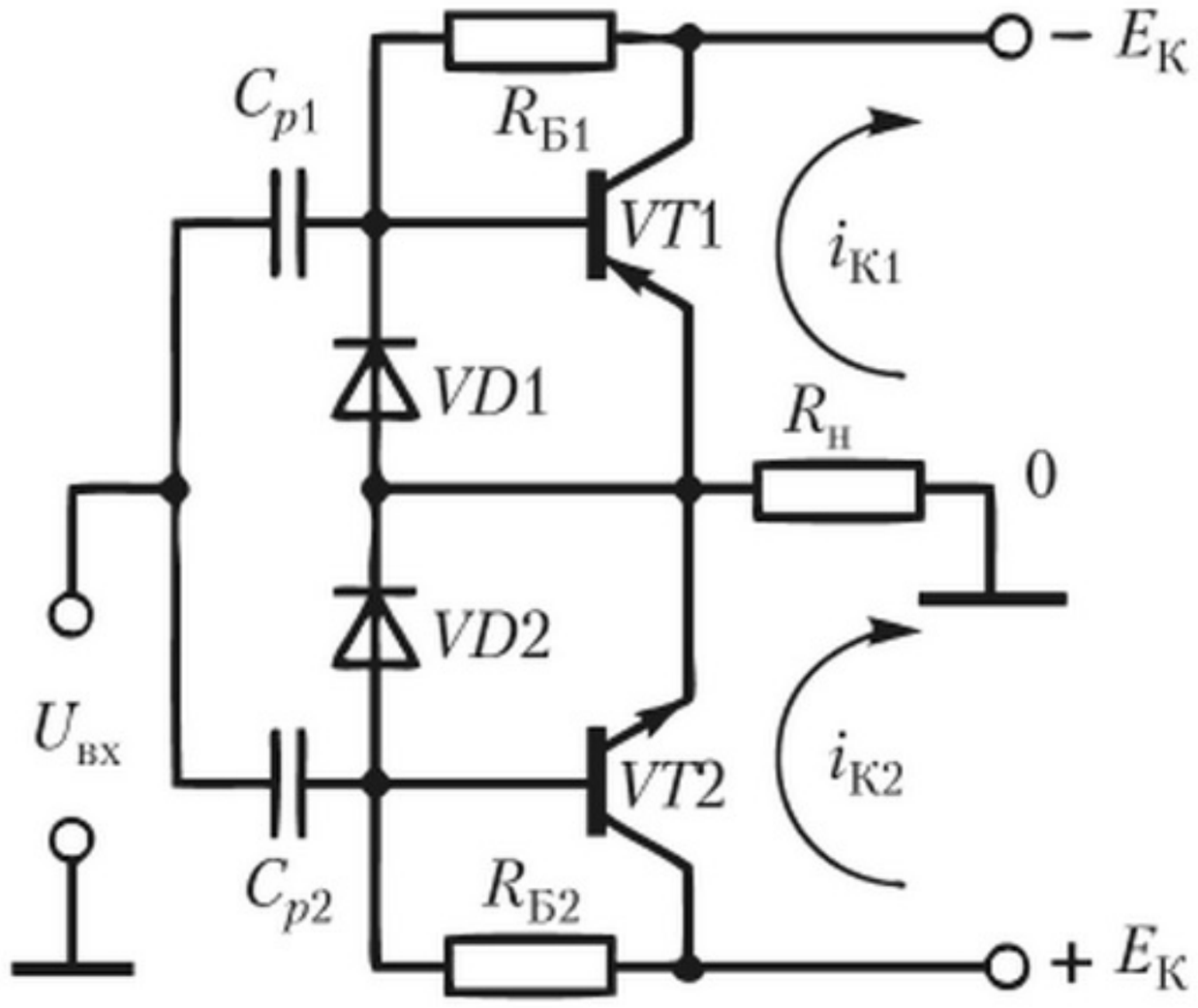


Рис. 2.12. **Двухтактный эмиттерный повторитель**

по напряжению, а в выходном каскаде происходит усиление по току, мощности и обеспечивается низкое выходное сопротивление. В этом случае часто в качестве выходного каскада используют двухтактный эмиттерный повторитель (рис. 2.12). Входной сигнал проходит через разделительные конденсаторы и поступает на базы транзисторов *VT*1 и *VT*2. Эти транзисторы разных типов проводимости, т.е. *VT*1 — типа *p-n-p*, а *VT*2 — типа *n-p-n*. Транзистор *VT*1 управляется положительным напряжением, а *VT*2 — отрицательным. Положительный полупериод синусоиды входного сигнала усиливается транзистором *VT*1. В это время транзистор *VT*2 закрыт и ток в нагрузку течет по цепи «корпус — $R_н$ — эмиттер *VT*1 — коллектор *VT*1 — $-E_К$». В отрицательный полупериод транзисторы меняются ролями и работает транзистор *VT*2, а *VT*1 закрыт. Ток в нагрузке течет по цепи «$+E_К$ — коллектор *VT*2 — эмиттер *VT*2 — $R_н$ — корпус».

Чтобы обеспечить положение рабочей точки транзисторов, необходимо установить в состоянии покоя напряжения смещения на базах транзисторов. Для этого используются цепочки «резистор $R_{Б1}$ — диод *VD*1» для транзистора *VT*1 и «резистор $R_{Б2}$ — диод *VD*2» для транзистора *VT*2. Протекающий в них ток обеспечивает необходимое напряжение смещения на база—эмиттерных переходах транзисторов.

Как видно, схему двухтактного эмиттерного повторителя можно разделить на две симметричные части — верхнюю и нижнюю, которые называются плечами каскада. Транзи-

сторы в данном каскаде работают в режиме *AB*. Хотя каждое плечо дает большое искажение синусоидального сигнала (только в одном полупериоде), вместе они формируют результирующий ток, имеющий синусоидальную форму. Режим *AB* в двухтактном эмиттерном повторителе обеспечивает низкие нелинейные искажения и высокий КПД — около 70%. Недостатком двухтактных каскадов является то, что параметры мощных транзисторов, используемых в разных плечах, должны иметь близкие характеристики.

2.4. Операционные усилители

По принципу действия операционный усилитель (ОУ) сходен с обычным. Как и обычный усилитель, он предназначен для усиления напряжения или мощности входного сигнала. Однако свойства и параметры обычного усилителя полностью определены его схемой, в то время как свойства и параметры ОУ определяются преимущественно параметрами цепи обратной связи.

Операционный усилитель обеспечивает усиление как медленно, так и быстро меняющихся сигналов, спектр частот которых может быть от нуля до нескольких мегагерц. Он имеет очень высокий коэффициент усиления по напряжению, высокое входное и низкое выходное сопротивления, очень низкий входной ток — доли микроампер и даже наноампер (10^{-9} А). Ранее подобные высококачественные усилители использовались исключительно в аналоговых вычислительных устройствах для выполнения таких математических операций, как суммирование и интегрирование. Отсюда и произошло их название — операционные усилители.

Операционные усилители выполняют в виде монолитных интегральных микросхем, и по своим размерам и цене они почти не отличаются от отдельно взятого транзистора. Благодаря практически идеальным характеристикам операционных усилителей реализация схем на их основе оказывается значительно проще, чем на отдельных транзисторах.

2.4.1. Свойства операционных усилителей

Обозначение ОУ на принципиальных электрических схемах в соответствии с ГОСТ приведено на рис. 2.13, *а*. Для пояснения принципа действия операционного усили-

теля и схем на его основе воспользуемся условным обозначением ОУ, применяемым ранее (рис. 2.13, *б*).

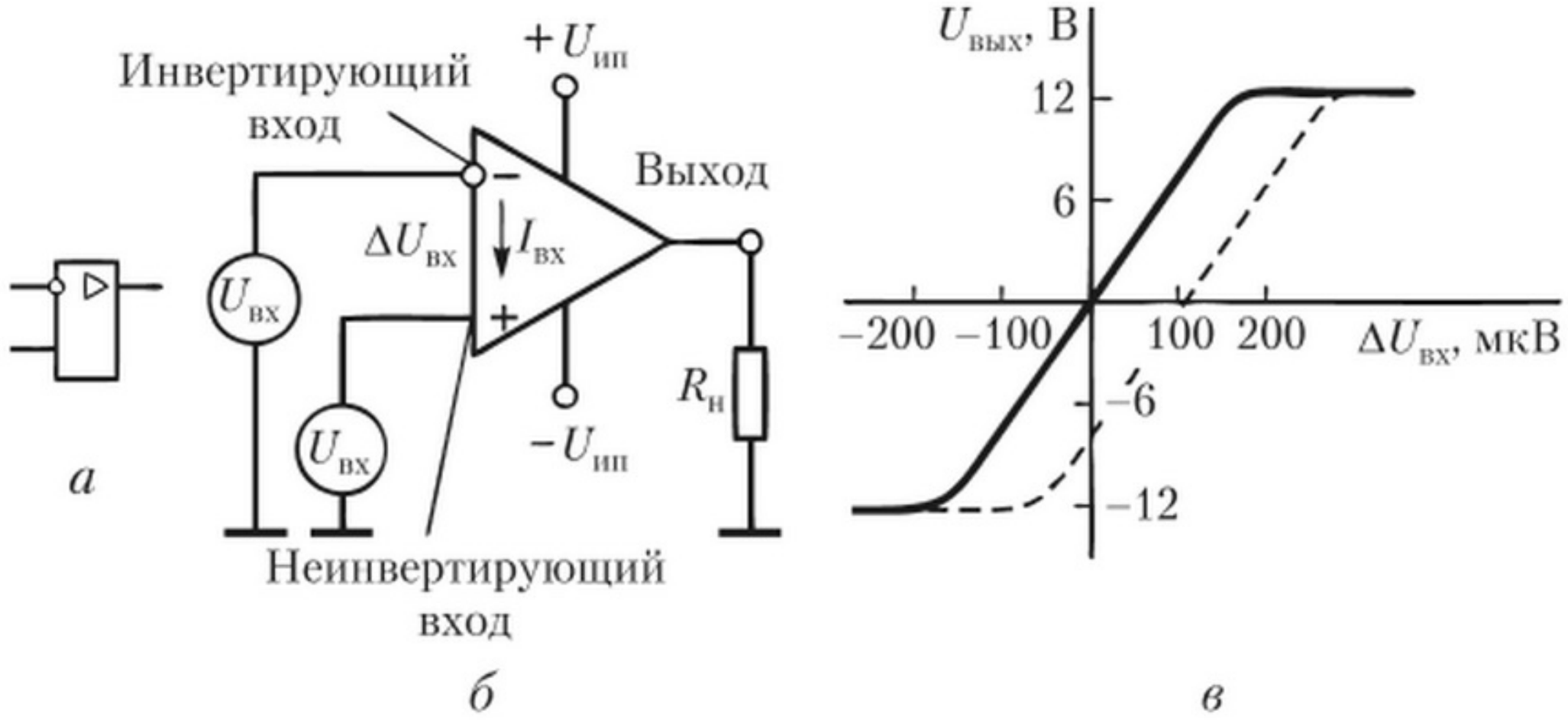


Рис. 2.13. **Условное обозначение ОУ в соответствии с ГОСТ (*а*); обозначение ОУ на функциональных схемах (*б*); зависимость выходного напряжения от дифференциального входного сигнала (*в*)**

Его входной каскад выполняется в виде дифференциального усилителя, так что операционный усилитель имеет два входа. Вход, обозначенный знаком «плюс», называется неинвертирующим, а вход, обозначенный знаком «минус», — инвертирующим. Входным сигналом для ОУ является разность между напряжениями на входах:

$$\Delta U_{вх} = U_{вх+} - U_{вх-}.$$

Чтобы обеспечить возможность работы ОУ как с положительными, так и с отрицательными входными сигналами, используется двухполярное питающее напряжение (на рис. 2.13, *б* это $+U_{ип}$ и $-U_{ип}$). Как правило, стандартные операционные усилители работают с напряжением питания ±15 В.

В действительности, разумеется, не существует идеальных ОУ. Для того чтобы можно было оценить, насколько тот или иной ОУ близок к идеалу, приводятся технические характеристики усилителей.

Дифференциальный коэффициент усиления операционного усилителя определяется как:

$$K_D = \Delta U_{вых} / \Delta U_{вх} \quad U_{вых} / (U_{вх+} \quad U_{вх-})$$

$$= \begin{cases} \Delta U_{вых} / \Delta U_{вх+} \text{ при } U_{вх-} = \text{const} \\ \Delta U_{вых} / \Delta U_{вх-} \text{ при } U_{вх+} = \text{const} \end{cases}.$$

Он имеет конечную величину, которая составляет от 10^4 до 10^6, и называется также *собственным коэффициентом усиления* ОУ, т.е. усиления при отсутствии обратной связи.

На рис. 2.13, *в* показана типовая зависимость выходного напряжения усилителя от входного сигнала. В диапазоне от −12 В до +12 В оно зависит от $\Delta U_{вх}$ почти линейно. Нелинейность наблюдается только при приближении к границам диапазона. Этот диапазон выходного напряжения называется *областью усиления*. В *области насыщения* с ростом $\Delta U_{вх}$ увеличения выходного напряжения не происходит. Границы области усиления отстоят приблизительно на 3 В от соответствующих положительного и отрицательного напряжений питания. При работе ОУ с напряжением питания ±15 В типовой диапазон области усиления по выходному напряжению составляет ±12 В.

Идеальный ОУ при нулевом входном сигнале $\Delta U_{вх} = 0$ должен иметь нулевое значение выходного напряжения. У реального усилителя это не так. Его характеристика показана пунктирной линией на рис. 2.13, *в*, и при нулевом входном сигнале на его выходе присутствует ненулевое выходное напряжение. Для того чтобы сделать выходное напряжение равным нулю, необходимо на вход ОУ подать некоторое напряжение, называемое *напряжением смещения нуля* U_0. Оно составляет обычно несколько милливольт.

Поскольку ОУ выполняется на основе полупроводников, его характеристики зависят от температуры. Так, с уменьшением температуры резко снижается коэффициент усиления. Изменение температуры влияет на величину напряжения смещения нуля и входной ток усилителя, а это оказывает влияние на точность работы схем на основе ОУ.

Влияние температуры оценивается такими характеристиками реального усилителя, как температурный дрейф напряжения смещения нуля и температурный дрейф входного тока. У лучших операционных усилителей температурный дрейф напряжения смещения нуля составляет единицы — доли микровольт на один градус, а дрейф входного тока составляет доли наноампер — единицы пикоампер (10^{-12} А) на один градус.

Кроме температуры еще одним фактором, изменяющим параметры ОУ, является время работы. С течением времени

происходит дрейф напряжения смещения нуля и входного тока ОУ. Однако эти параметры усилителя очень трудно нормировать.

2.4.2. Основные схемы включения операционных усилителей

Для анализа работы схем на основе ОУ примем два допущения.

1. Поскольку ОУ имеет очень высокий собственный коэффициент усиления K_D (порядка 10^5), а его выходное напряжение может изменяться от −12 В до +12 В, то его входной сигнал $\Delta U_{вх} = \Delta U_{вых}/K_D$ не превышает долей милливольт. Поэтому для анализа схем можно принять, что потенциалы на инвертирующем и неинвертирующем входах практически равны.

2. Поскольку входной ток реального усилителя составляет доли микроампер, можно считать, что он практически равен нулю.

Рассмотрим схему на рис. 2.14, *а*. Так как один из входов соединен с корпусом, потенциал которого принят за нуль, потенциал точки A (по первому допущению) тоже близок к нулю, а значит, ток через резистор R_1 равен

$$I_1 = (U_{вх} - U_A) / R_1 \approx (U_{вх} - 0) / R_1 \approx U_{вх} / R_1. \qquad (2.1)$$

В точке A ток I_1 разделяется на $I_{вх}$ и I_2. При этом согласно второму допущению током $I_{вх}$ можно пренебречь, поэтому:

$$I_2 = I_1 - I_{вх} \approx I_1. \qquad (2.2)$$

Напряжение $U_{вых}$ должно установиться таким, чтобы обеспечить ток I_2, протекающий через R_2 под действием разности потенциалов ($U_A - U_{вых}$), т.е. (с учетом того, что $U_A \approx 0$):

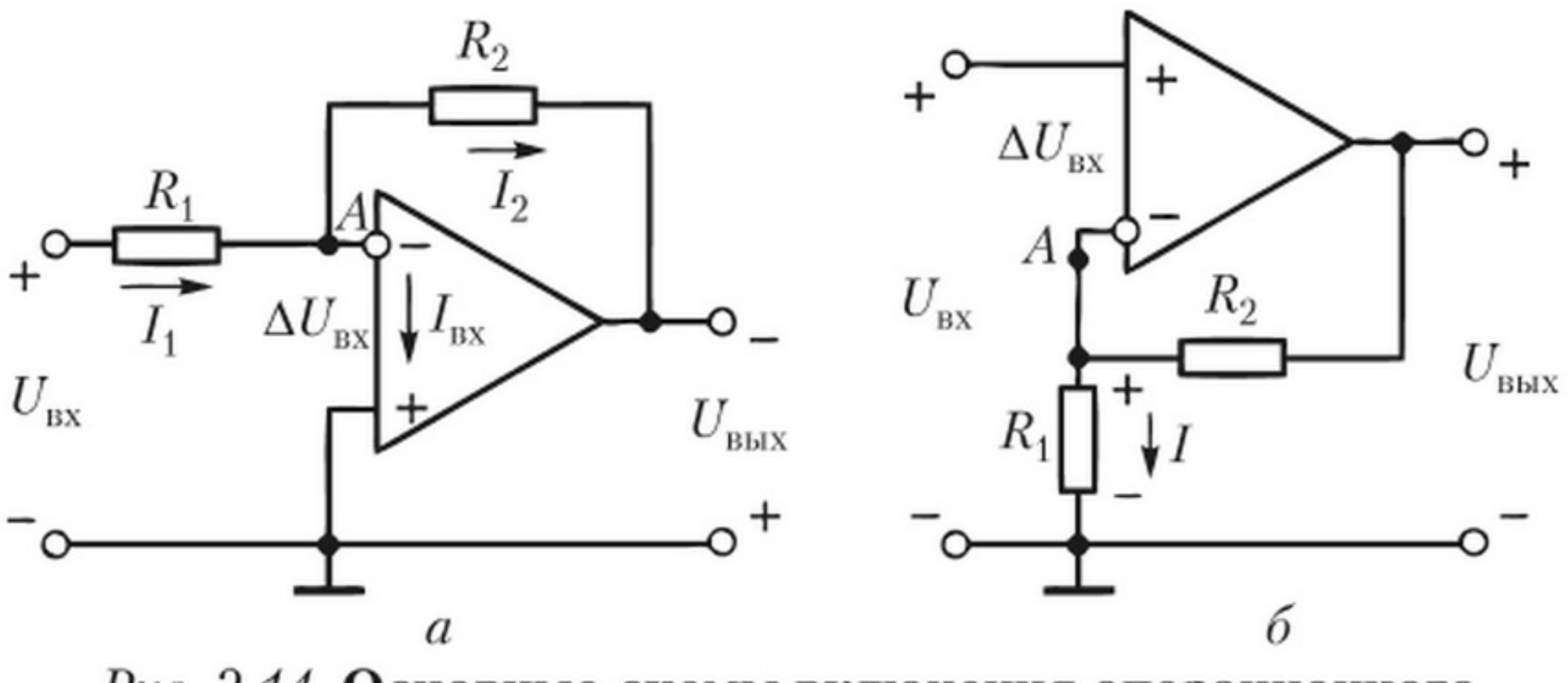
$$I_2 = (U_A - U_{вых}) / R_2 \approx -U_{вых} / R_2. \qquad (2.3)$$

Подставив (2.1) и (2.2) в (2.3), получим

$$U_{вых} = U_{вх} \frac{R_2}{R_1} \quad K_U U_{вх},$$

т.е. коэффициент усиления схемы определяется только отношением сопротивлений и не зависит от собственного ко-

эффициента усиления усилителя. Знак «минус» показывает, что подача сигнала на инвертирующий вход усилителя приводит к тому, что полярность выходного сигнала противоположна полярности входного, т.е. происходит ее инверсия. Приведенный анализ справедлив только при условии, что входной сигнал не превышает таких значений, при которых напряжение $U_{вых}$ находится на линейном участке характеристики усилителя.



а *б*

Рис. 2.14. **Основные схемы включения операционного усилителя:**

а — инвертирующее; *б* — неинвертирующее

На рис. 2.14, *б* приведена схема, в которой $U_{вх}$ подается на неинвертирующий вход ОУ. Ее анализ также легко провести с помощью указанных допущений.

При подаче $U_{вх}$ указанной полярности напряжение $U_{вых}$ начнет возрастать до тех пор, пока не достигнет значения, при котором падение напряжения на R_1 от появившегося тока I не создаст потенциал точки A, примерно равный $U_{вх}$ (по первому допущению).

При $U_A \approx U_{вх}$ окажутся выполненными оба допущения (точнее, оба условия, соответствующие установившемуся режиму работы операционного усилителя):

$$U_{вх} \approx U_A = I R_1 = U_{вых} R_1 / (R_1 + R_2).$$

Отсюда коэффициент усиления схемы

$$K_U = U_{вых} / U_{вх} = (R_1 + R_2) / R_1 = 1 + R_2 / R_1.$$

В обеих схемах включения ОУ используется отрицательная обратная связь. Действительно, в первой схеме выходное напряжение через резистор R_2 подается на вход операционного усилителя и вычитается из входного сигнала, подаваемого на вход через резистор R_1. Так как на инвертирующем входе осуществляется сравнение этих двух сигналов, эта точка называется суммирующей. Сопротивление R_1 находится в прямой цепи, а сопротивление R_2 — в цепи обратной связи. Таким образом, коэффициент усиления схемы, определяющийся соотношением R_2/R_1, зависит как от параметров цепи обратной связи, так и от параметров прямой цепи. Принцип действия отрицательной обратной связи в схеме можно сформулировать следующим образом: за счет этой связи ОУ вынужден обеспечивать такую величину выходного напряжения, чтобы напряжение на его инвертирующем входе равнялось нулю.

Во второй схеме отрицательная обратная связь формируется за счет того, что часть выходного напряжения с делителя подается на инвертирующий вход. Входное напряжение $\Delta U_{вх} = U_{вх} - U_A$ при этом уменьшается. Если представить коэффициент обратной связи $K_{ос}$ как отношение напряжения, подаваемого через обратную связь к выходному, то:

$$K_{ос} = \frac{U_A}{U_{вых}} \quad \frac{U_{вых} R_1}{(R_1 + R_2) U_{вых}} = \frac{R_1}{R_1 + R_2}.$$

Коэффициент усиления второй схемы, таким образом, целиком определяется коэффициентом обратной связи.

Важным особым случаем неинвертирующего усилителя является случай, когда коэффициент усиления схемы K_U равен единице. Для этого выход ОУ непосредственно соединяется с инвертирующим входом, т.е. $R_2 = 0$, а связь с корпусом через R_1 исключается, т.е. $R_1 = \infty$. Подобная схема включения ОУ называется следящей. Эту схему используют, когда источник сигнала имеет высокое внутреннее сопротивление. Благодаря тому, что ОУимеет низкое выходное сопротивление, схема обеспечивает преобразование сопротивлений. Разница между входным и выходным напряжением в схеме составляет несколько милливольт.

Все отклонения реального ОУот идеального, несмотря на их малую величину, оказывают влияние на работу схем, выполненных на его основе. Особенно вредно это влияние сказывается в измерительных схемах, от которых требуется высокая точность и стабильность работы в течение достаточно долгого времени. Входные токи, напряжение смещения нуля и их температурный дрейф искажают выходной сигнал. Например, в схеме неинвертирующего усилителя (см. рис. 2.14, *б*) напряжение смещения нуля эквивалентно входному сигналу и усиливается точно с таким же коэффициентом усиления. Аналогично действует напряжение смещения нуля и в схеме инвертирующего усилителя.

Действие входного тока удобно рассмотреть с помощью схемы инвертирующего усилителя. Действительно, при отсутствии входного тока ток цепи обратной связи определяется только входным сигналом. Но входной ток усилителя добавляется к этому току и, следовательно, искажает выходной сигнал на величину

$$\Delta U_i = i_{\text{вх}} R_2.$$

При настройке схем на основе ОУ искажения выходного сигнала, имеющие постоянную величину и не изменяющиеся под действием температуры и с течением времени, как правило, можно устранить с помощью регулировок. Но искажения, связанные с дрейфом, учесть при настройке не представляется возможным, поэтому так важно, чтобы ОУ обладали малым дрейфом.

2.5. Вычислительные схемы на основе операционных усилителей

Современные компьютеры позволяют с высокой точностью выполнять любые математические операции с числами. Однако на практике часто приходится оперировать с величинами, представленными в виде напряжения или тока. Так, большинство датчиков различных физических величин дают информацию об измеряемом параметре именно в таком виде. В этих случаях, чтобы использовать компьютер, приходится применять аналого-

цифровые и цифроаналоговые преобразователи. Конечно, такие системы достаточно сложные и дорогостоящие, что не всегда экономически оправдано.

Аналоговая схемотехника на основе ОУ позволяет создавать самые различные вычислительные схемы весьма просто и эффективно. На основе ОУ могут быть реализованы четыре основных действия арифметики, операции дифференцирования и интегрирования, трансцендентные функции. Недостатком этих вычислительных схем является невысокая точность вычислений — не более 0,1%. Однако этой точности в целом ряде случаев бывает вполне достаточно.

2.5.1. Схема суммирования

Для суммирования нескольких напряжений можно применить ОУ в инвертирующем включении (рис. 2.15, *а*). Входные напряжения через резисторы R_1, ..., R_n подаются на инвертирующий вход усилителя, который в этой схеме является суммирующей точкой. Поскольку эта точка является виртуальным нулем (неинвертирующий вход заземлен, а $\Delta U_{вх} \approx 0$), то токи, создаваемые входными напряжениями, будут суммироваться и течь через резистор $R_{ос}$:

$$\frac{U_1}{R_1} + \frac{U_2}{R_2} + \ldots + \frac{U_n}{R_n} = \frac{U_{вых}}{R_{ос}}.$$

Если все резисторы взять одинаковыми, то

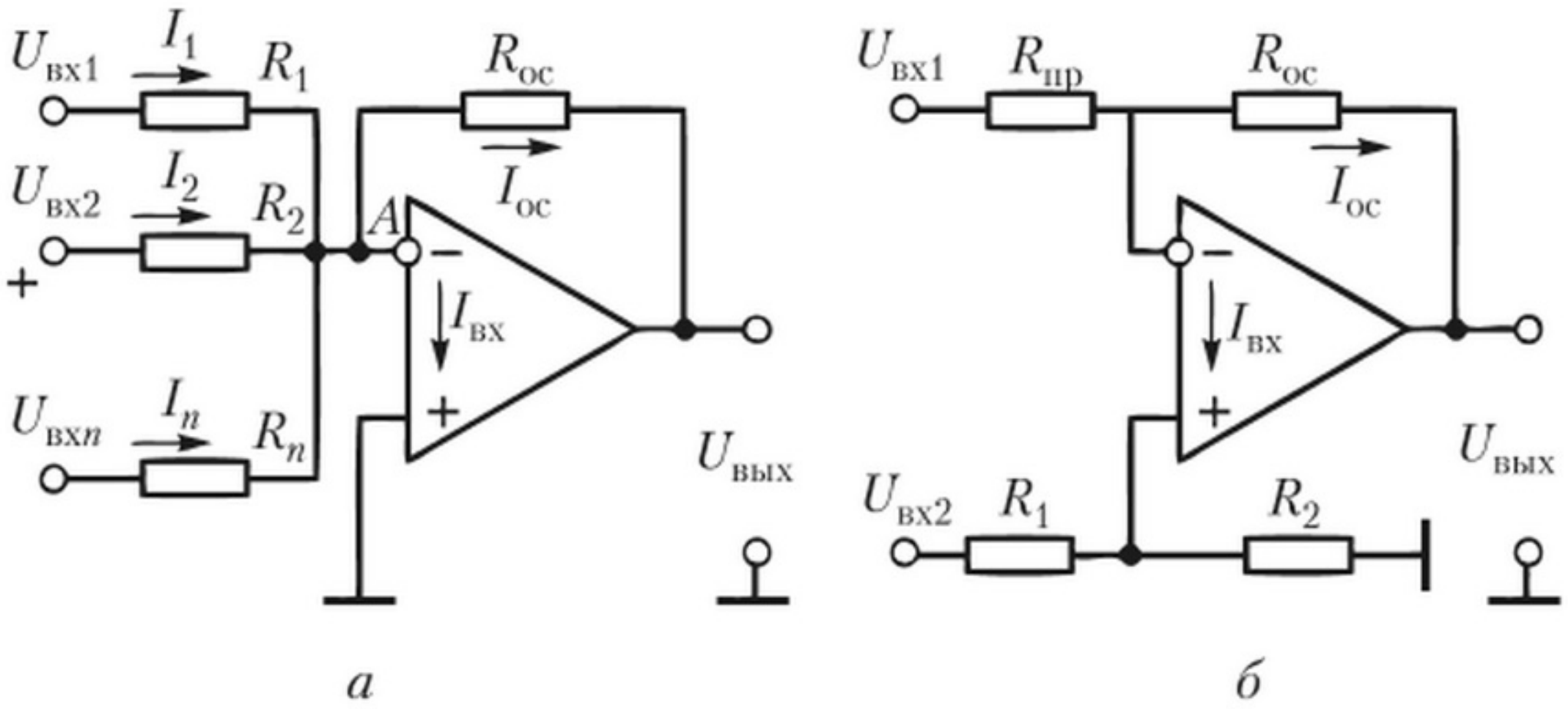
$$U_{вых} = -\sum_{i=1}^{n} U_i.$$

Суммирование входных напряжений можно также проводить с любым постоянным смещением. Для этого на один из входов подается постоянное напряжение смещения.

2.5.2. Схема вычитания

На рис. 2.15, *б* изображена схема, обеспечивающая вычитание двух напряжений:

$$U_{вых} = U_2 - U_1.$$



а *б*

Рис. 2.15. **Схемы суммирования (*а*) и вычитания (*б*)**

Если все сопротивления сделать одинаковыми, то при отсутствии второго напряжения, т.е. $U_2 = 0$, остается схема с инвертирующим включением усилителя (см. рис. 2.14, *а*), для которого

$$U_{вых} = \frac{R_{ос}}{R_{пр}} U_1 \quad \frac{R}{R} U_1 \quad U_1.$$

При отсутствии первого напряжения, т.е. $U_1 = 0$, усилитель должен создать на выходе такое напряжение $U_{вых}$, чтобы потенциалы на инвертирующем и неинвертирующем входах были бы равны. Потенциал на неинвертирующем входе равен:

$$U_+ = \frac{U_2 R_2}{R_1 + R_2} \quad \frac{U_2 R}{R + R} \quad \frac{U_2}{2}. \qquad =$$

Потенциал на инвертирующем входе (см. рис. 2.14, *б*) равен:

$$U_- = \frac{U_{вых} R_1}{R_1 + R_2} \quad \frac{U_{вых} R}{R + R} \quad \frac{U_{вых}}{2}. \qquad =$$

Таким образом, $U_{вых} = U_2$. Если же на обоих входах присутствуют сигналы, то $U_{вых} = U_2 - U_1$. Изменяя соотношение сопротивлений в делителях таким образом, чтобы

$R_{пр} = R_{ос}/\alpha$, $R_1 = R_2/\alpha$, а $R_{ос} = R_2$, операцию вычитания можно выполнять с коэффициентом α:

$$U_{вых} = \alpha (U_2 - U_1).$$

Наконец, если во входных цепях инвертирующего и неинвертирующего входов включить схемы, подобные входной схеме суммирующего усилителя и через резисторы подключить параллельно несколько входных напряжений, можно получить схему сложения и вычитания большого числа сигналов:

$$U_{вых} = \alpha \Sigma U_i - \beta \Sigma U_j,$$

где U_i — сигналы в цепи неинвертирующего входа; U_j — в цепи инвертирующего входа.

2.5.3. Схемы интегрирования и дифференцирования

Схема интегратора может строиться как на основе инвертирующего, так и на основе неинвертирующего усилителя. На рис. 2.16, *а* показана схема интегратора на основе инвертирующего усилителя.

В цепи обратной связи вместо резистора включается конденсатор. Как известно, конденсатор заряжается током, который на основании двух допущений (см. рис. 2.14, *а*) равен:

$$I_C \approx I = \frac{U_{вх}}{R}.$$

Как известно из основ электротехники, ток, заряжающий конденсатор, определяется производной от разности напряжений на его обкладках, т.е.:

$$I_C = C\frac{d(U_A - U_{вых})}{dt},$$

а учитывая первое допущение ($U_A = 0$):

$$I_C = -C\frac{d(U_{вых})}{dt}.$$

Приравняв оба выражения для тока I_C, получим:

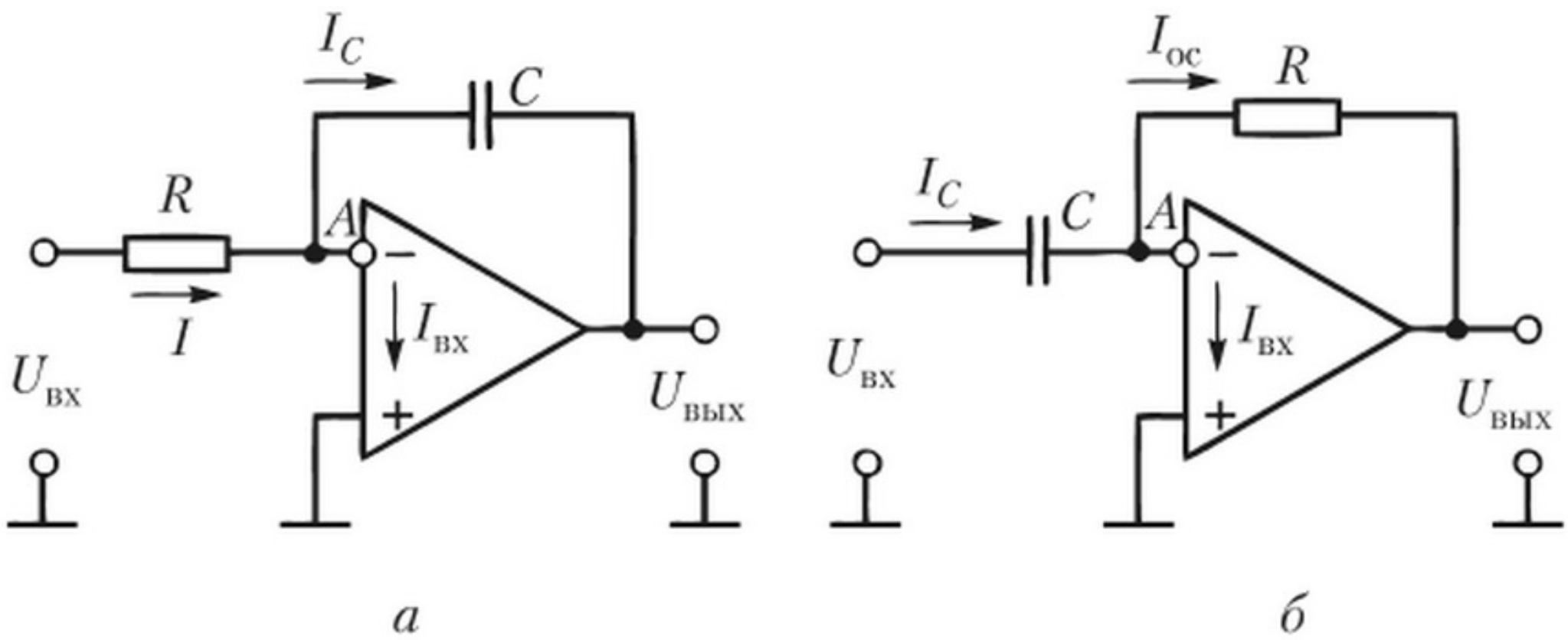
$$-C\frac{d(U_{вых})}{dt} = \frac{U_{вх}}{R}, \quad \text{или} = \frac{d(U_{вых})}{dt} \qquad \frac{1}{RC}U_{вх}.$$

Интегрируя это выражение, получим:

$$U_{\text{вых}} = -\frac{1}{RC}\int_0^t U_{\text{вх}}dt.$$

Таким образом, выходное напряжение схемы пропорционально интегралу от входного напряжения. Значения сопротивления и емкости определяют постоянный коэффициент схемы интегрирования, что можно объяснить и с чисто физических соображений. Чем больше R и C, тем меньше ток, заряжающий конденсатор, и тем больше величина его емкости, поэтому заряд конденсатора, а следовательно и выходное напряжение, будут нарастать медленнее.

Чтобы иметь схему, обеспечивающую вычисление интеграла с высокой степенью точности, используют усилители с малыми входными токами и дрейфом напряжения смещения нуля, высокоточные резисторы и конденсаторы с малыми токами утечки.



а *б*

Рис. 2.16. **Схемы интегрирования (*а*) и дифференцирования (*б*)**

Поменяв местами сопротивление и конденсатор в интеграторе (см. рис. 2.16, *а*), получим схему дифференцирования (рис. 2.16, *б*). Ток конденсатора (с учетом, что $U_{\text{A}} = 0$):

$$I_C = C\frac{d(U_{\text{вх}} - U_A)}{dt} \approx C\frac{d(U_{\text{вх}})}{dt},$$

а ток в цепи обратной связи по закону Ома:

$$I_{\text{ос}} = \frac{U_A - U_{\text{вых}}}{R} \quad \frac{U_{\text{вых}}}{R}. \approx -$$

Согласно второму допущению эти токи можно считать равными, т.е.:

$$-\frac{U_{\text{вых}}}{R} = C\frac{d(U_{\text{вх}})}{dt},$$

откуда:

$$U_{\text{вых}} \approx -RC\frac{d(U_{\text{вх}})}{dt}.$$

Таким образом, выходное напряжение в схеме пропорционально первой производной от входного напряжения.

Многие задачи описываются простыми дифференциальными уравнениями. Такие задачи можно решить, реализуя исходное дифференциальное уравнение с помощью аналоговых интегрирующих схем и измеряя установившееся выходное напряжение. Обычное дифференциальное уравнение — это зависимость функции y и ее производных от переменной x. Например, линейное дифференциальное уравнение второго порядка выглядит следующим образом:

$$y'' + k_1 y' + k_0 y = f(x).$$

Для решения уравнения производят замену переменной x на время t, т.е. $x = t/\tau$. Тогда первая и вторая производные $y' = {dy}/{dx}$ и $y'' = {d^2y}/{dx^2}$ равны:

$$y' = \frac{1}{x'(t)}\frac{dy}{dt} \quad \frac{dy}{dt} \text{ и } y'' \quad {}_2\frac{d^2y}{dt^2}.$$

Путем таких преобразований дифференциальное уравнение приводится к виду, которое может быть реализовано на основе схем интегрирования:

$$\tau^2\frac{d^2y}{dt^2} + k_1\tau\frac{dy}{dt} + k_0 y = f(t/\ ),\tau$$

ИЛИ

$$\tau\frac{d^2y}{dt^2}+k_1\frac{dy}{dt}=\frac{1}{\tau}\left[-k_0y \quad f(t/\ )\right].$$

Интегрируя левую и правую части уравнения, получим:

$$\tau\frac{dy}{dt}+k_1y=\frac{1}{\tau}\int\left[k_0y \quad f(t/\ )\right]dt.$$

Выражение, стоящее справа, реализуется с помощью интегратора. Обозначим его выходной сигнал переменой z, тогда:

$$z=\frac{1}{\tau}\int\left[-k_0y \quad f(t/\ )\right]dt.$$

С использованием переменной z уравнение преобразуется к виду

$$\tau\frac{dy}{dt}+k_1y=z, \quad \text{или} - \frac{dy}{dt}=\left[\frac{1}{\tau}\left[-z \quad k_1y\right]\right].$$

Проинтегрировав обе части уравнения, получим:

$$-y=\frac{1}{\tau}\int\left[z- \quad k_1y\right]dt.$$

Правая часть этого уравнения, так же как и в предыдущем случае, реализуется с помощью интегратора. Его выходной сигнал соответствует функции $-y$. Сигнал z поступает с выхода предыдущего интегратора, а сигнал $-k_1y$ — через обратную связь с собственного выхода.

Таким образом, для реализации дифференциального уравнения второго порядка требуется два интегратора и один инвертирующий усилитель (рис. 2.17). Инвертирующий усилитель нужен для изменения знака сигнала со второго интегратора, формирующего $-y$. Выходной сигнал с этого усилителя с коэффициентом k_0 подается на вход первого интегратора.

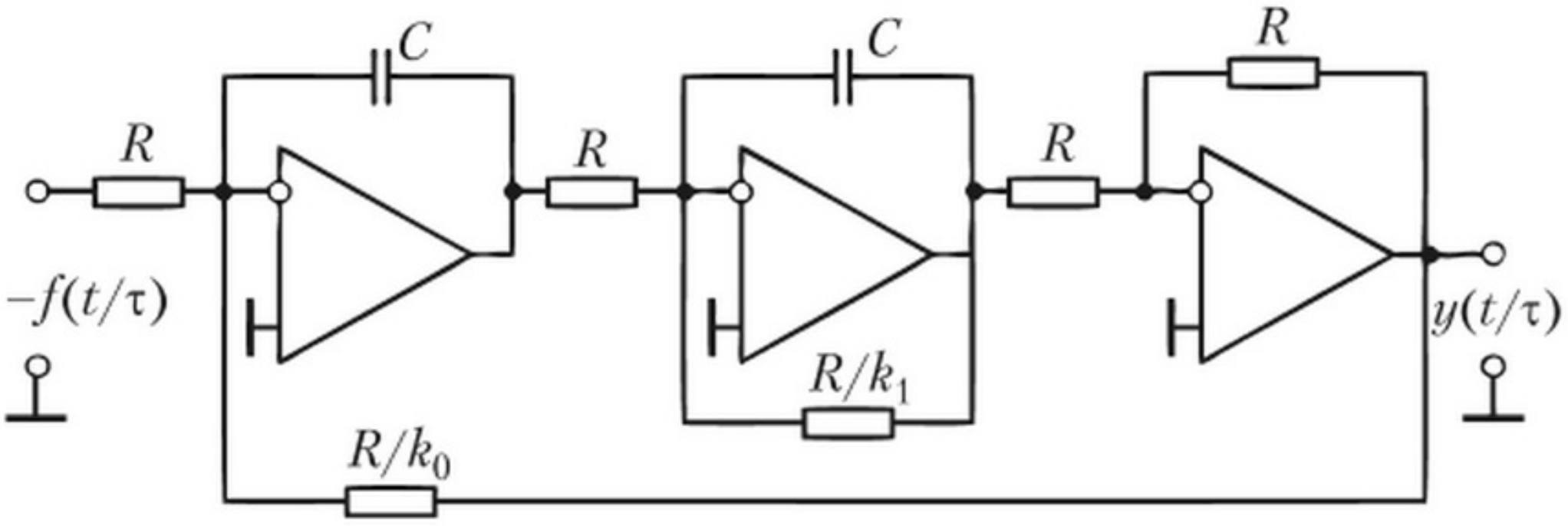


Рис. 2.17. **Аналоговая схема решения дифференциального уравнения**

2.5.4. Функциональные преобразователи

На основе ОУ можно формировать сигналы, реализующие различные математические функции — логарифм, экспоненту, синус и косинус. Для этого используются либо нелинейность характеристик полупроводниковых приборов — транзисторов или диодов, либо осуществляется аппроксимация функции полиномом или степенным рядом.

Логарифм. Для формирования выходного сигнала усилителя, пропорционального логарифму входного, можно использовать характеристику диода в схеме на рис. 2.18, *а*. Взаимосвязь между прямым током диода $I_{\text{пр}}$ и падающим на нем напряжением $U_{\text{пр}}$ имеет следующий вид:

$$I_{\text{пр}} = I_{\text{обр}}(\exp\frac{U_{\text{пр}}}{mU_T} - 1),$$

где $I_{\text{обр}}$ — обратный ток, образованный неосновными носителями; m — корректирующий множитель; $U_T = kT/e_0$ — термический потенциал; k — постоянная Больцмана; T — температура; e_0 — заряд электрона.

В рабочей области, где $I_{\text{пр}} >> I_{\text{обр}}$, можно считать, что

$$I_{\text{пр}} = I_{\text{обр}} \exp\frac{U_{\text{пр}}}{mU_T}.$$

Отсюда следует, что

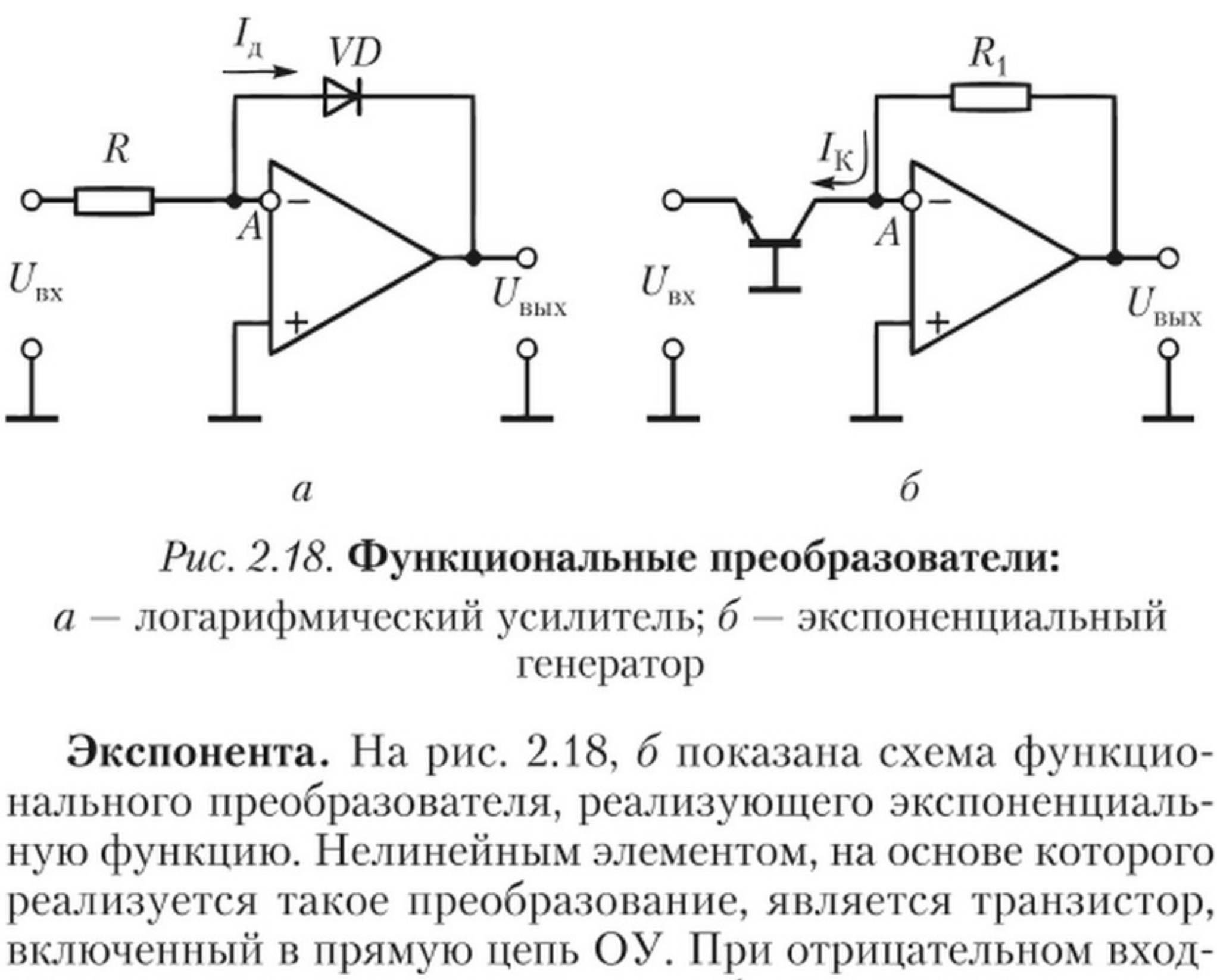
$$U_{\text{пр}} = mU_T \ln (I_{\text{пр}}/I_{\text{обр}}).$$

Напряжение на диоде, по существу, равно выходному напряжению, только с обратным знаком $U_{вых} = -U_{пр}$. Ток в прямой цепи, создаваемый входным напряжением $I_{пр} = U_{вх} / R$, равен току в цепи обратной связи, т.е. прямому току диода, поэтому взаимосвязь выходного напряжения с входным имеет следующий вид:

$$U_{вых} = -mU_T \ln (U_{вх}/RI_{обр}).$$

Так как значения m, U_T, R и $I_{обр}$ постоянны, выходное напряжение становится функцией только входного напряжения и пропорционально его логарифму.

Вместо диода в схемах логарифмического усилителя применяются также транзисторы, у которых зависимость коллекторного тока от напряжения $U_{БЭ}$ также имеет логарифмический вид.



а *б*

Рис. 2.18. **Функциональные преобразователи:**
а — логарифмический усилитель; *б* — экспоненциальный генератор

Экспонента. На рис. 2.18, *б* показана схема функционального преобразователя, реализующего экспоненциальную функцию. Нелинейным элементом, на основе которого реализуется такое преобразование, является транзистор, включенный в прямую цепь ОУ. При отрицательном входном напряжении через транзистор будет течь ток:

$$I_К = I_{К0} \exp \frac{U_{БЭ}}{U_T}.$$

Выходное напряжение определяется падением напряжения на R_1, которое зависит от тока в цепи обратной

связи $U_{вых} = I_{ос}R_1$. Так как ток коллектора $I_К$ равен току $I_{ос}$, а напряжение $U_{БЭ}$ соответствует входному напряжению, только со знаком «минус», то взаимосвязь между входным и выходным напряжениями будет иметь следующий вид:

$$U_{вых} = I_К R_1 \quad I_{К0} R_1 \exp \frac{-U_{вх}}{U_T}.$$

На основе логарифмических и экспоненциальных усилителей выполняются схемы для решения степенных уравнений. Логарифмические усилители также являются основой для схем аналогового умножения.

Функции синус и косинус. Чтобы реализовать синусоидальную зависимость выходного напряжения от входного, используется метод *кусочно-линейной аппроксимации*.

График функции (рис. 2.19) в диапазоне от –90° до +90° аппроксимируется ломаной линией. Каждый линейный участок можно реализовать с помощью усилителя с соответствующим коэффициентом передачи. На первом участке от точки 0 до точки *a* коэффициент передачи входного напряжения самый высокий. Когда входное напряжение достигнет второго участка (между точками *a* и *b*), коэффициент передачи усилителя уменьшается и выходное напряжение растет медленнее. На третьем участке между точками *b* и *c* — еще медленнее, пока входное напряжение не достигнет точки, соответствующей +90°.

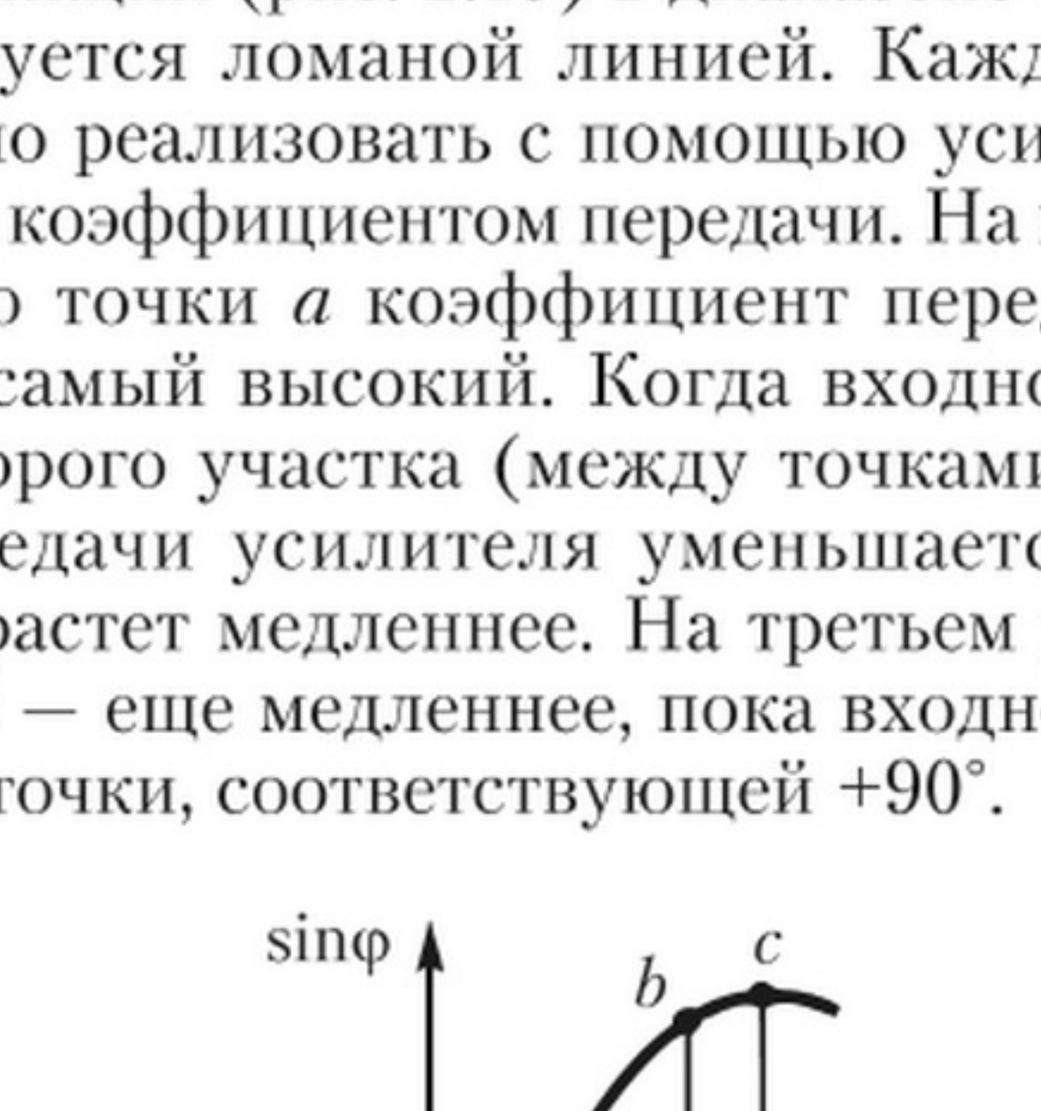


Рис. 2.19. **Аппроксимация функции sin**

Содержание

Так как коэффициент передачи усилителя зависит от соотношения сопротивлений в прямой цепи и цепи обратной связи, то изменять коэффициент передачи можно путем динамического уменьшения сопротивления в одной из цепей с ростом входного напряжения. Для этого используют диоды, которые при достижении определенного значения напряжения открываются и подключают параллельно основному сопротивлению цепи дополнительные сопротивления.

2.6. Генераторы на основе операционных усилителей

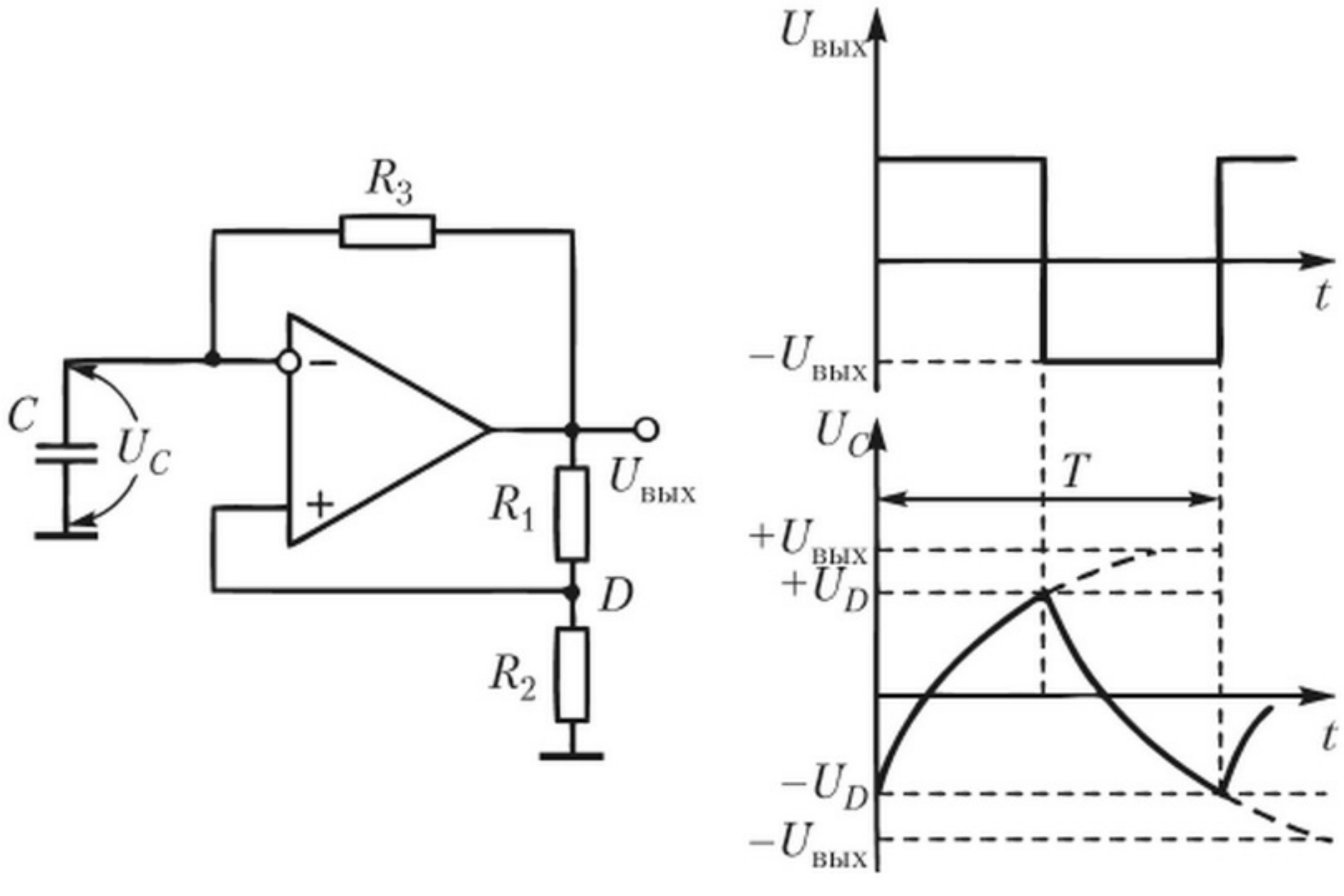
Электронным генератором называют устройство для формирования незатухающих электрических колебаний различной формы, частоты и мощности. Очень часто генераторы выполняют на основе ОУ.

2.6.1. Мультивибратор

Мультивибратором называют генератор напряжения с формой, близкой прямоугольной. Его название отражает тот факт, что такое напряжение при разложении в ряд Фурье представляется рядом, содержащим много высших гармоник (*мульти* — много).

По характеристике ОУ (см. рис. 2.13, *б*) видно, что выходное напряжение усилителя линейно зависит от входного только в очень узком диапазоне — сотнях микровольт. Если входное напряжение выходит за пределы этого диапазона, то выходной сигнал может принимать только два значения: $+U_{вых}$ (≈ +12 В) и $-U_{вых}$ (≈ −12 В). На этой особенности операционного усилителя основан принцип формирования прямоугольного напряжения мультивибратора (рис. 2.20, *а*).

Предположим, что в момент включения между входами усилителя небольшая (достаточно единиц милливольт) отрицательная разность потенциалов. При этом на выходе сформируется напряжение $+U_{вых}$, а на неинвертирующий вход с делителя R_1, R_2 будет подан положительный потенциал $+U_D$. Конденсатор начнет заряжаться по цепи «$U_{вых}$—R_3—C—корпус», стремясь достичь потенциала $+U_{вых}$. Потенциал на инвертирующем входе начнет расти до тех пор, пока не превысит потенциал на неинвертирующем входе $+U_D$. В этот момент усилитель сформирует



а *б*

Рис. 2.20. **Мультивибратор (*а*) и графики, поясняющие его работу (*б*)**

на выходе отрицательное напряжение $-U_{вых}$ и создаст на неинвертирующем входе отрицательный потенциал $-U_D$. Теперь конденсатор начнет перезаряжаться, стремясь достичь потенциала $-U_{вых}$. Однако как только потенциал на инвертирующем входе станет ниже потенциала на неинвертирующем входе $-U_D$, усилитель сформирует на выходе положительное напряжение $+U_{вых}$. Такой скачкообразный процесс изменения выходного напряжения с $+U_{вых}$ до $-U_{вых}$ и обратно будет повторяться до тех пор, пока с операционного усилителя не будет снято питающее напряжение. Графики, демонстрирующие описанные процессы, показаны на рис. 2.20, *б*. Период T-колебаний определяется постоянной времени заряда конденсатора $\tau = R_3C$, а также тем, насколько потенциал, формируемый делителем R_1, R_2, меньше напряжения $U_{вых}$.

2.6.2. Генератор пилообразного напряжения

Напряжение на конденсаторе прямолинейно возрастает, если его заряжать постоянным током, не зависящим от напряжения на нем, и предотвратить влияние на этот ток сопротивления нагрузки, т.е. должно выполняться условие $R_н >> R$. Интегрируя по времени выражение

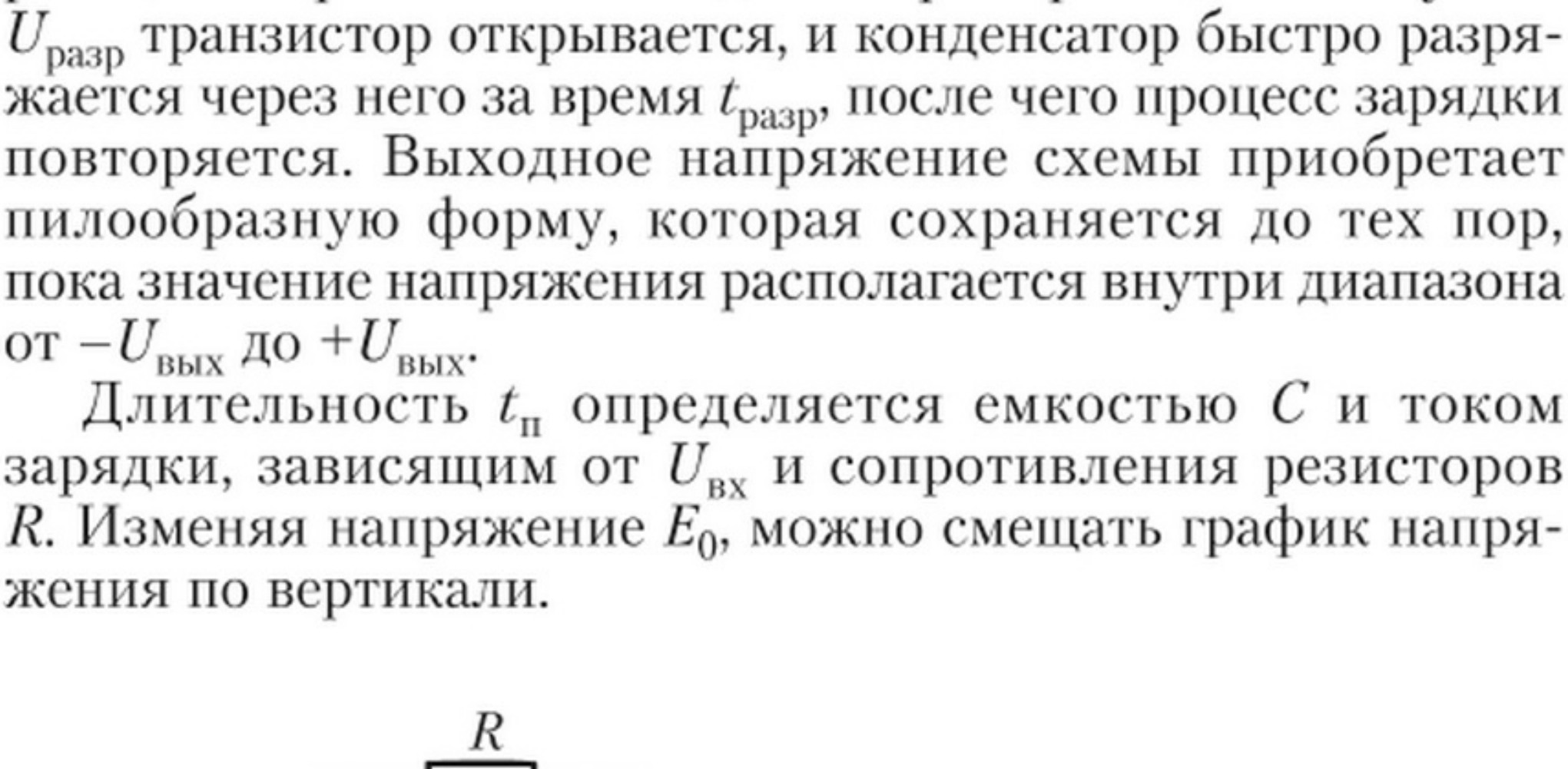
$$I_C = C\frac{dU_C}{dt},$$

получим:

$$\int dU_C = \frac{I_C}{C}\int dt, \text{ или } U_C = \frac{I_C}{C}t.$$

Условие I_C = const в схеме генератора пилообразного напряжения (ГПН) на основе ОУ (рис. 2.21, *а*) обеспечивается постоянным напряжением $U_{вх}$. Пока транзистор заперт, в течение времени $t_п$ происходит зарядка конденсатора и напряжение на нем нарастает по прямой. Усилитель, стремясь сделать разность потенциалов на его входах, близкой к нулю, формирует выходное напряжение, повторяющее напряжение на конденсаторе. При подаче импульса $U_{разр}$ транзистор открывается, и конденсатор быстро разряжается через него за время $t_{разр}$, после чего процесс зарядки повторяется. Выходное напряжение схемы приобретает пилообразную форму, которая сохраняется до тех пор, пока значение напряжения располагается внутри диапазона от $-U_{вых}$ до $+U_{вых}$.

Длительность $t_п$ определяется емкостью C и током зарядки, зависящим от $U_{вх}$ и сопротивления резисторов R. Изменяя напряжение E_0, можно смещать график напряжения по вертикали.



а *б*

Рис. 2.21. **Генератор пилообразного напряжения (*а*), форма выходного напряжения (*б*)**

2.7. Компараторы, цифроаналоговые и аналого-цифровые преобразователи

Как отмечалось в параграфе 2.1, электронные устройства делятся на аналоговые и цифровые. В современных системах управления различными процессами, в том числе и технологическими, присутствуют устройства обоих типов. Аналоговые устройства обычно обеспечивают съем первичной информации с датчиков системы управления приводами исполнительных устройств и механизмов, управление же самим процессом в соответствии с заданной алгоритмом программой выполняют цифровые устройства. Взаимодействие между аналоговой частью системы и цифровой (преобразование информации из аналоговой формы в цифровую и обратно) обеспечивают цифроаналоговые (ЦАП) и аналого-цифровые преобразователи (АЦП).

Из большого числа известных методов построения таких преобразователей рассмотрим наиболее типичные, используемые в выпускаемых промышленностью БИС.

Очень важную роль при аналого-цифровом преобразовании играют компараторы. *Компаратором* называют устройство, предназначенное для сравнения изменяющегося аналогового входного сигнала с опорным напряжением. При этом в зависимости от того, больше входной сигнал опорного или меньше (на доли милливольт), на выходе компаратора должно установиться напряжение «логический нуль» (лог. 0) или «логическая единица» (лог. 1). Так как выходной сигнал компаратора подается обычно на логические схемы, его выходное напряжение согласуется с цифровыми логическими схемами.

Функцию сравнения двух напряжений может выполнить и ОУ, если на один из его входов подать опорное напряжение, а на другой — входной сигнал. Однако специализированные устройства — компараторы — имеют преимущество в быстродействии, которое получают, предотвращая режим насыщения его транзисторов, а следовательно, и длительное рассасывание неосновных носителей. Схемы компараторов похожи на схемы ОУ, но менее сложные. Компараторы выпускаются в интегральном исполнении, а также являются составной частью микросхем АЦП.

2.7.1. Цифроаналоговые преобразователи

ЦАП обеспечивают прием от цифрового устройства двоичного кода и преобразование его в напряжение постоянного тока, пропорциональное этому коду, для последующей передачи его аналоговым устройствам.

На рис. 2.22 пояснен принцип цифроаналогового преобразования с помощью суммирующего ОУ. Вследствие большого входного сопротивления входной ток ОУ пренебрежимо мал, поэтому сумма токов ΣI_i, притекающих к точке A, равна току $I_{ос}$, вытекающему из этой точки. Разность потенциалов между инвертирующим и неинвертирующим входами вследствие большого коэффициента усиления собственно ОУ (при условии, что он находится на линейном участке характеристики) также пренебрежимо мала, поэтому потенциал точки A оказывается близким к нулевому потенциалу корпуса. С учетом этих допущений выходное напряжение (см. рис. 2.15, *а*) равно:

$$U_{вых} = \Sigma I_i R_{ос}.$$

Предположим, что с помощью ключей $Кл_i$ к точке A могут быть поданы токи, значения которых пропорциональны весу разрядов двоичного кода (если младшему разряду соответствует ток I, то ток каждого следующего разряда в 2 раза больше, т.е. $2I$, $4I$, $8I$ и т.д. до $2^{n-1}I$ для старшего разряда n). Если при значении i-го разряда, равном 1, ключ $Кл_i$

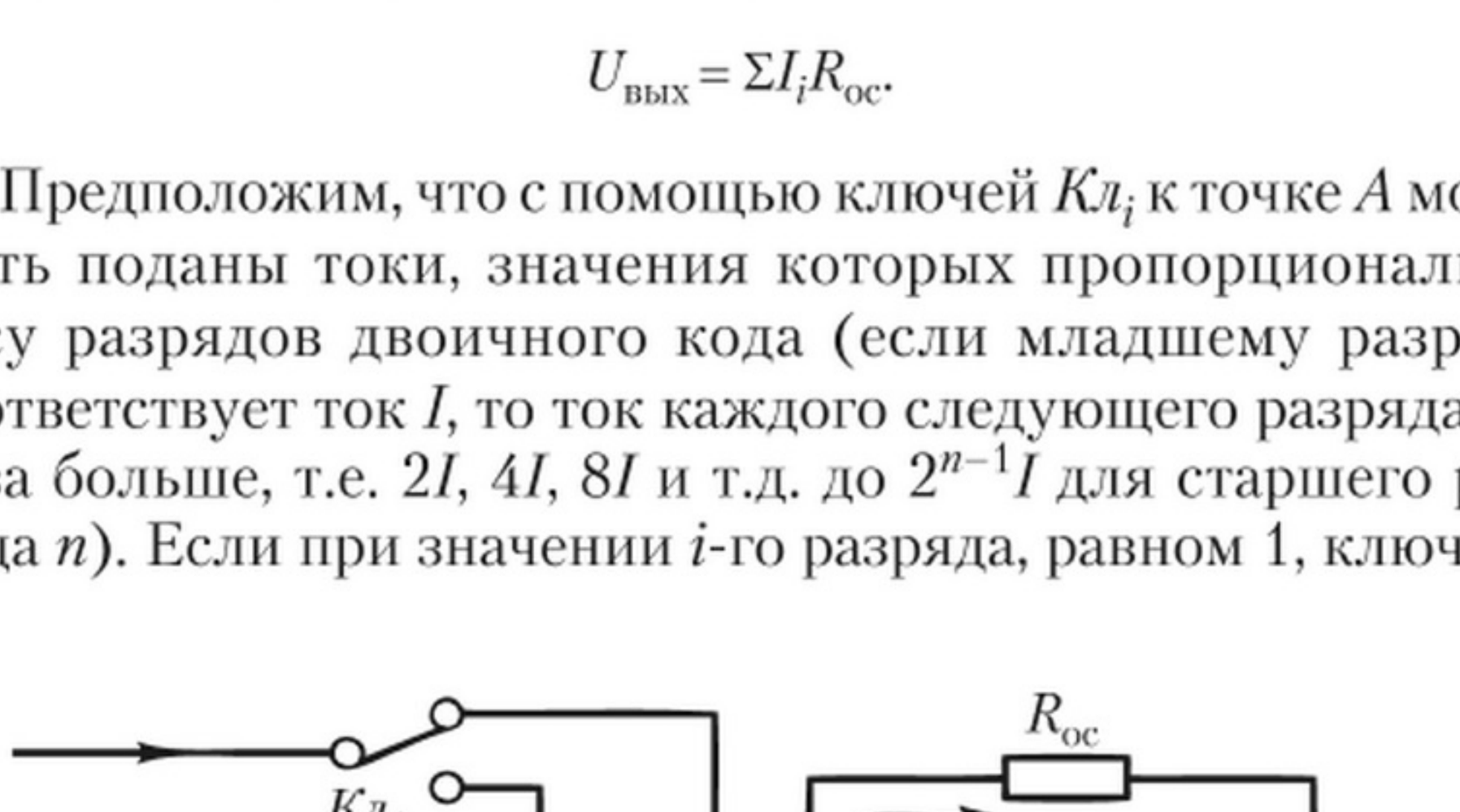
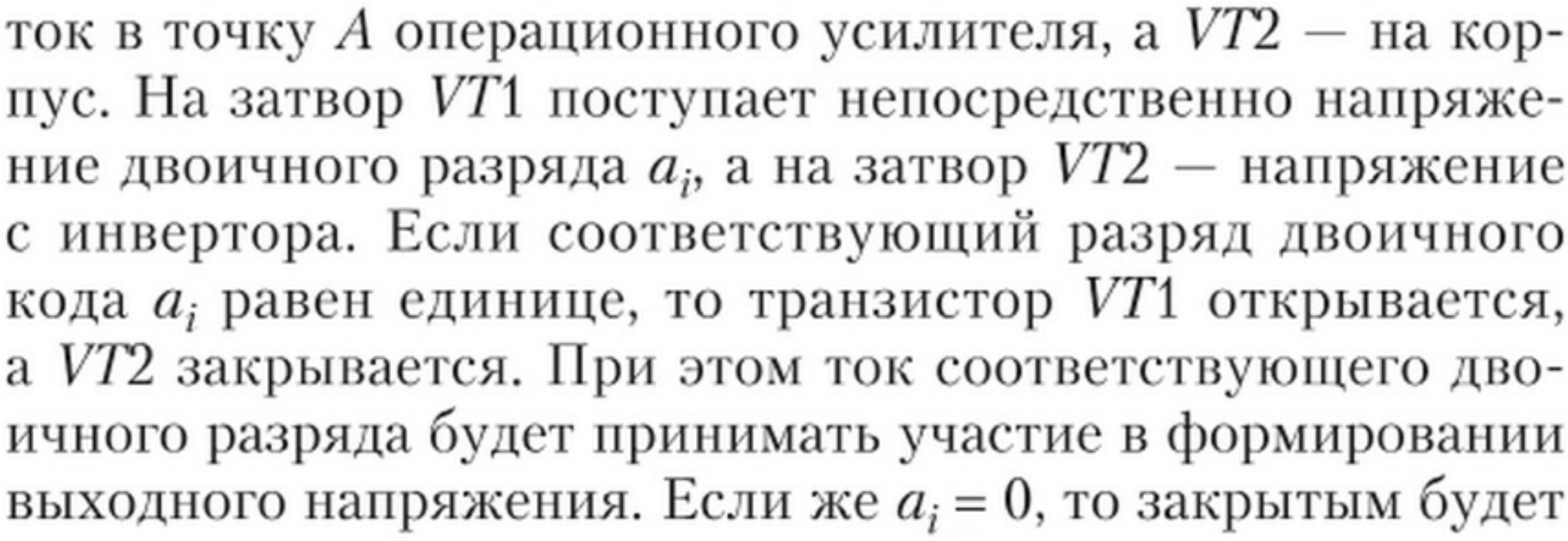


Рис. 2.22. **Принцип работы ЦАП на основе ОУ**

замкнут, а при 0 — разомкнут, то сумма токов ΣI_i, а значит, и $U_{вых}$ окажутся пропорциональны значению n-разрядного двоичного числа, код которого управляет ключами.

Для создания токов, пропорциональных весу двоичных разрядов, используется так называемая матрица R–$2R$ (рис. 2.23, *а*), которую подключают к источнику стабильного напряжения $U_{оп}$. Нетрудно убедиться, что для каждого узла матрицы сопротивление всей цепи, расположенной правее узла, всегда равно $2R$. Поэтому ток, втекающий в узел, разделяется поровну между резистором $2R$ (вниз) и параллельной ему цепью (вправо). В следующем узле опять происходит деление втекающего тока пополам, но поскольку он в два раза меньше, чем ток, втекающий в предыдущий узел, то и через резистор $2R$ будет течь в два раза меньший ток. Таким образом, токи, текущие через резисторы $2R$, будут пропорциональны весу двоичных разрядов a_i. Для коммутации токов используется схема, изображенная на рис. 2.23, *б*. Роль ключей выполняет пара полевых транзисторов, при этом $VT1$ направляет ток в точку A операционного усилителя, а $VT2$ — на корпус. На затвор $VT1$ поступает непосредственно напряжение двоичного разряда a_i, а на затвор $VT2$ — напряжение с инвертора. Если соответствующий разряд двоичного кода a_i равен единице, то транзистор $VT1$ открывается, а $VT2$ закрывается. При этом ток соответствующего двоичного разряда будет принимать участие в формировании выходного напряжения. Если же $a_i = 0$, то закрытым будет транзистор $VT1$, а открытым — $VT2$, и соответствующий ток замыкается на корпус.



а *б*

Рис. 2.23. **Матрица R–$2R$ (*а*), схема коммутации токов (*б*)**

Содержание

2.7.2. Аналого-цифровые преобразователи

Если переменные информационные величины представлены в аналоговой форме в виде изменяющегося напряжения постоянного тока, их непосредственная обработка цифровыми устройствами невозможна без предварительного представления в виде n-разрядного кода. Эту операцию осуществляют АЦП.

Существует ряд способов такого преобразования, обладающих своими преимуществами и недостатками; рассмотрим их.

В способе *последовательного приближения* схема формирует пробные коды, поступающие на ЦАП, выходной сигнал которого сравнивается с помощью компаратора с входным аналоговым сигналом (рис. 2.24).

Выходной двоичный код формируется в регистре, управляемом устройством управления (УУ). Регистр связан с ЦАП, который формирует напряжение, пропорциональное коду, подаваемое на один из входов компаратора. На другой вход компаратора подается напряжение $U_{вх}$, подлежащее преобразованию в код. В исходном состоянии устройство управления устанавливает все разряды регистра в «0». Затем в старший разряд заносится «1». Если при этом $U_{вх} > U_{ЦАП}$, то устройство управления оставляет «1» в старшем n-м разряде, если же $U_{вх} < U_{ЦАП}$, то в старший разряд заносится «0». Затем устройство управления заносит «1» в следующий $(n - 1)$-й разряд и вновь в зависимости от результата сравнения устройство управления либо

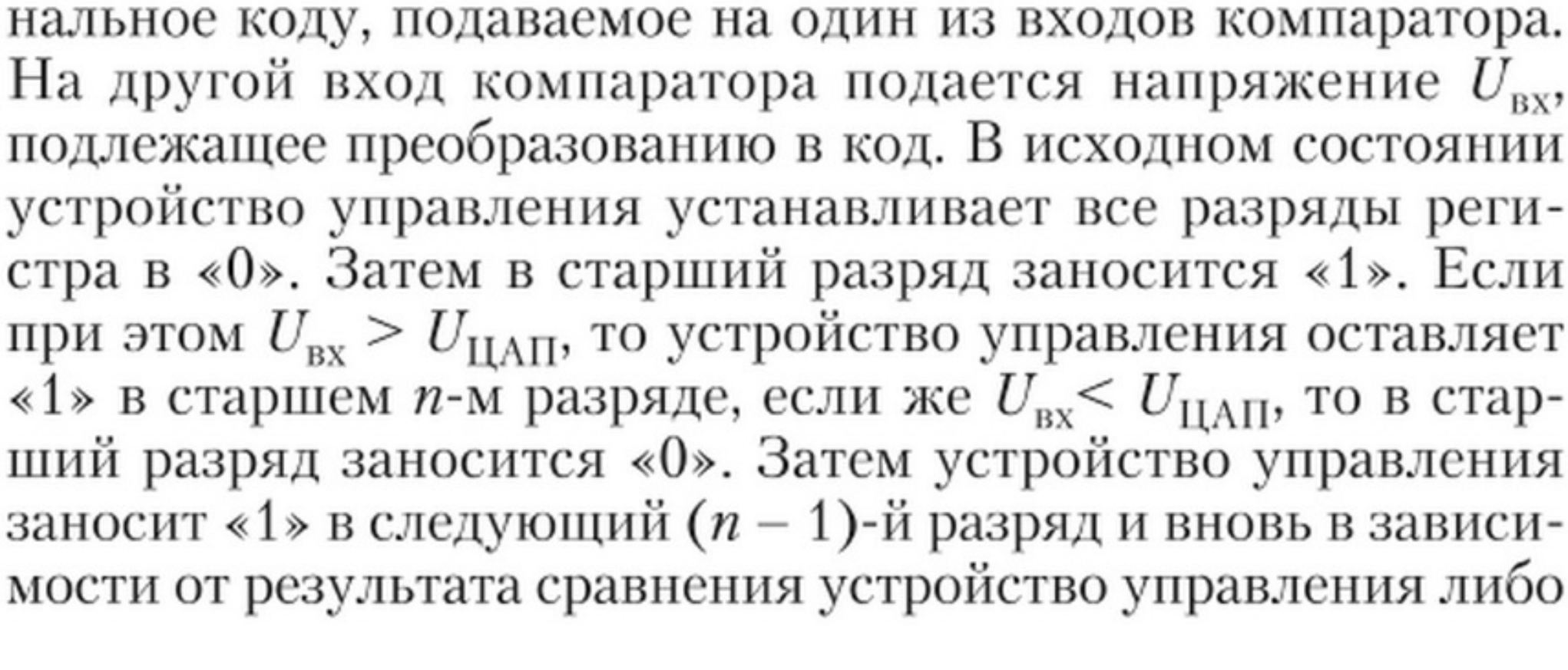


Рис. 2.24. **АЦП последовательного приближения**

оставляет «1» в этом разряде, либо записывает «0». Таким образом, устройство управления заносит во все разряды регистра «1» или «0», начиная от самого старшего и кончая самым младшим. Работа АЦП синхронизируется генератором тактовых импульсов (ГТИ). После n тактов сравнения $U_{вх}$ с $U_{ЦАП}$ на выходе АЦП получается n-разрядный двоичный код, эквивалентный входному аналоговому сигналу. Такие преобразователи имеют относительно высокую точность, однако для n-разрядного преобразования требуют n тактов. При этом, если за время преобразования входной сигнал изменяется, возникает ошибка, особенно заметная при коротких выбросах входного сигнала.

В АЦП, использующего способ *пилообразного напряжения*, нет цифроаналогового преобразования (рис. 2.25). Принцип его работы основан на измерении интервала времени, в течение которого линейно нарастающее напряжение достигнет уровня $U_{вх}$. По команде «Начало преобразования» одновременно запускается ГПН и взводится триггер, разрешающий прохождение тактовых импульсов на вход счетчика. Линейно нарастающее напряжение $U_{п}$, формируемое ГПН, поступает на один из входов компаратора, на другой вход которого подается напряжение $U_{вх}$, подлежащее преобразованию в код. В момент, когда напряжение $U_{п}$ сравняется с $U_{вх}$, компаратор выдаст команду, сбрасывающую триггер и запрещающую прохождение тактовых импульсов на вход счетчика. Таким образом, код на выходе счетчика будет прямо пропорционален отрезку времени,

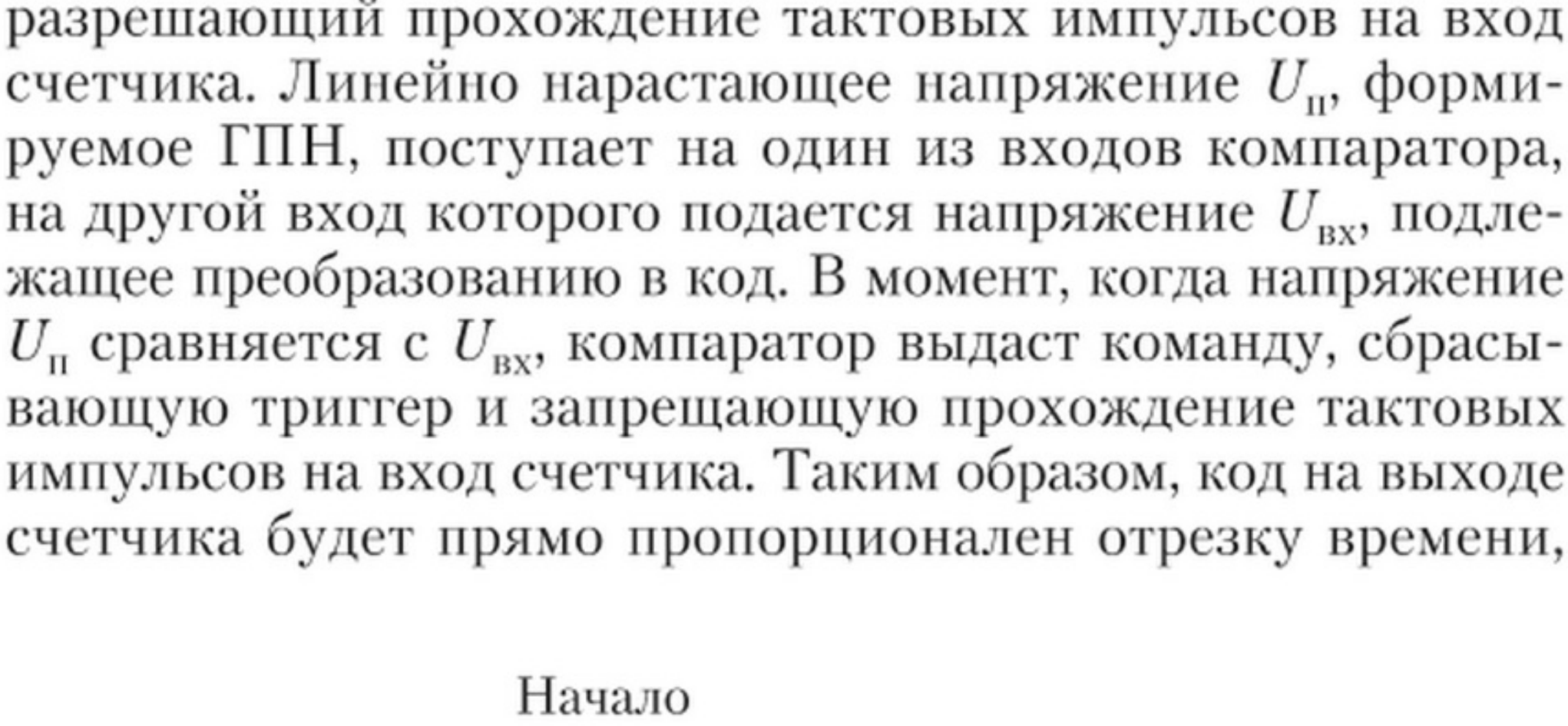


Рис. 2.25. **АЦП пилообразного напряжения**

Содержание

в течение которого напряжение $U_{п}$ достигнет уровня $U_{вх}$, т.е. самой величине $U_{вх}$.

Как следует из принципа работы данного АЦП, он имеет невысокое быстродействие. Точность преобразования определяется стабильностью ГПН, которая в основном зависит от температурной нестабильности параметров времязадающей *RC*-цепи.

Самым быстродействующим и в то же время самым сложным является АЦП, использующий способ *параллельного кодирования* (рис. 2.26). Входное напряжение $U_{вх}$ подается одновременно на верхние входы всех компараторов. На нижние входы компараторов подается напряжение с делителя, состоящего из резисторов одного номинала *R*. Таким образом, напряжение, с которым осуществляется сравнение входного сигнала у двух соседних компараторов, отличается на величину, соответствующую цене самого младшего разряда. Приоритетный шифратор формирует выходной цифровой код, соответствующий самому старшему из сработавших компараторов. По единичному сигналу «Запись» *n*-разрядный код с шифратора через конъюнкторы поступает в параллельный регистр.

Высокое быстродействие АЦП, реализующего этот способ, достигается за счет значительных аппаратурных затрат и большой потребляемой мощности. Например, для вось-

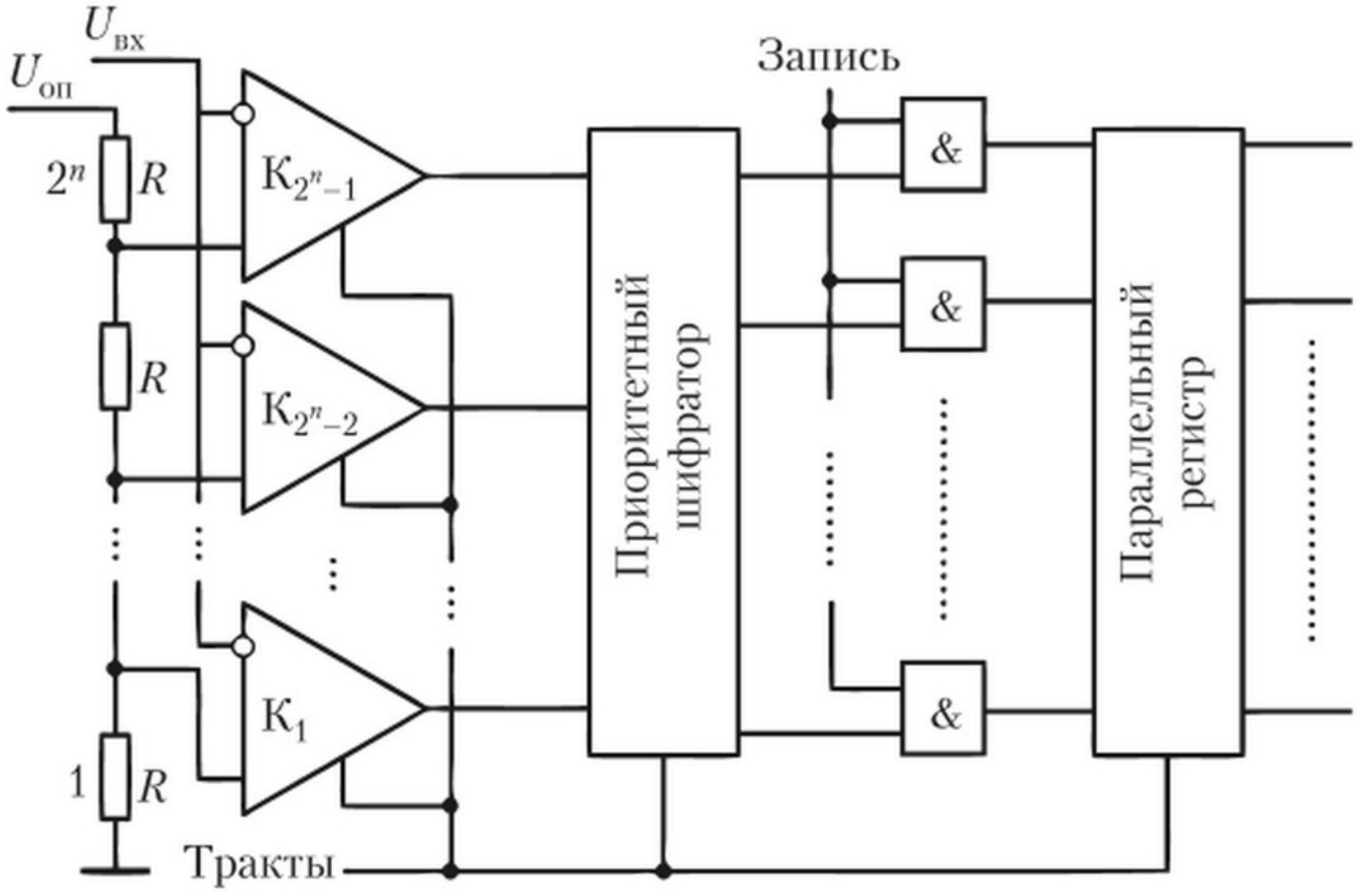


Рис. 2.26. **АЦП параллельного кодирования**

миразрядного АЦП требуется 255 компараторов и около $3 \cdot 10^4$ активных элементов, потребляющих примерно 2,5 Вт.

2.8. Источники вторичного электропитания

2.8.1. Структуры источников электропитания

Для любого электронного устройства необходим источник питания, который должен давать одно или несколько значений постоянного напряжения. Конечно, в качестве источника питания можно использовать гальванические батареи, но при большом потреблении мощности это неэкономично. В этом случае применяют специальные электронные устройства, обеспечивающие формирование требуемых питающих напряжений и называемые источниками электропитания.

Источники электрической энергии, необходимой для питания любой электронной аппаратуры, принято делить на источники первичного и вторичного электропитания.

К *первичным источникам электропитания* относят трехфазную (или однофазную) сеть промышленной частоты 50 Гц (для стационарной аппаратуры) и генераторы постоянного или переменного напряжения повышенной частоты 400—500 Гц. Химические гальванические элементы и солнечные батареи используются только для питания бортовой аппаратуры, устанавливаемой на подвижных объектах и требующих автономного электропитания.

Источники вторичного электропитания (ИВЭП) выполняют функции преобразования вида тока (переменный — постоянный), стабилизации и регулировки напряжения или тока, фильтрации различных помех, возникающих при переключении, стабилизации и регулировке напряжения и т.д. Преобразование переменного тока в постоянный осуществляется *выпрямителями*, а обратное преобразование постоянного тока в переменное — *инверторами* (не путать с инверторами — электронными схемами «НЕ»).

Подавляющая часть устройств информационно-вычислительных систем потребляет электрическую энергию в виде постоянного тока. Если первичным источником служит сеть переменного тока $U_{\sim}$, то ИВЭП чаще всего имеет структуру, приведенную на рис. 2.27, *а*.

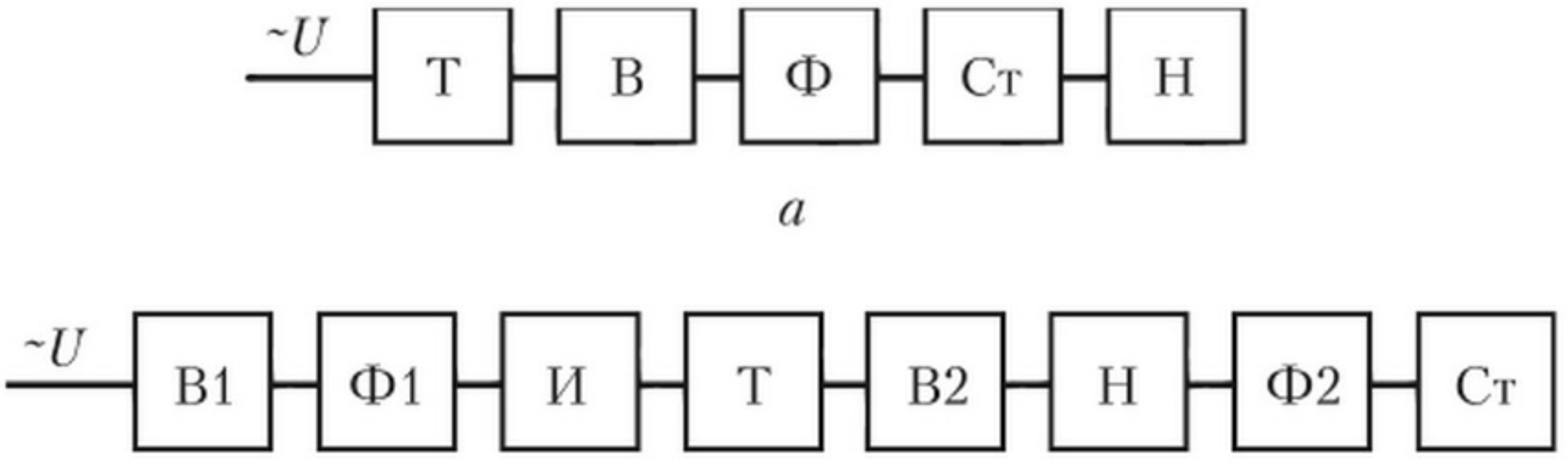


Рис. 2.27. **Структурные схемы ИВЭП**

Мощный трансформатор Т, как правило, понижает напряжение, затем оно преобразуется выпрямителем В в постоянное напряжение, пульсации которого сглаживаются фильтром Ф, и при необходимости уровень этого напряжения с помощью стабилизатора Ст поддерживается неизменным, не зависящим от изменений напряжения сети, температуры, тока нагрузки Н и других дестабилизирующих факторов.

ИВЭП являются неотъемлемой частью любой электронной аппаратуры и в значительной степени определяют ее технико-экономические показатели. На долю источников питания нередко приходится до 40% общей массы и объема аппаратуры, поэтому одной из задач, стоящих перед проектировщиками, является их комплексная миниатюризация.

Из всех узлов ИВЭП наиболее громоздкие, как правило, — узлы, выполненные на магнитопроводах из ферромагнитных материалов (трансформаторы и дроссели фильтров).

Стремление уменьшить массу и габаритные размеры источников электропитания привело к структурной схеме, называемой «ИВЭП с бестрансформаторным входом» (рис. 2.27, *б*). В этой структуре переменное напряжение первичной сети сначала выпрямляется В1 и фильтруется Ф1, а затем инвертируется в инверторе И в переменное, но с частотой 20—50 кГц. На этой частоте напряжение трансформируется Т до нужного уровня, вновь выпрямляется В2, фильтруется Ф2 и при необходимости стабилизируется Ст. Масса и габариты магнитопровода трансформатора (дросселя) обратно пропорциональны частоте переменного тока. Поэтому на высокой частоте происхо-

дит резкое уменьшение размеров трансформатора и фильтра Ф2. За счет этого, несмотря на большее, по сравнению со структурой рис. 2.27, число узлов и двойное преобразование вида тока, источник получается значительно меньше по массе и габаритным размерам.

2.8.2. Выпрямительные устройства

Выпрямители служат для преобразования переменного (синусоидального) напряжения в напряжение постоянного тока с помощью полупроводниковых выпрямительных диодов.

В схеме **однополупериодного выпрямителя** (рис. 2.28, *а*) ток $I_н$ через диод проходит в нагрузку $R_н$ только в положительные полупериоды напряжения U_2, так как в отрицательные полупериоды оно запирает диод. Ток в нагрузке (заштрихован) имеет прерывистый характер, а его постоянная составляющая I_0 представляет собой среднее значение тока, протекающего за период, и создает на нагрузке постоянную составляющую напряжения, равную (согласно разложению функции в ряд Фурье):

$$U_0 = I_0 R_н \quad \frac{I_{2max}}{\pi} R_н \quad \frac{U_{2max}}{\pi} \quad \frac{\sqrt{2}\cdot U_2}{\pi} \quad 0{,}45U_{\overline{2}}, \qquad \approx$$

где U_{2max} и U_2 — амплитудное и действующее значения синусоидального напряжения вторичной обмотки трансформатора. По заданному U_0 можно найти U_2 и определить необходимый коэффициент трансформации U_1/U_2 при известном напряжении U_1 питающей сети переменного тока.

В отрицательный полупериод к запертому диоду приложено обратное напряжение

$$U_{обр} = U_{2max} = \pi U_0,$$

поэтому диод выбирают так, чтобы его допустимое обратное напряжение удовлетворяло условию $U_{обр.доп} \geqslant \pi U_0$, а допустимый через диод прямой ток — условию $I_{ср.доп} \geqslant I_0$.

Схема **двухполупериодного выпрямителя** состоит из четырех диодов, включенных по так называемой мостовой схеме (рис. 2.28, *б*).

Напряжение U_2 в положительный полупериод открывает диоды *VD*1 и *VD*3 и от точки *A* к точке *B* по цепи «*VD*1—

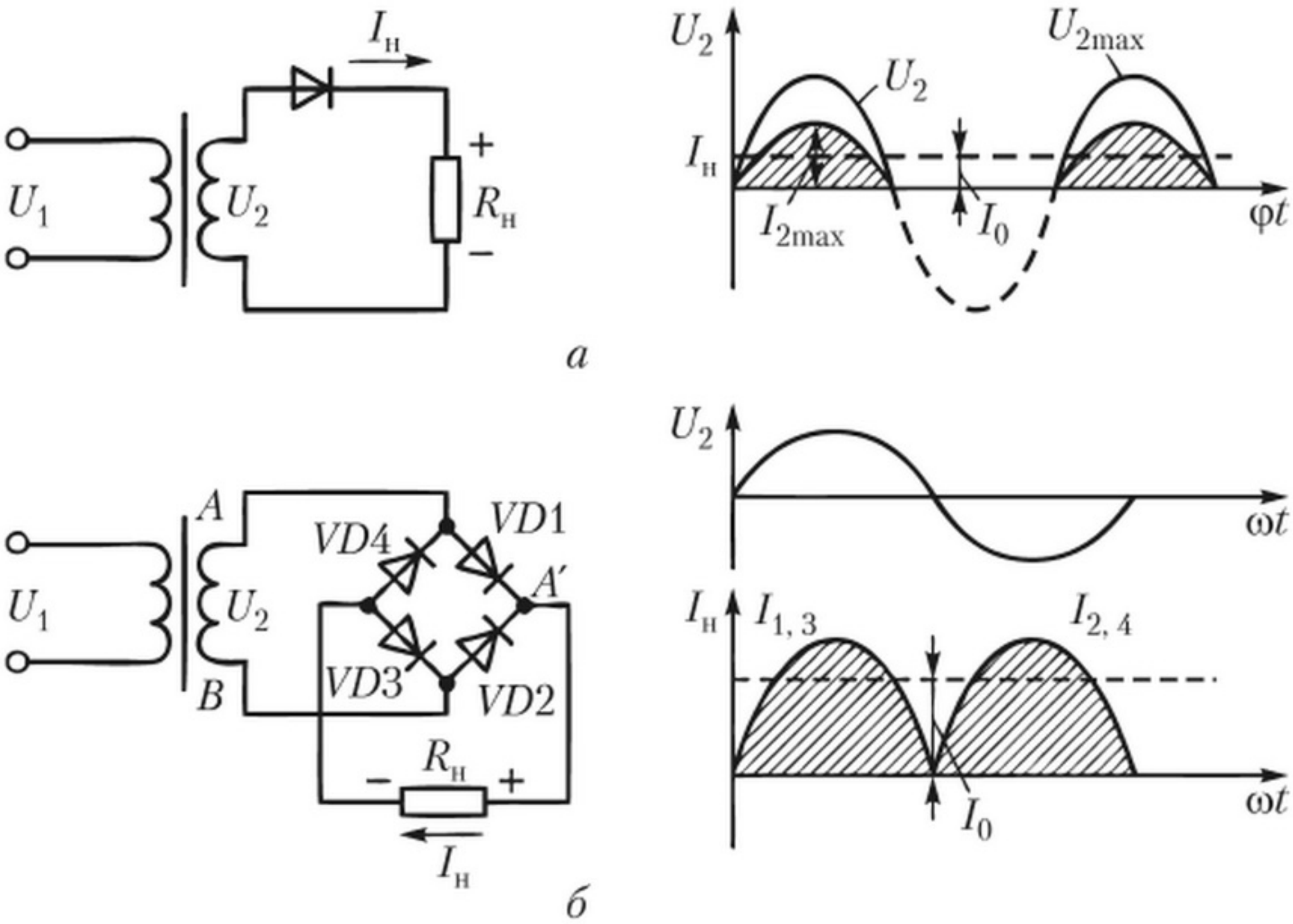


Рис. 2.28. **Выпрямительные устройства и графики, поясняющие их работу:**

а — однополупериодный выпрямитель; *б* — двухполупериодный выпрямитель

$R_н$—*VD*3» протекает ток нагрузки $I_н = I_{1,3}$. При этом диоды *VD*2 и *VD*4 заперты.

В отрицательный полупериод напряжение U_2 открывает диоды *VD*2 и *VD*4 и ток $I_н = I_{2,4}$ течет от точки *B* к точке *A* по цепи «*VD*2—$R_н$—*VD*4», проходя по нагрузке в одном и том же направлении.

В этой схеме постоянные составляющие тока I_0 и напряжения U_0 в два раза выше, чем в однополупериодной схеме:

$$I_0 = \frac{2}{\pi} I_{2max} \quad \text{и} \quad U_0 = \frac{2}{\pi} U_{2max} \quad \frac{2\sqrt{2}}{\pi} U_{\overline{2}} \approx 0{,}9U_2.$$

Оценим обратное напряжение, приложенное, например, к диоду *VD*2 в положительный полупериод напряжения U_2.

При открытом диоде *VD*1 потенциал точки *A*′ близок к положительному потенциалу точки *A* обмотки трансформатора, а ее отрицательный потенциал точки *B* приложен к другому выводу диода *VD*2. Значит, к диоду *VD*2 прило-

жено обратное напряжение вторичной обмотки трансформатора $U_{обр} = U_{2max}$, т.е. такое же, как и в однополупериодной схеме.

Ток, протекающий через каждый из диодов, $I_{ср} = I_0 / 2$, т.е. в два раза меньше, чем в однополупериодном выпрямителе.

Пример 2.1. Найти параметры диодов и коэффициент трансформации трансформатора для мостовой схемы выпрямителя (см. рис. 2.28, *б*), если необходимо получить мощность в нагрузке P_0 = 600 Вт и напряжение U_0 = 115 В при напряжении сети 220 В (нагрузка чисто активная).

Решение. Напряжение вторичной обмотки трансформатора должно быть

$$U_2 = U_0/0{,}9 = 115 / 0{,}9 = 127 \text{ В}.$$

Следовательно, необходим понижающий трансформатор с коэффициентом трансформации K = 220 / 127 = 1,73.

Постоянный ток в нагрузке

$$I_0 = P_0 / U_0 = 600 / 115 = 5{,}2 \text{ А}.$$

Следовательно, необходимы диоды, допустимый ток которых $I_{ср.доп} \geqslant 5{,}2 / 2 = 2{,}6$ А, а допустимое обратное напряжение

$$U_{обр.доп} \geqslant U_{2max} = \sqrt{2} \cdot U_2 = \sqrt{2} \cdot 127 = 180 \text{ В}.$$

Трехфазные выпрямители применяют при больших мощностях, так как они равномерно нагружают трехфазную сеть.

Вторичная обмотка трансформатора, соединенная звездой с нулевым проводом, подключена к нагрузке $R_н$ через три диода (рис. 2.29, *а*). Ток через каждый диод протекает в течение 1/3 периода T, когда напряжение фазной обмотки, с которой соединен диод, выше напряжения обмоток двух других фаз. На рис. 2.29, *в* заштрихована форма напряжения, подаваемого к нагрузке, и имеющая значительно меньшие пульсации, чем в схемах рис. 2.28.

В трехфазном выпрямителе соотношения для чисто активной нагрузки таковы:

$$U_0 = \frac{3\sqrt{3}}{2\pi} U_{2max} \quad 0{,}827 U_{2max} \quad 1{,}17 U_2;\ U_{обр} \cdot \sqrt{3}\ U_{2max} \quad 2{,}09 U_0,$$

Содержание

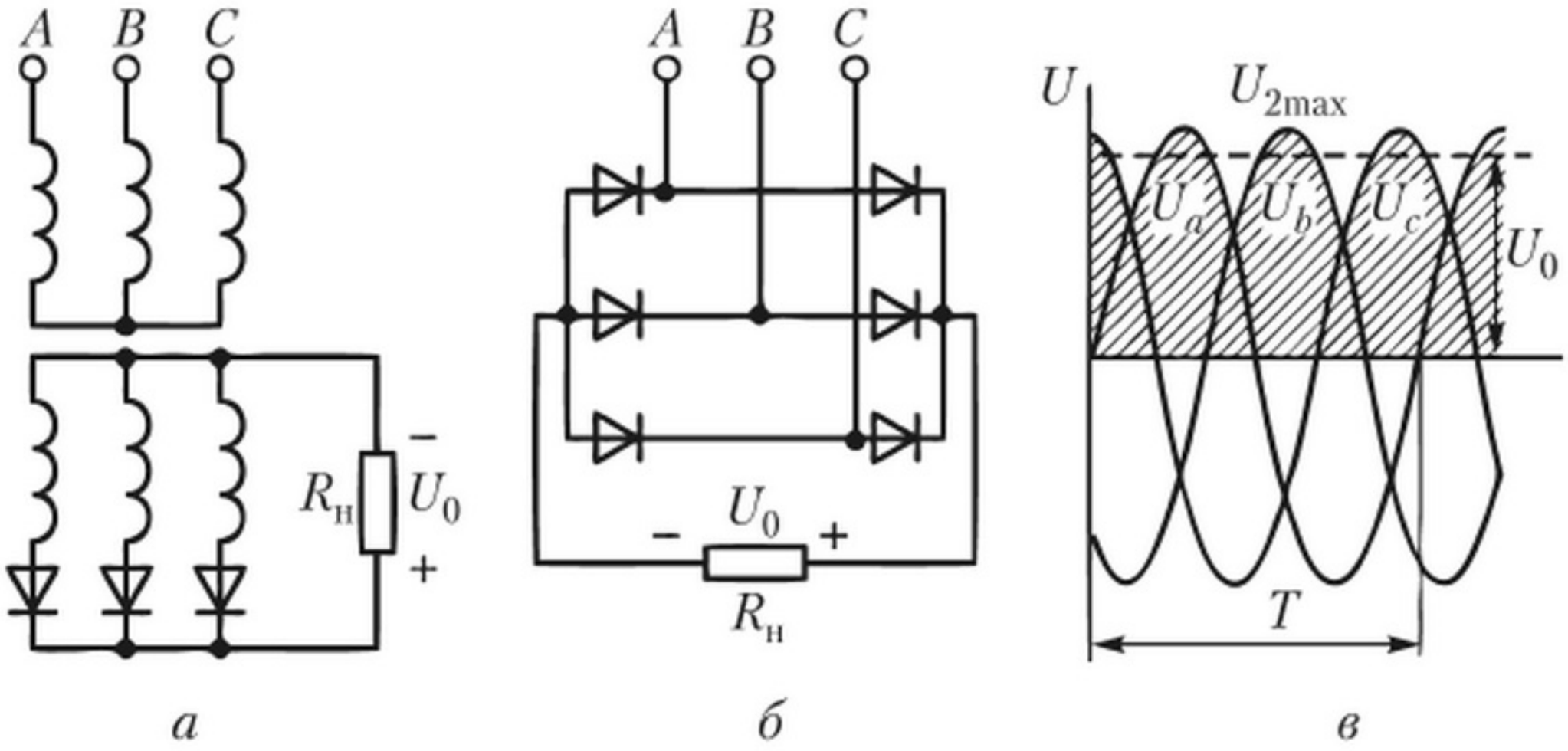


Рис. 2.29. **Трехфазные выпрямители:**
а — трансформаторный; *б* — бестрансформаторный;
в — график, поясняющий работу выпрямителя

где U_{2max} и U_2 — амплитудное и действующее значения фазного напряжения, а средний ток через диод $I_{ср} = I_0 / 3$.

В *бестрансформаторном трехфазном выпрямителе* (рис. 2.29, *б*) соотношения для чисто активной нагрузки равны:

$$U_0 = \frac{3\sqrt{2}}{\pi} U \quad 1{,}35U \quad U_{обр} \quad U_{max} \quad 1{,}05U_0,$$

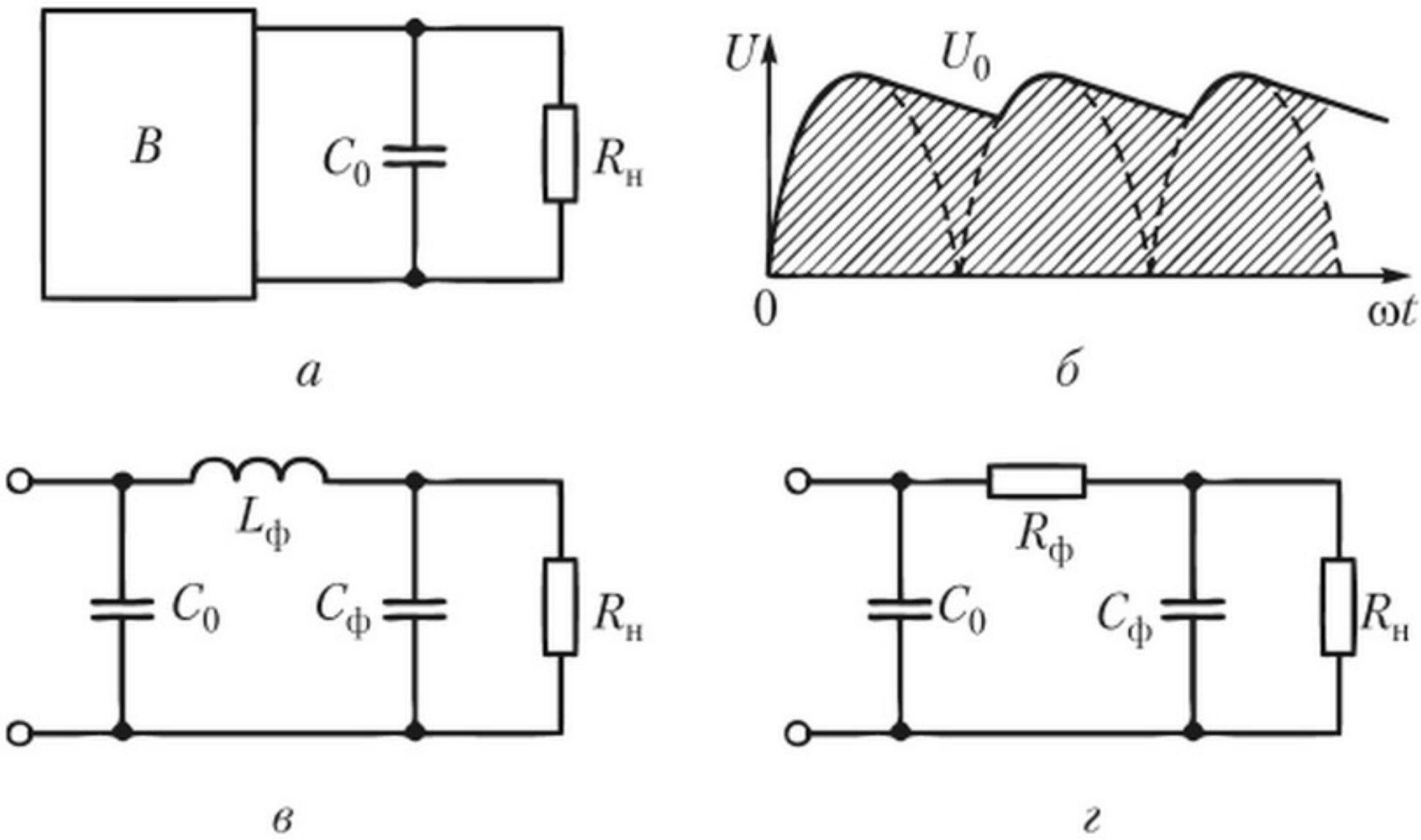
где U_{max} и U — амплитудное и действующее значения линейного напряжения сети, а средний ток через каждый из шести диодов $I_{ср} = I_0 / 3$.

Пример 2.2. Чему равно выпрямленное напряжение бестрансформаторного трехфазного выпрямителя (см. рис. 2.29, *б*), подключенного к сети с линейным напряжением $U = 380$ В, и на какое обратное напряжение должны быть рассчитаны диоды?

Решение. Постоянное напряжение равно $U_0 \approx 1{,}35 \cdot 380 = 513$ В и диоды должны быть рассчитаны на обратное напряжение $U_{обр} = U_{max} \quad \sqrt{2}\ U \cdot \quad \sqrt{2}\ 380 \quad 537$ В.

2.8.3. Сглаживающие фильтры

Для питания электронной аппаратуры допускается пульсация напряжения, не превышающая долей процента, однако на выходе выпрямителей пульсации значительно



а *б* *в* *г*

Рис. 2.30. **Сглаживающие фильтры:**
а — *RC*-фильтр; *б* — график, поясняющий работу фильтра; *в* — *LC*-фильтр; *г* — П-образный *RC*-фильтр

больше. Для их уменьшения применяют сглаживающие фильтры, которые должны максимально уменьшить (подавить) переменные составляющие и с возможно меньшими потерями пропустить постоянную составляющую выпрямленного напряжения.

Простейшим фильтром служит конденсатор, включенный на выходе выпрямителя *B* параллельно нагрузке (рис. 2.30, *а*), который запасает энергию, заряжаясь во время возрастания напряжения выпрямителя, и отдает ее, разряжаясь на сопротивление нагрузки, когда оно снижается. На рис. 2.30, *б* показана форма напряжения на конденсаторе U_C (а значит, и на параллельно включенном $R_н$) при двухполупериодном выпрямителе.

Для дальнейшего снижения пульсаций применяют Г-образные *LC*-фильтры (рис. 2.30, *в*). Индуктивное сопротивление $X_L = \omega L_ф$ стремятся сделать значительно больше $R_н$ для того, чтобы переменные составляющие выпрямленного напряжения с частотами пульсаций от основной ω и выше «задерживались» бы фильтром в виде падения напряжения на X_L, не достигая нагрузки. Емкостное же сопротивление $X_C = 1/\omega C_ф$ выполняют значительно меньше, чем $R_н$, для того, чтобы переменные составляющие выпрямленного тока замыкались через X_C, минуя $R_н$. При этом постоянная составляющая тока, для которой $X_L = 0$, $X_C = \infty$, не создает

падения напряжения на $L_ф$ и не замыкается через $C_ф$, целиком поступая в нагрузку.

Недостатком LC-фильтров является громоздкость и трудность изготовления индуктивности в микроэлектронном исполнении. Поэтому в интегральных микросхемах при токах нагрузки в несколько миллиампер используют П-образные RC-фильтры (рис. 2.30, *г*), несмотря на их несколько худшие сглаживающие свойства и меньшие КПД.

2.8.4. Линейные стабилизаторы напряжения

Стабилизатором напряжения называется устройство, автоматически поддерживающее напряжение на нагрузке при изменении в определенных пределах таких дестабилизирующих факторов, как напряжение первичного источника, сопротивление нагрузки, температура окружающей среды.

Существует два вида стабилизаторов — параметрические и компенсационные.

Параметрический стабилизатор использует элементы, в которых напряжение остается неизменным при изменении протекающего через них тока. Такими элементами являются стабилитроны, в которых при изменении тока в очень широких пределах падение напряжения изменяется на доли процента (см. параграф 1.2). Параметрические стабилизаторы применяются, как правило, в качестве источников опорного (эталонного) напряжения в мощных компенсационных стабилизаторах (рис. 2.31).

Принцип работы *компенсационного стабилизатора* основан на сравнении фактического напряжения на нагрузке

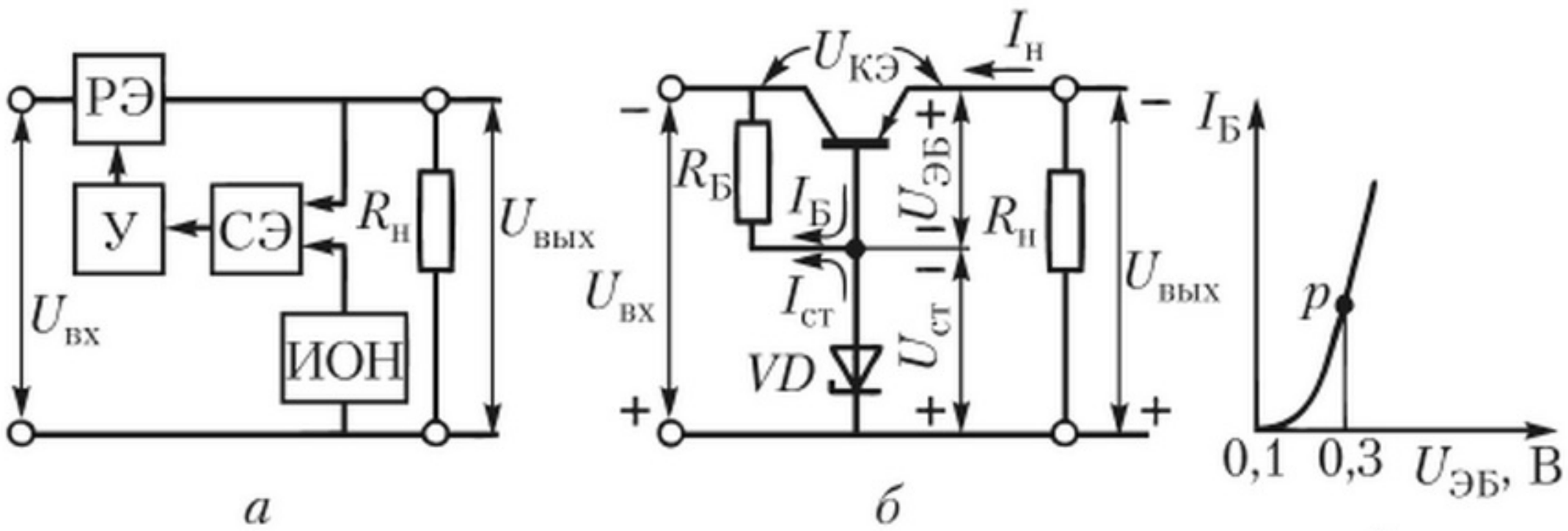


Рис. 2.31. **Структура компенсационного стабилизатора напряжения (*а*), его простейшая реализация (*б*) и график, поясняющий выбор рабочей точки (*в*)**

Содержание

с эталонным и увеличении или уменьшении в зависимости от этого отклонения выходного напряжения. Эталонное напряжение формируется источником опорного напряжения (ИОН). В сравнивающем элементе (СЭ) происходит сравнение напряжения на нагрузке с эталонным и выработка управляющего сигнала рассогласования. Этот сигнал усиливается усилителем (У) и подается на регулирующий элемент (РЭ), который обеспечивает такое изменение выходного напряжения, которое приводит к приближению фактического напряжения на нагрузке к эталонному значению.

Основным параметром стабилизатора является *коэффициент стабилизации* — отношение относительного изменения напряжения на входе к относительному изменению напряжения на выходе:

$$K_{ст} = \frac{\Delta U_{вх} / U_{вх}}{\Delta U_{вых} / U_{вых}}.$$

В простейшем компенсационном стабилизаторе опорным напряжением является напряжение $U_{ст}$ стабилитрона *VD*, а сравнивающим элементом, усилителем и одновременно регулирующим элементом — транзистор (см. рис. 2.31, *б*).

Выходное напряжение (как это видно по знакам «+» и «−» на схеме) $U_{вых} = U_{ст} - U_{ЭБ}$. Ток через резистор $R_Б$ образуется сложением двух токов: тока стабилитрона $I_{ст}$ и тока базы $I_Б$. Режим работы транзистора выбирают таким образом, чтобы исходная рабочая точка *p* располагалась на середине линейного участка его входной характеристики (см. рис. 2.31, *в*). Напряжение $U_{ЭБ}$ при этом составляет 0,—0,3 В. Так как напряжение стабилитрона обычно около 8 В, то $U_{вых} \approx U_{ст}$.

Предположим, что по каким-либо причинам напряжение на нагрузке уменьшилось. Это приведет к увеличению падения напряжения $U_{ЭБ} = U_{ст} - U_{вых}$, что, в свою очередь, увеличит степень открытия транзистора. В результате падение напряжения на транзисторе $U_{КЭ}$ уменьшится, а значит, увеличится напряжение на нагрузке $U_{вых} = U_{вх} - U_{КЭ}$, и в итоге напряжение на нагрузке восстановится. Аналогичное восстановление выходного напряжения произойдет и при его увеличении. Только в этом случае произойдет уменьшение

степени открытия транзистора и соответствующее увеличение падающего на нем напряжения $U_{КЭ}$.

Транзистор включен по схеме эмиттерного повторителя, входным напряжением которого является $U_{ст}$. Так как $I_Б << I_н$, схема позволяет отдавать в нагрузку значительную мощность. Коэффициент стабилизации такой схемы составляет $K_{ст}$ = 150—300. В рассмотренной схеме сигнал рассогласования формируется на самом регулирующем транзисторе. Более высокую степень стабилизации обеспечивают схемы, в которых на базу регулирующего транзистора поступает предварительно усиленный сигнал рассогласования.

В рассмотренных стабилизаторах напряжения регулирующий транзистор всегда открыт, а саморегулирование осуществляется путем изменения степени его открытия, т.е. линейно. Поэтому такие стабилизаторы называются *линейными*.

2.8.5. Импульсные стабилизаторы напряжения

В отличие от ранее рассмотренных линейных стабилизаторов, в *импульсных стабилизаторах напряжения* транзистор, через который течет ток в нагрузку, периодически открывается и закрывается, т.е. работает в ключевом режиме. Причем регулирование осуществляется путем изменения паузы, в течение которой через ключевой транзистор течет ток нагрузки от источника постоянного входного напряжения. Таким образом, транзистор работает не в линейном, а в импульсном режиме: он либо полностью открыт, либо полностью закрыт. В таких стабилизаторах среднее значение напряжения на нагрузке $U_{вых} = U_{вх}t / T$, где T — период следования импульсов; t — продолжительность замкнутого состояния ключа.

Импульсные стабилизаторы обеспечивают более высокий КПД, так как в полностью открытом состоянии на транзисторе падает очень небольшое напряжение, а следовательно, мощность, рассеиваемая на транзисторе, гораздо меньше той мощности, которая рассеивается в линейных стабилизаторах.

Поскольку регулирование осуществляется путем изменения ширины импульсов t, этот принцип работы получил название широтно-импульсной модуляции (ШИМ). Импульсные стабилизаторы (рис. 2.32), так же как и линейные, являются компенсационными. Сигнал рассогласо-

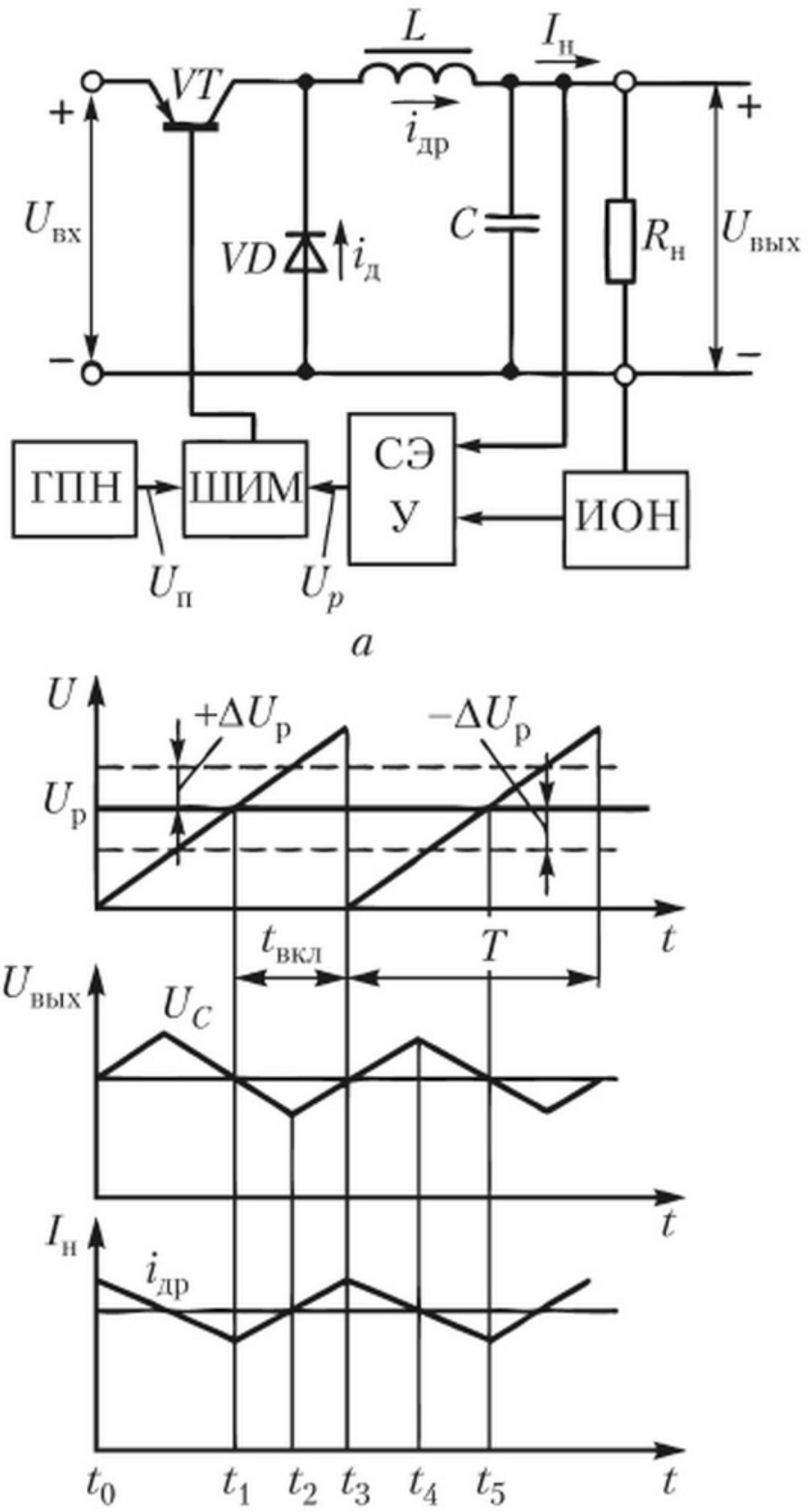


Рис. 2.32. **Импульсный стабилизатор напряжения (*а*) и происходящие в нем процессы (*б*)**

вания $U_р$, образованный сравнивающим элементом СЭ и усиленный усилителем (У), преобразуется в импульсы, следующие с одинаковой частотой, длительность которых $t_{вкл}$ изменяется под действием сигнала рассогласования. Эти импульсы открывают и закрывают ключевой транзистор *VT*, который вместе с диодом *VD* и *LC*-фильтром образует импульсный регулирующий элемент.

Пока напряжение пилообразного напряжения $U_п < U_р$ (участок $t_0 - t_1$ на рис. 2.32, *б*), транзистор заперт. В течение

времени $t_1 - t_3$, когда $U_п > U_р$, транзистор открыт, и напряжение $U_{вх}$ приложено к дросселю. Под действием $U_{вх}$ диод VD запирается, а ток через дроссель $i_{др}$ увеличивается, запасая энергию в индуктивности. До тех пор, пока ток дросселя не достигнет значения постоянного тока нагрузки $I_н$ (участок $t_1 - t_2$), конденсатор C разряжается на нагрузку и напряжение на нем U_C снижается. С момента времени t_2, когда $i_{др} > I_н$, конденсатор начнет подзаряжаться разностью токов $i_{др} - I_н$. В момент t_3 запирания транзистора ЭДС самоиндукции дросселя открывает диод и ток дросселя, замыкаясь через диод, протекает по нагрузке и до момента t_4 продолжает заряжать конденсатор, отдавая ему запасенную дросселем энергию. На участке $t_4 - t_5$ ток дросселя меньше тока нагрузки и нагрузка подпитывается током разрядки конденсатора. С момента t_5 процесс повторяется.

Пусть выходное напряжение станет меньше заданного значения и напряжение рассогласования $U_р$ уменьшится на величину $-\Delta U_р$. Тогда момент времени, когда пилообразное напряжение, формируемое ГПН, станет равным напряжению $U_р$, наступит раньше, а время открытого состояния транзистора $t_{вкл}$, формируемого ШИМ, увеличится. Это приведет к возрастанию выходного напряжения $U_{вых}$ и восстановлению его заданного значения. Если же выходное напряжение увеличится, напряжение рассогласования также увеличится на величину $+\Delta U_р$. Это приведет к тому, что момент открытия транзистора, формируемый ШИМ, наступит позже, а время открытого состояния транзистора $t_{вкл}$ уменьшится. В результате выходное напряжение уменьшится и его заданное значение восстановится.

2.9. Силовые устройства на основе тиристоров и мощных транзисторов

2.9.1. Управляемые выпрямители

К силовым устройствам относят такие электронные устройства, которые обеспечивают преобразование энергии в электрических цепях, токи в которых измеряются десятками, сотнями и даже тысячами ампер, а величины напряжения — сотнями и тысячами вольт. Такие устройства чаще всего выполняются на основе тиристоров, которые, по существу, являются мощными управляемыми ключами.

В отличие от ключей на полевых транзисторах, тиристорный ключ пропускает ток только в прямом направлении, как и обычный выпрямительный диод, но для этого он должен быть открыт подачей импульса напряжения на управляющий электрод.

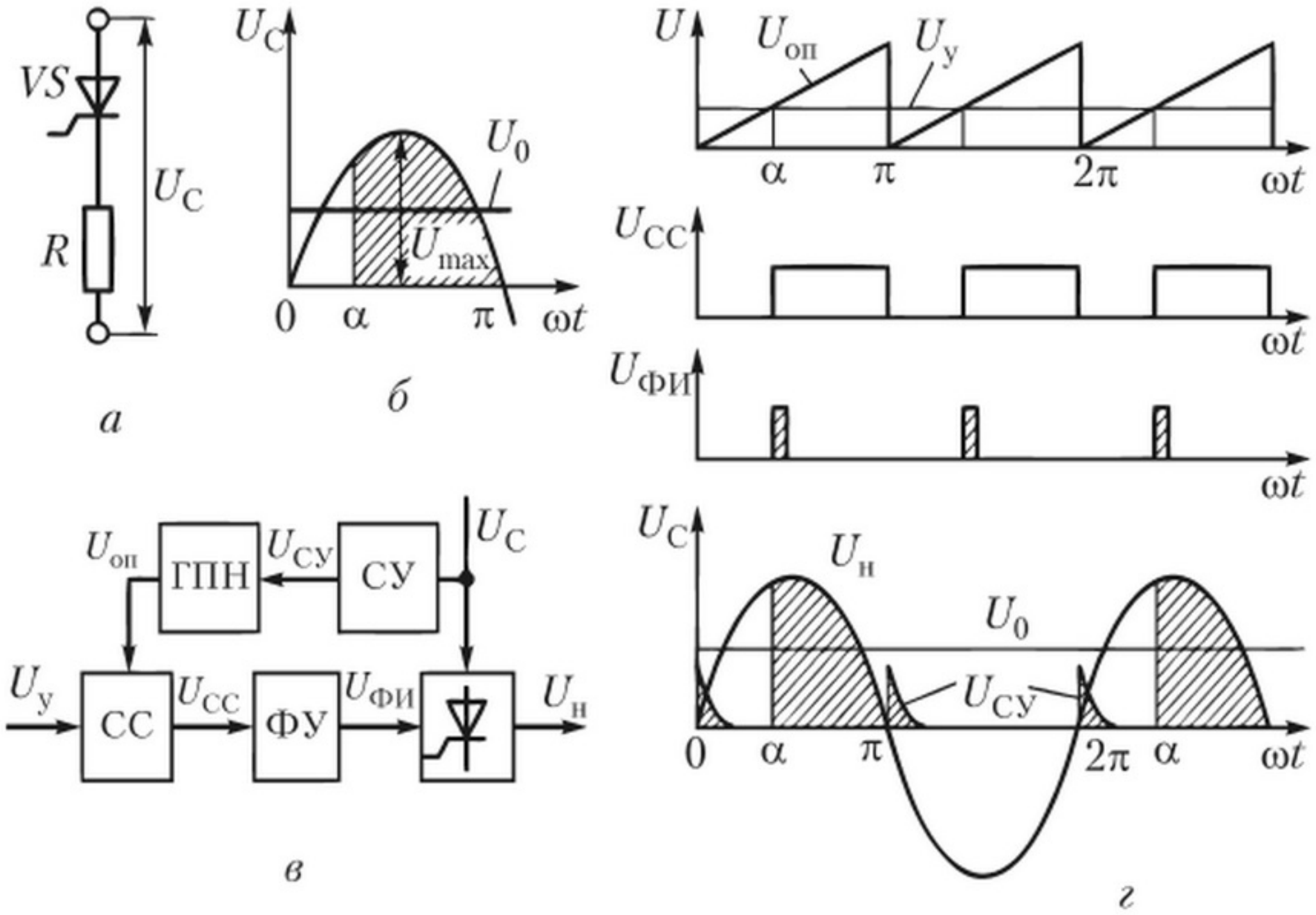
Тиристоры используются как для преобразования переменного напряжения в постоянное, так и наоборот. К первым относят управляемые выпрямители, ко вторым — инверторы. Управляемые выпрямители на основе тиристоров позволяют изменять постоянную составляющую напряжения U_0 от нуля до максимального значения, определяемого формулами для диодных выпрямителей, при неизменном напряжении сети переменного тока. Для этого в схемах рис. 2.28 и 2.29 выпрямительные диоды заменяют тиристорами (см. параграф 1.5).

Рассмотрим принцип работы таких схем на примере однополупериодного выпрямителя (см. рис. 2.28, *а*).

К синусоидальному напряжению сети U_C с амплитудой U_{max} подключены нагрузка R и тиристор VS (рис. 2.33, *а*). Тиристор открывается в момент времени, определяемый подачей на управляющий электрод импульса напряжения от схемы управления (если полупериод соответствует π, то этот момент времени соответствует углу α). В результате в течение интервала α—π к нагрузке подводится напряжение, заштрихованное на рис. 2.33, *б*, и по ней протекает ток.

В момент времени $\omega t = \pi$ ток уменьшается до нуля и тиристор запирается. Этот процесс повторяется каждый положительный полупериод (в отрицательные полупериоды тиристор заперт напряжением сети).

Структурная схема управления однополупериодного тиристорного выпрямителя показана на рис. 2.33, *в*, а графики, поясняющие работу выпрямителя, — на рис. 2.33, *г*. Напряжение сети поступает на синхронизирующее устройство (СУ), которое в моменты перехода напряжения через нуль осуществляет запуск ГПН. Таким образом, в начале каждого полупериода напряжения сети U_C ГПН формирует линейно нарастающее опорное напряжение $U_{оп}$. Это напряжение поступает на один из входов схемы сравнения СС, выполняемой на основе компаратора. На другой вход СС поступает управляющее напряжение $U_у$ постоянного тока. В течение времени, пока напряжение $U_{оп}$ меньше напряжения $U_у$, на выходе СС низкий уровень напряжения $U_{СС}$,



а *б* *в* *г*

Рис. 2.33. **Однополупериодный тиристорный выпрямитель:**
а — схема подключения; *б* — принцип работы;
в — структурная схема; *г* — графики, поясняющие работу выпрямителя

когда же напряжение $U_{оп}$ больше напряжения $U_у$, на выходе СС высокий уровень напряжения $U_{СС}$. По переднему фронту импульса $U_{СС}$ формирователь импульсов (ФИ) формирует импульсы напряжения $U_{ФИ}$, которые подаются на УЭ тиристора и открывают его.

Изменяя $U_у$, обеспечивают изменение угла α, т.е. момента открытия тиристора, и, таким образом, изменение времени, в течение которого через нагрузку течет ток. В результате изменяется и постоянная составляющая напряжения U_0, равная среднему за период напряжению на нагрузке:

$$U_0 = \frac{1}{2\pi}\int_{\alpha}^{\pi} U_{max}\sin(\omega t)d\omega t = \frac{1}{2\pi}U_{max}(1 \quad \cos \quad). + \qquad \alpha$$

Напряжение U_0 увеличивается вдвое при использовании двухполупериодного (см. рис. 2.28, *б*) управляемого выпрямителя. Так как ток нагрузки в один полупериод протекает через диоды *VD*1, *VD*3, а в другой — через *VD*2, *VD*4, для управления током в каждом полупериоде достаточно одного тиристора. Таким образом, управляемый выпрями-

тель можно построить на основе мостовой схемы, заменив два диода, например *VD*1 и *VD*2, на тиристоры.

Аналогично получают и трехфазные управляемые выпрямители, заменяя в схемах (см. рис. 2.29, *а* и *в*) диоды на тиристоры.

Такие управляемые выпрямители используют для регулирования частоты вращения двигателей постоянного тока в электроприводах металлорежущих станков и транспортных средств.

2.9.2. Инверторы и преобразователи частоты

Инверторы служат для преобразования энергии постоянного тока в энергию переменного тока требуемой частоты.

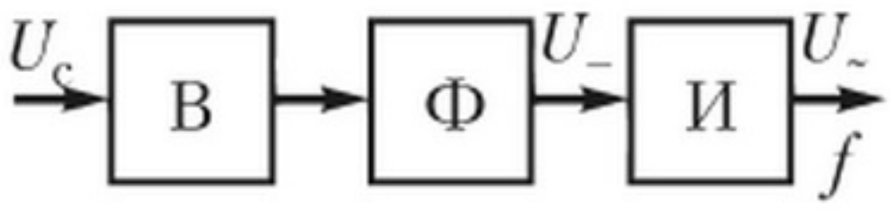
В качестве переключающих приборов в сильноточных инверторах применяют тиристоры. В цепях с относительно небольшими значениями протекающих токов могут использоваться мощные полевые или биполярные транзисторы.

На рис. 2.34, *а* приведена структурная схема преобразователя частоты с использованием инвертора для питания трехфазной нагрузки, соединенной звездой, переменным трехфазным напряжением с регулировкой его значения U и частоты f. В качестве силовых переключающих приборов применены мощные биполярные транзисторы.

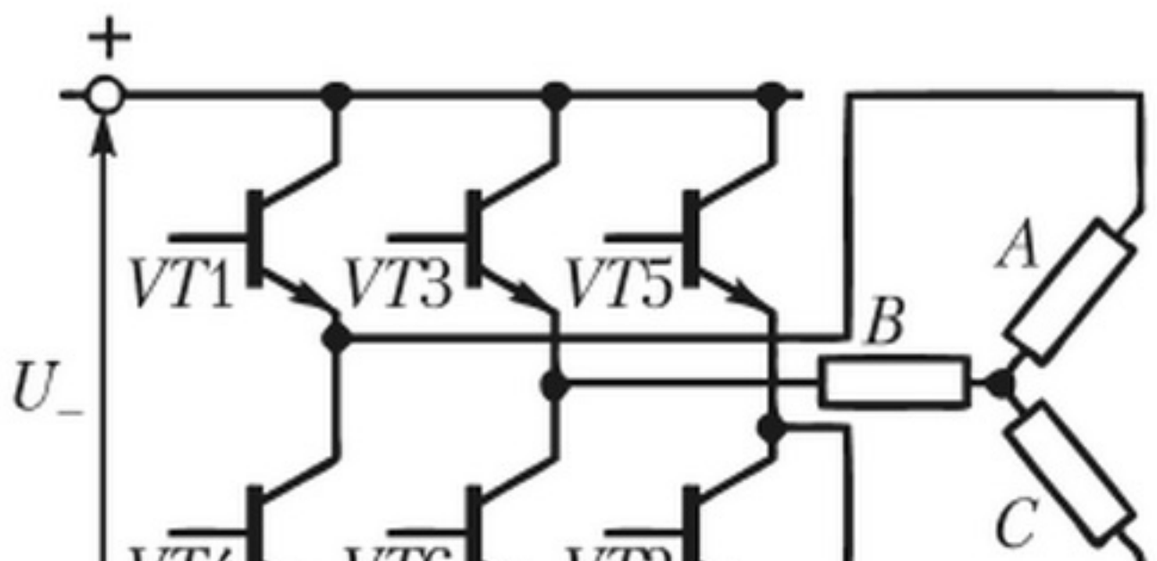
Напряжение с частотой промышленной сети U_C преобразуется сначала управляемым выпрямителем (В) с фильтром (Ф) в напряжение постоянного тока $U_{_}$ требуемого значения. Затем это напряжение поступает на инвертор (И) (рис. 2.34, *б*), состоящий из шести транзисторов *VT*1–*VT*6, пронумерованных в порядке последовательности их включения, показанной на рис. 2.34, *в*. Каждый транзистор открывается на время τ, равное длительности одного полупериода $T/2$ требуемого переменного напряжения.

На рис. 2.34, *г* приведено ступенчатое напряжение, формируемое на фазе *A* нагрузки, а на рис. 2.35 пояснен процесс его формирования.

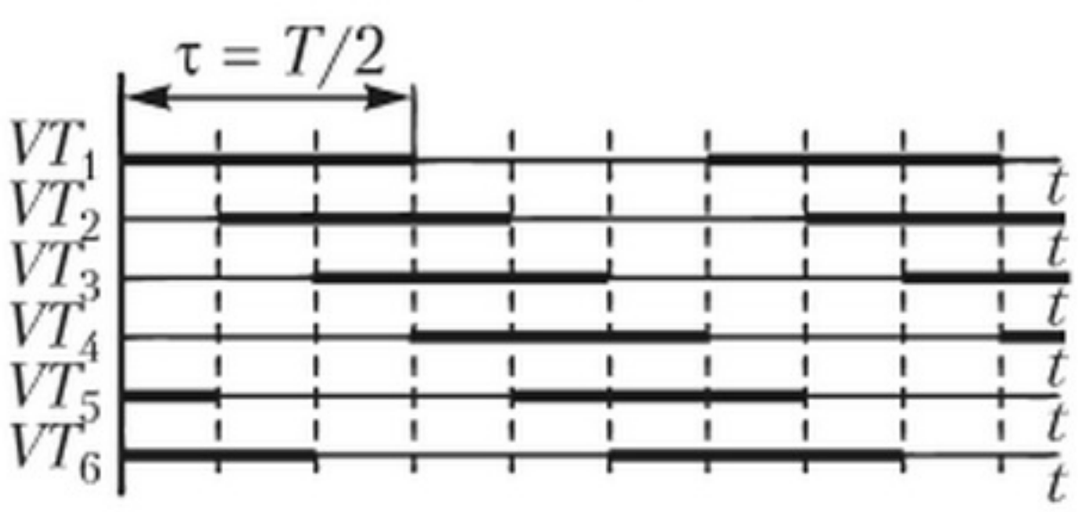
На схемах рис. 2.35 в виде замкнутых ключей изображены лишь открытые транзисторы для шести последовательных состояний инвертора, соответствующих временным диаграммам рис. 2.34, *в*. Из анализа этих схем очевидно, что, когда фаза *A* включена параллельно *B* или *C*, на ней выделяется одна треть напряжения $U_{=}$, а когда



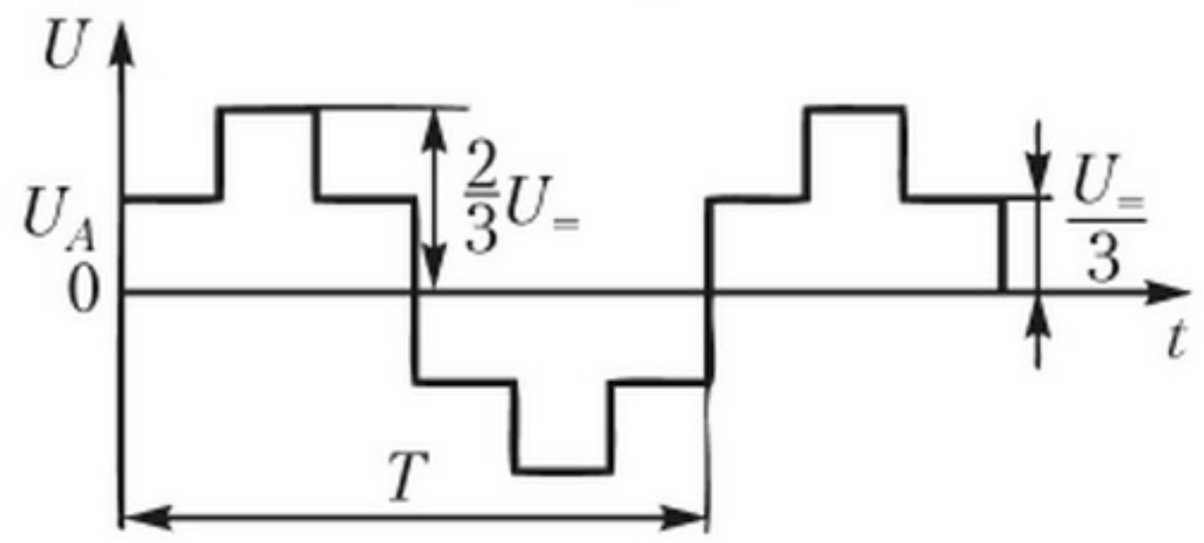
а



б



в



г

Рис. 2.34. **Преобразователь частоты на базе инвертора:**
а — структурная схема; *б* — схема инвертора; *в* — временная диаграмма; *г* — график выходного напряжения

фаза *A* оказывается включенной последовательно с параллельно соединенными фазами *B* и *C*, на ней выделяется две трети напряжения $U_{=}$.

При этом в первых трех состояниях напряжение на фазе *A* соответствует положительному, а в последних двух — отрицательному полупериоду приложенного к ней напряжения переменного тока ступенчатой формы.

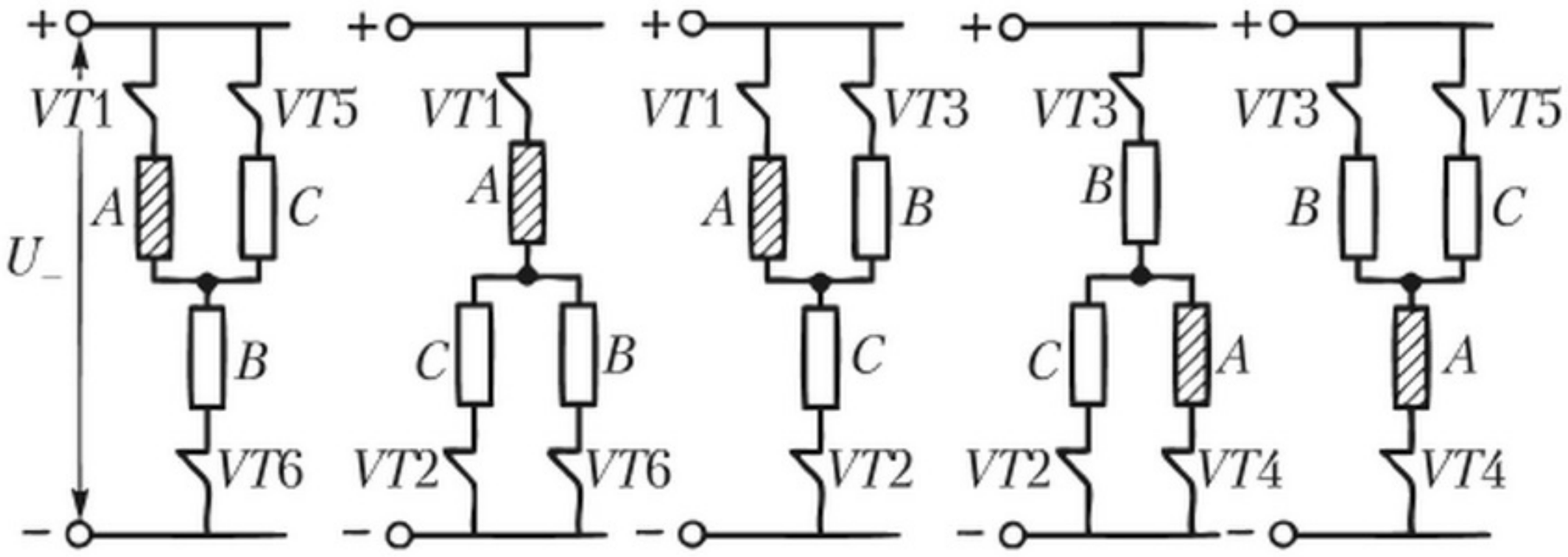


Рис. 2.35. **Процесс формирования выходного напряжения преобразователя частоты**

Рассуждая подобным образом, можно убедиться, что к фазам *B* и *C* будет приложено такое же, как к фазе *A*, напряжение, но сдвинутое соответственно на одну треть и две трети периода *T*, образуя трехфазную систему напряжений.

Изменяя с помощью схемы управления длительность α открытого состояния транзистора, можно в широких пределах регулировать частоту формируемого трехфазного напряжения, поэтому такие инверторы применяют для плавного регулирования частоты вращения трехфазных асинхронных двигателей.

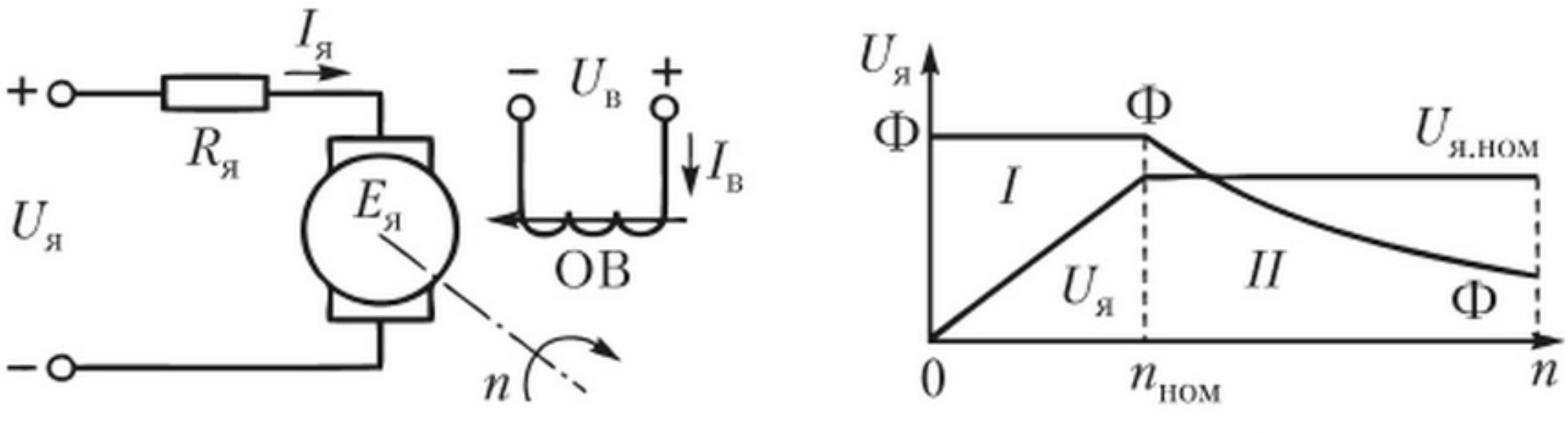
2.9.3. Тиристорное управление двигателем постоянного тока

В приводах главного движения и подачи инструмента металлорежущих станков широко используются двигатели постоянного тока с независимым возбуждением, которые способны обеспечивать регулирование скорости вращения в широких пределах. Такой двигатель (рис. 2.36, *а*) состоит из статора, на полюсах которого намотаны обмотки возбуждения (ОВ), и ротора, называемого якорем.

Ток возбуждения $I_в$, проходя по ОВ под действием напряжения $U_в$, создает магнитный поток Ф. К якорю через щетки подводится напряжение якоря $U_я$, создающее ток якоря $I_я$. Протекая по виткам обмотки якоря, ток $I_я$, взаимодействуя с потоком Ф, создает вращающий момент $M_{вр}$

$$M_{вр} = K\Phi I_я,$$

где K — коэффициент, зависящий от конструкции двигателя (размеров, числа витков обмоток и т.п.).



а *б*

Рис. 2.36. **Схема подключения двигателя постоянного тока (*а*) и график, поясняющий принципы управления скоростью вращения двигателя (*б*)**

При вращении двигателя в обмотке якоря наводится ЭДС $E_я$, направленная согласно правилу Ленца встречно приложенному напряжению $U_я$ и пропорциональная числу оборотов вала двигателя n:

$$E_я = cn\Phi,$$

где c — коэффициент, зависящий от конструкции двигателя.

Для цепи якоря при равномерном вращении вала по второму закону Кирхгофа можно записать равенство:

$$U_я = E_я + R_я I_я,$$

где $R_я$ — активное сопротивление обмотки якоря, включая сопротивление контакта щетки — коллектор (в двигателях коллектором называют набор контактных площадок, через которые с щеток подается напряжение в обмотку вращающегося якоря).

Подставив в это равенство выражение $E_я$ и значение тока $I_я$, полученное из выражения для вращающего момента, получим:

$$U_я = cn\Phi \quad \frac{R_я}{K\Phi} + M_{вр},$$

откуда число оборотов двигателя равно:

$$n = \frac{U_я}{c\Phi} \quad \frac{R_я}{cK\Phi^2} M_{вр}.$$

Из полученной формулы очевидны два способа (две зоны) управления скоростью вращения двигателя

(рис. 2.36, *б*). В зоне *I* скорость изменяют от 0 до номинального значения $n_{ном}$, увеличивая напряжение $U_я$ при неизменном значении магнитного потока Ф, а значит, и неизменном напряжении возбуждения $U_в$. При достижении напряжением $U_я$ номинального значения дальнейшее его увеличение невозможно, так как может привести к пробою изоляции. В то же время для быстрого перемещения, например, инструмента на холостом ходу или ускоренного вращения шпинделя необходимо увеличить скорость вращения двигателя в 3—5 раз выше $n_{ном}$. Для этого используют зону *II*, в которой при неизменном напряжении $U_{я.ном}$ уменьшают значение магнитного потока Ф с помощью соответствующего понижения $U_в$, а значит, и тока возбуждения $I_в$. Заметим, однако, что в зоне *II* приходится мириться с соответствующим понижением и вращающего момента двигателя, т.е. нагружать двигатель меньшим моментом сопротивления, который он должен преодолевать своим вращающим моментом. Действительно, как это следует из формулы для $M_{вр}$, при уменьшении потока Ф вращающий момент снижается, а скомпенсировать его увеличением тока $I_я$ нельзя, так как это приведет к перегреву двигателя.

В выпускаемых промышленностью тиристорных преобразователях регулирование скорости вращения в зоне *I* осуществляется применением двух управляемых мощных (до нескольких десятков киловатт) трехфазных выпрямителей (на рис. 2.37 они обведены пунктиром).

Один из трехфазных выпрямителей обеспечивает правое направление вращения двигателя, а другой — левое, изме-

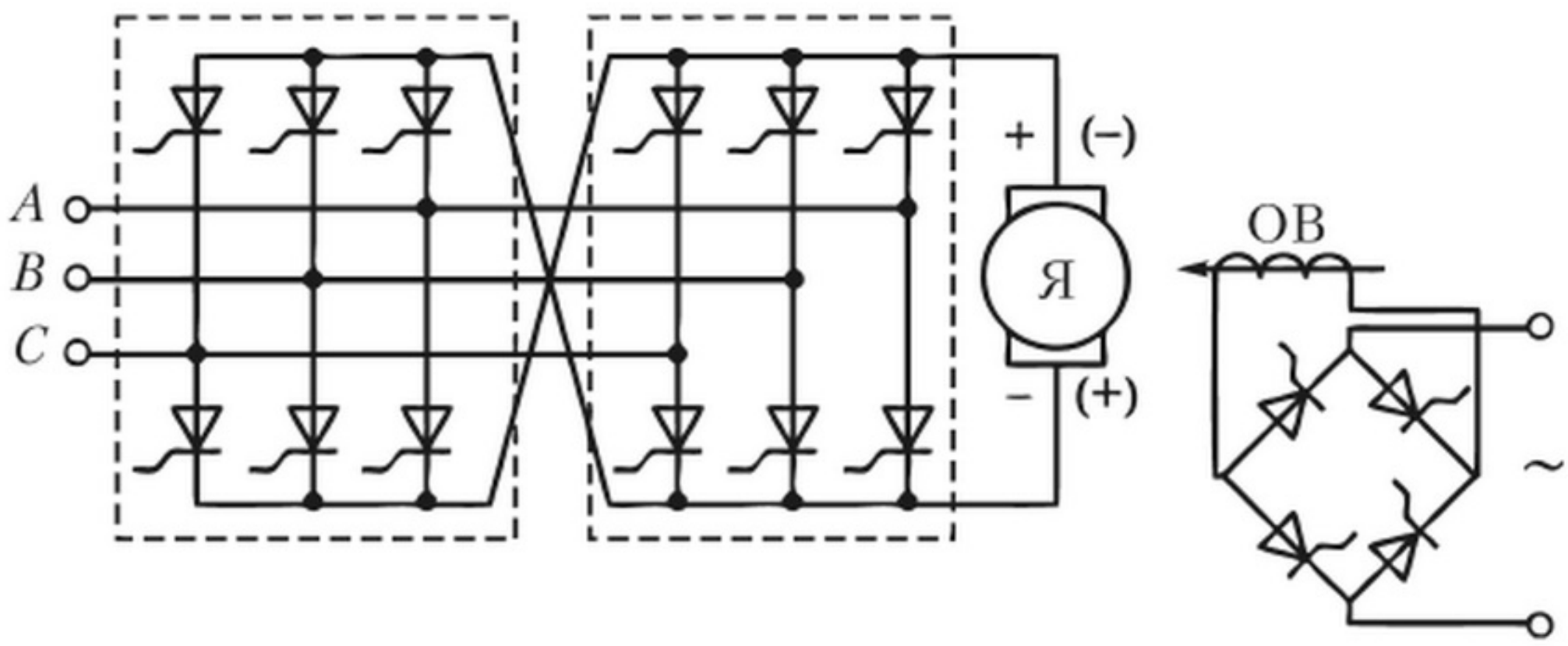


Рис. 2.37. **Схема тиристорного регулятора скорости вращения двигателя**

Содержание

няя полярность $U_я$ на противоположную. Естественно, что эти выпрямители должны работать раздельно во избежание короткого замыкания между ними, что и обеспечивает схема управления выпрямителями, разрешая включение одного из них лишь через несколько миллисекунд после отключения другого. Схемы управления тиристорами выполнены по принципу, рассмотренному в параграфе 2.9 и на рис. 2.33.

Для управления скоростью вращения в зоне *II* используется однофазная мостовая схема тиристорного выпрямителя, обеспечивающая питание ОВ. Схема позволяет лишь уменьшать значение тока возбуждения $I_в$, сохраняя его полярность. Схема управления тиристорами мостовой схемы также выполнена по принципу, рассмотренному в параграфе 2.9 и на рис. 2.33.

2.10. Электромагнитная совместимость электронных устройств

Работа силовых электронных устройств, как правило, сопряжена с резкими скачками протекающего по цепям электрического тока. Такие скачки могут происходить при коммутациях цепей питания электронных устройств, включениях исполнительных устройств — двигателей, пускателей; при резких изменениях тока нагрузки. Импульсы тока приводят к скачкам напряжения в цепях питания, а также к появлению электромагнитных помех, распространяемых через эфир. Эти помехи оказывают отрицательное влияние на работу других электронных устройств. У аналоговых устройств — усилителей, преобразователей — снижается точность их работы. В импульсных и цифровых устройствах такие импульсные помехи воспринимаются как ложные сигналы, которые приводят к сбоям в работе.

2.10.1. Помехи в цепях питания

Наиболее сильное влияние на работу электронных устройств оказывают помехи, проходящие по цепям питания. Как было показано в параграфе 2.8, существует две основные структуры источников электропитания — линейные ИВЭП (см. рис. 2.27, *а*), подключаемые к первичному питанию через трансформатор, и ИВЭП с бестрансформаторным входом (см. рис. 2.27, *б*). В линейных ИВЭП регули-

рующий элемент работает в линейном режиме, увеличивая или уменьшая ток, поступающий в нагрузку. Сами по себе эти ИВЭП не создают мощных помех. Однако скачки тока в нагрузке и создаваемые ими помехи будут передаваться через ИВЭП в первичную сеть, через которую могут воздействовать на работу других устройств. Величина этих помех, однако, не велика, так как входной развязывающий трансформатор существенно их ослабляет. Кроме того, развязывающий трансформатор может дополняться специальными емкостной и компенсационной обмотками, которые дополнительно снижают как влияние флуктуаций входного первичного напряжения на работу ИВЭП, так и выбросов и провалов выходного напряжения ИВЭП на первичную сеть.

По-иному обстоит дело с использованием ИВЭП с бестрансформаторным входом. Применение этих ИВЭП чрезвычайно эффективно ввиду значительно более высокого КПД и существенно меньших массы и габаритных параметров. Однако уровень помех, создаваемых этими ИВЭП, требует специальных мер защиты. Инвертор ИВЭП, работающий в ключевом режиме, осуществляет преобразование выпрямленного постоянного напряжения в переменное с частотой прямоугольных импульсов от 20 до 100 кГц. Сила тока импульсов определяется мощностью, которая должна отдаваться ИВЭП в нагрузку. Естественно, что такие импульсы неизбежно создают мощные помехи, распространяющиеся по цепям первичного питания и в виде электромагнитного излучения через эфир.

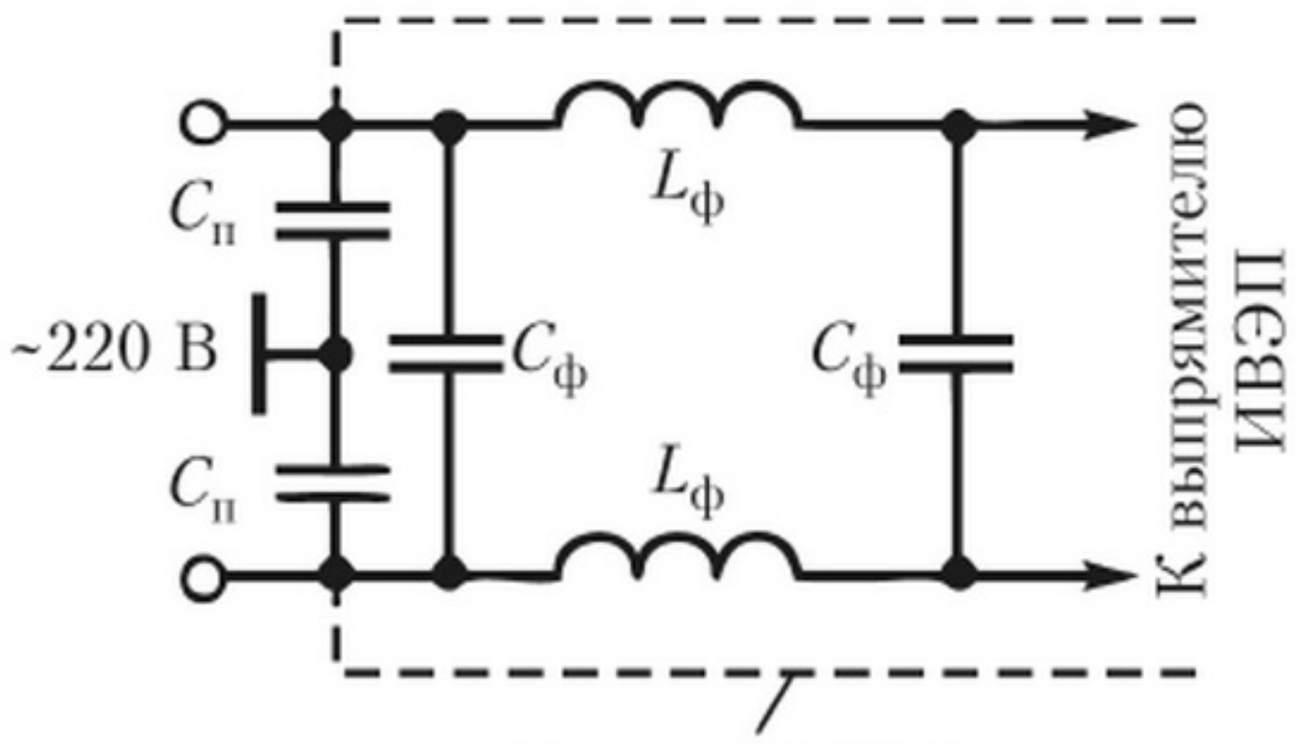


Рис. 2.38. **Фильтр, применяемый для защиты первичного питания от помех, создаваемых ИВЭП с бестрансформаторным входом**

Содержание

Для защиты первичного питания от помех, создаваемых инвертором, во входных цепях применяются специальные LC-фильтры (рис. 2.38), похожие на сглаживающие фильтры, рассмотренные в параграфе 2.8 (см. рис. 2.30, *в*).

Эти фильтры рассчитываются на частоты свыше десятков килогерц. Индуктивное сопротивление дросселя $L_ф\, x_L = \omega L$ на частоте первичной сети очень невелико и не влияет на передачу энергии из первичной сети ~220 В в ИВЭП. На частоте помехи это сопротивление становится весьма большим и препятствует распространению помех в первичную сеть. Конденсаторы, напротив, на частоте первичной сети имеют высокое емкостное сопротивление $x_C = 1/\omega C$, поэтому потерь передачи энергии из первичной сети в ИВЭП не вызывают. Для импульсов помехи их емкостное сопротивление становится низким, и через это сопротивление происходит шунтирование сигнала помехи. Кроме того, во входных цепях применяются специальные проходные конденсаторы $C_п$. Эти конденсаторы представляют собой проводник, окруженный шайбой. Между шайбой и проводником — слой диэлектрика, так что проводник и шайба образуют конденсатор. Шайба впаивается в корпус ИВЭП, который обычно заземляется. В результате между проводником и заземленным корпусом образуется емкостная связь. Через эту емкость на высокой частоте помехи происходит ее замыкание на землю.

Другой вид помех, связанный с работой ИВЭП, заключается в кратковременных скачках выходного напряжения при резких изменениях тока нагрузки. Скачкообразное увеличение или уменьшение потребляемого тока может привести к выбросам выходного напряжения ИВЭП, дестабилизирующим работу питаемых им устройств. В этих случаях применяют комбинированный метод стабилизации, заключающийся в применении наряду с ИВЭП с импульсным стабилизатором специального устройства подавления (УП).

В зависимости от выполняемых функций существуют УП провалов и УП выбросов. В УП провалов содержится дополнительный источник питания, который подключается только в переходных режимах при скачкообразном увеличении тока нагрузки. Сигнал управления при появлении таких скачков приводит в действие УП провалов и к нагрузке поступает дополнительный ток, поддерживающий выходное напряжение на допустимом уровне. По достижении выходным напряжением установившегося

значения ток, протекающий через УП провалов, прекращается.

УП выбросов предотвращает всплески выходного напряжения, которые могут возникать при скачкообразном уменьшении тока нагрузки. Принцип подавления выбросов заключается в том, что в течение переходного процесса сигнал управления включает УП выбросов, который шунтирует выход ИВЭП и избыток выходного тока ИВЭП замыкается через него.

2.10.2. Экранирование электронных устройств

Одним из эффективных способов борьбы с помехами, распространяющимися по эфиру, является экранирование. Электронное устройство помещают в глухой проводящий корпус, который служит экраном, препятствующим распространению электромагнитных помех. Обычно экранами окружают устройства — источники электромагнитных помех (например, инвертор ИВЭП с бестрансформаторным входом или вся конструкция такого ИВЭП), а также особо чувствительные к электромагнитным помехам электронные устройства, чаще всего измерительно-преобразовательные устройства, высокочувствительные усилители. В первом случае обеспечивается блокирование электромагнитных помех внутри самого источника, во втором — защита устройства от влияния внешних электромагнитных помех.

Реальные конструкции экранов всегда негерметичны, имеют достаточно широкие щели, малые и большие отверстия (окна для теплоотвода, радиаторы), стыки и швы. Поэтому при оценке проникновения электромагнитных помех через экранирующие металлоконструкции проводятся расчеты полей, проникающих через щели и отверстия.

Максимальное проникновение электромагнитной помехи наблюдается в тех случаях, когда вектор напряженности магнитного поля направлен по касательной к плоскости отверстия, а электрического поля — по перпендикуляру. При этом в непосредственной близости от отверстия помеха оказывает максимально вредное воздействие. С удалением от отверстия напряженность поля убывает обратно пропорционально кубу расстояния, а с удалением от щели — обратно пропорционально его квадрату. Поэтому при разработке конструкции электронного устройства его особенно чувствительные элементы стараются разместить вдали от щелей и отверстий.

Существенно снижают экранирующие свойства корпусов устройств такие пути проникновения помех, как вводы проводников и кабелей, которые обычно выполняют в виде разъемных соединений. При этом влияние помехи будет тем больше, чем больше контактное сопротивление разъемного соединения. Снижение влияния помехи обеспечивается путем уменьшения этого сопротивления, для чего контакты соединений покрывают золотом или его сплавами.

2.10.3. Использование гальванической развязки

Одним из основных источников помех в радиоэлектронной аппаратуре является так называемый общий провод, в качестве которого может использоваться корпус. Обычно он служит одним из полюсов электропитания, поэтому подключение к нему разных электронных устройств, в особенности силовых, приводит к тому, что при коммутациях сильноточных цепей на нем «наводятся» кратковременные импульсные помехи. В то же время для передачи информации чисто электрическим путем между устройствами — источником и приемником информации — должен быть один общий потенциал, в качестве которого и выступает общий провод. В результате помехи, возникающие при коммутациях сильноточных цепей и распространяющиеся через корпус, неизбежно будут вызывать сбои в работе электронных устройств.

Эффективным методом борьбы с такого рода помехами является гальваническая развязка источников помех и устройств, на работу которых помехи оказывают вредное влияние. Для этого осуществляют раздельное питание силовых и слаботочных устройств, устройств выработки управляющих сигналов и исполнительных. В идеальном случае каждое из устройств должно иметь собственный ИВЭП.

Помимо борьбы с помехами, гальваническая развязка обеспечивает решение еще одной проблемы — совместной работы устройств, находящихся под разными потенциалами, так как простое электрическое соединение цепей таких устройств приведет к их выгоранию.

Однако при этом возникает проблема передачи информационных сигналов между разными устройствами. Для ее разрешения применяют различные способы гальванической развязки информационных цепей.

Использование **оптронной развязки** позволяет эффективно решать проблему передачи цифровой информации. Как это было показано в параграфе 1.2, оптрон содержит пару «светодиод — фотодиод», которая обеспечивает передачу сигнала через оптическое излучение. Поскольку оптическое излучение электрически нейтрально, оно не подвержено влиянию электромагнитных помех. Входные и выходные цепи при этом электрически никак не связаны, т.е. гальванически развязаны (рис. 2.39).

Оптронную развязку используют главным образом при передаче цифровой информации. Передаче двух цифровых уровней (единицы и нуля) соответствует либо включенное, либо отключенное состояние светодиода. В затемненном состоянии ток через фотодиод практически отсутствует. При облучении фотодиода светодиодом происходит генерация фототока, который в отличие от обычного прямого тока диодов является обратным и направлен от катода к аноду. Ток фотодиода преобразуется в цифровые уровни обычно с помощью транзистора. При отсутствии фототока, а значит, и базового тока, транзистор закрыт. Появление фототока вызывает ток базы транзистора, достаточный для того, чтобы ввести транзистор в насыщение. Таким образом, транзистор работает в ключевом режиме (см. параграф 3.3) и обеспечивает формирование цифровых уровней.

Кроме пар «светодиод — фотодиод» в оптронах используются также пары «светодиод — фототранзистор» и «светодиод — фототиристор». Для оптронной развязки выпускают также готовые микросхемы, содержащие ту или иную оптронную пару, а также транзисторные каскады, обеспечи-

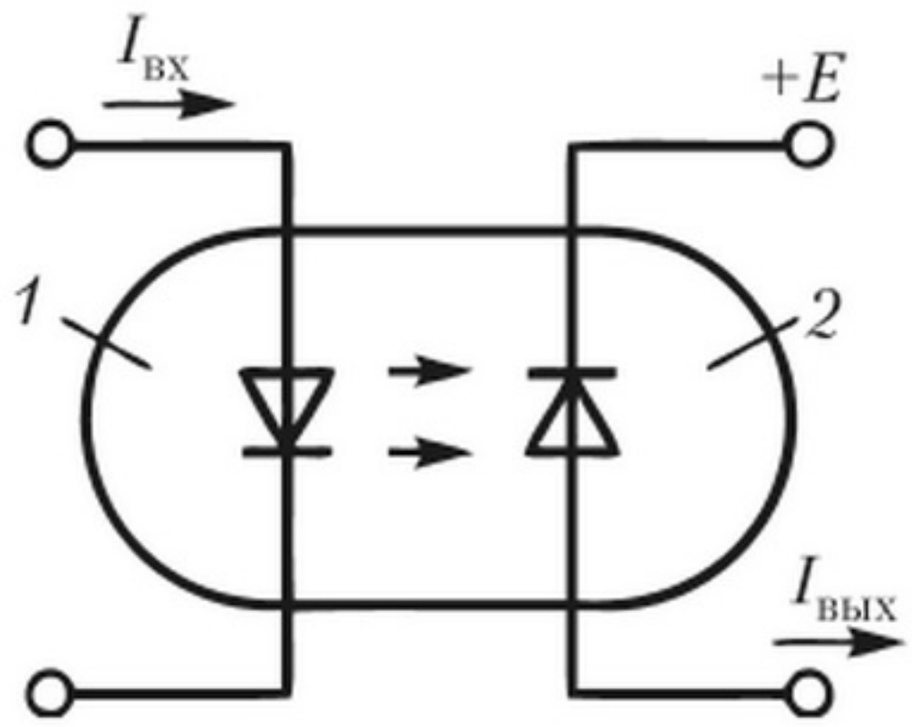


Рис. 2.39. **Схема гальванической развязки информационных цепей:**
1 — светодиод; *2* — фотодиод

вающие не только формирование двух цифровых уровней напряжения, но и необходимую нагрузочную способность.

На основе оптронной развязки выполняют передачу цифровой информации между гальванически развязанными электронными устройствами в сложных системах обработки информации. Очень важная функция оптронной развязки — обеспечение взаимосвязи между датчиками и исполнительными устройствами технологической системы и аппаратурой управления технологическими процессами, в основе которой, как правило, используется микропроцессорное вычислительное устройство (см. параграф 4.4). Через оптронную развязку осуществляется прием входной информации с датчиков релейного типа (имеющих два устойчивых состояния) и ввод ее в цифровое устройство управления — программируемый логический контроллер, а также выдача управляющих команд на исполнительные механизмы. При этом разность потенциалов, под которыми находятся контроллер и технологическая система, может достигать 500 В.

Релейная развязка широко применяется при коммутации сильноточных цепей управления исполнительными механизмами. Наиболее часто используется подача силового питания на исполнительное устройство (рис. 2.40).

Если допустимый ток, протекающий через замкнутые контакты реле, недостаточен для обеспечения тока нагрузки, применяют двухтактную релейную развязку, в которой контакты первого реле замыкают цепь обмотки второго сильноточного реле. Последнее может иметь пита-

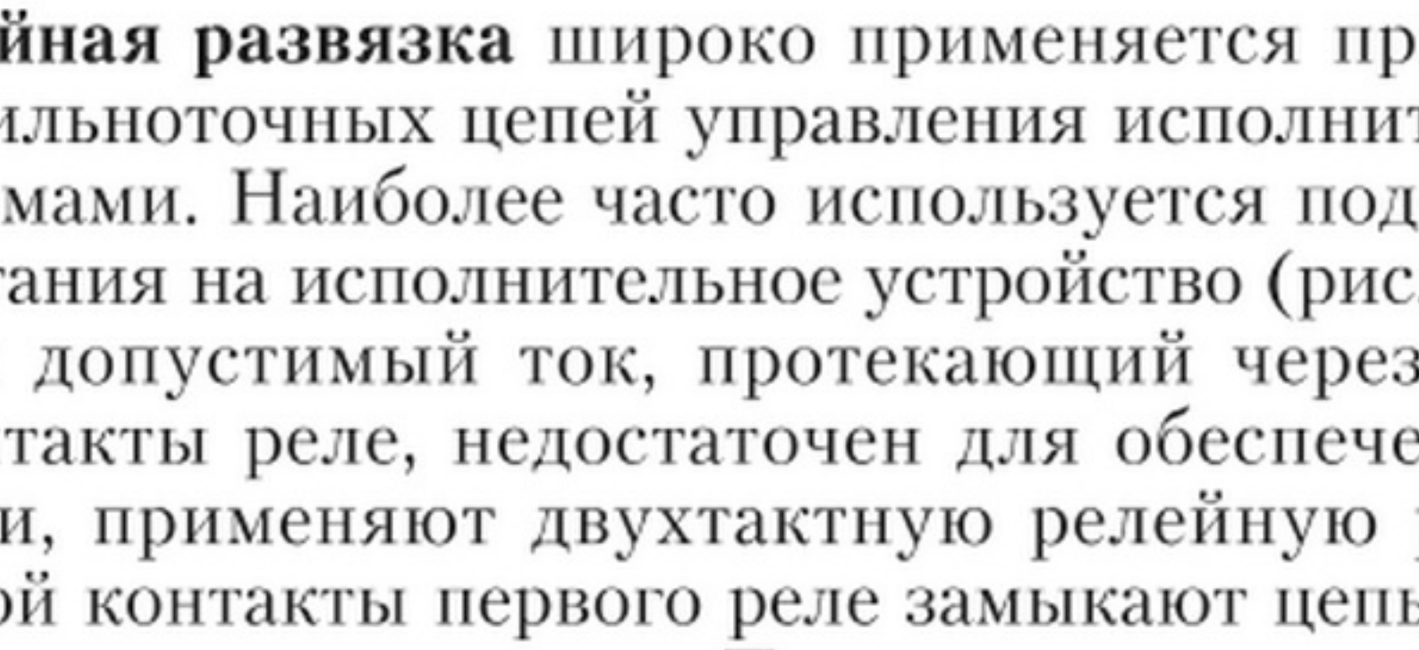


Рис. 2.40. **Схема релейной развязки**

ние от сети переменного тока 220 В и коммутировать трехфазное напряжение. Такие мощные электромагнитные реле, называемые пускателями, обеспечивают коммутацию электродвигателей и других сильноточных исполнительных устройств. Промышленные контроллеры, как правило, содержат набор специальных модулей, выполненных на основе электромагнитных реле и обеспечивающих гальваническую развязку контроллера и сильноточных управляющих цепей.

Трансформаторная развязка применяется при передаче аналоговых сигналов. Как известно, через трансформатор можно передавать импульсные сигналы, поэтому схема трансформаторной развязки (рис. 2.41) должна содержать устройства преобразования медленно изменяющихся сигналов в импульсный и обратно. Эту функцию выполняют модулятор, стоящий во входной цепи, и демодулятор, устанавливаемый в выходной цепи. Модулятор обеспечивает преобразование входного напряжения в последовательность импульсов, огибающая которых повторяет форму медленно изменяющегося входного напряжения $U_{вх}$.

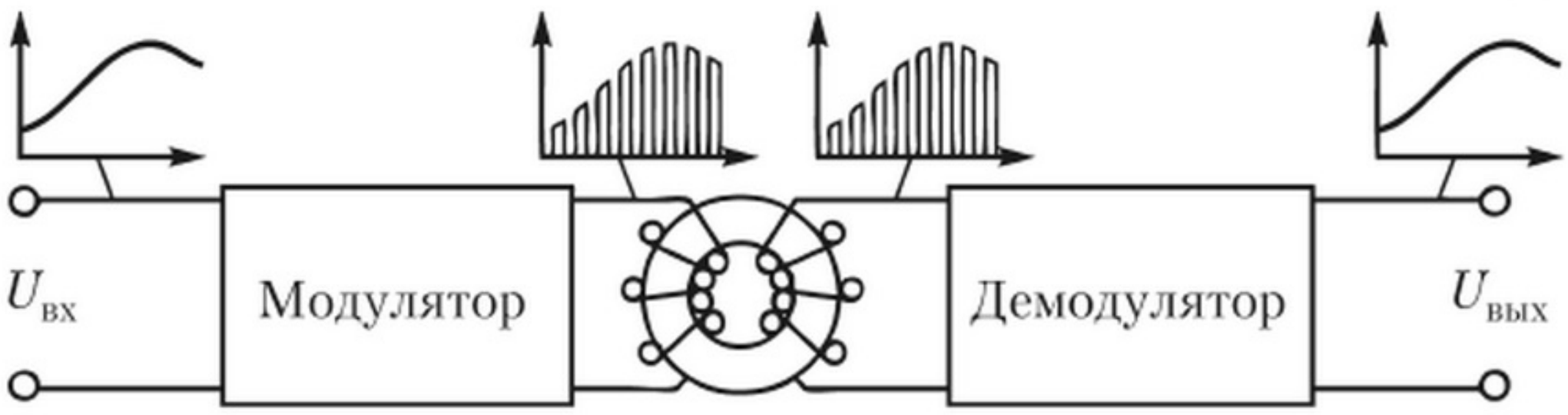


Рис. 2.41. **Схема трансформаторной гальванической развязки**

Импульсный сигнал передается через развязывающий импульсный трансформатор, после чего становится гальванически развязанным с цепью входного сигнала. Демодулятор обеспечивает обратное преобразование импульсного сигнала в медленно изменяющийся выходной сигнал $U_{вых}$.

2.10.4. Конструктивные методы борьбы с помехами

Влияние помех на работу электронных устройств можно снизить, если размещение элементов и устройств, их электрические связи выполнять с учетом возможного воздействия помех. Исходя из этого, при разработке конструкции электронной аппаратуры придерживаются следующих правил:

• чувствительные схемы располагают поблизости от источников полезных сигналов;

• мощные схемы, являющиеся источниками помех, располагают вблизи нагрузок;

• маломощные и мощные схемы располагают как можно дальше друг от друга;

• стремятся обеспечить как можно более короткие линии связи и контуры тока.

Устройства, входящие в аппаратуру, группируют в подсистемы в соответствии с тем, являются ли они источниками помех, или насколько велика их чувствительность к помехам. В соответствии с этим в сложной электронной системе выделяют несколько подсистем. В большинстве систем выделяют аналоговые подсистемы, как наиболее чувствительные к помехам; цифровые подсистемы, имеющие умеренную чувствительность к помехам; а также источники питания и подсистемы, содержащие электромагнитные приборы и переключатели. Высокочастотные и низкочастотные схемы также выделяют в разные подсистемы. Каждая подсистема должна быть как можно более компактной и снабжена собственными средствами подачи питания и заземления. Проводники в подсистемах должны быть как можно более короткими и иметь низкий импеданс, а контуры прохождения тока — минимальными.

При монтаже трансформаторов, соленоидов и других электромагнитных устройств обеспечивают взаимную перпендикулярность направлений их магнитных полей.

Группирование устройств, выполненных на цифровых ИМС, имеет некоторые особенности. В цифровых устройствах неизбежно возникают скачки тока в цепях питания, так как в состоянии логического нуля и логической единицы потребление логических элементов может резко отличаться. В связи с этим одновременное изменение состояния множества логических элементов приводит к появлению таких скачков в цепях питания.

При конструировании цифровых устройств, рассмотренных в гл. 3, 4, придерживаются следующих рекомендаций.

1. Цепи, потребляющие большой ток, питают от отдельного источника. В этом случае переменные составляющие тока питания не проникают в шины, подводящие питание к маломощным логическим схемам.

2. Особое внимание уделяют расположению и длине печатных проводников. Это связано с тем, что скачки тока в одном печатном проводнике через паразитные индуктивные и емкостные связи наводят всплески напряжения на других участках схемы. Поэтому при использовании многослойных печатных плат направления трассировки проводников в соседних слоях делают взаимно перпендикулярными.

3. Свободные, неиспользованные логические входы ИМС подключают к уровню либо логической единицы, либо логического нуля. В противном случае они могут вызвать ложные срабатывания вследствие всплесков напряжения в цепях питания.

4. В цепях питания цифровых ИМС обязательно устанавливаются фильтрующие конденсаторы. При этом для отдельных особо важных ИМС, например микропроцессорных БИС/СБИС, устанавливают индивидуальные фильтрующие конденсаторы, а для прочих — один конденсатор на несколько ИМС.

Еще одним важным конструктивным приемом повышения помехоустойчивости электронной аппаратуры является правильная разводка шины «земля» и шины «общий» (корпус). Проводник «общий» — это силовая цепь, соответствующая одному из полюсов источника питания. Через этот проводник на полюс источника питания замыкается весь ток нагрузки. Шина «земля» должна использоваться не для передачи мощности, а только для заземления. В электронной аппаратуре эти цепи должны иметь соединение только в одной точке системы. Соединение еще хотя бы в одной точке приводит к образованию замкнутого контура, являющегося источником помех для работы различных устройств и системы в целом.

Контрольные вопросы и задания

1. Почему для усилителя важно большое входное сопротивление и малое выходное?
2. В чем причина возникновения нелинейных искажений у усилителя?
3. Почему колебания выходного напряжения в усилительном каскаде с общим эмиттером сдвинуты по фазе относительно колебаний входного?
4. В чем заключается отрицательная обратная связь?

Содержание

5. Как дифференциальный каскад реагирует на синфазный и противофазный сигналы и почему?

6. В чем преимущества и недостатки режимов работы выходных каскадов *A*, *B* и *AB*?

7. Что такое собственный коэффициент усиления операционного усилителя?

8. Что произойдет, если на вход операционного усилителя не будет подаваться напряжение смещения нуля?

9. Чем определяется коэффициент усиления инвертирующей схемы включения операционного усилителя и почему он отрицательный?

10. Назовите основные допущения при анализе работы операционных усилителей.

11. Определите коэффициент усиления неинвертирующей схемы включения операционного усилителя.

12. Какие вычислительные схемы можно синтезировать на основе операционных усилителей?

13. Какие процессы задают частоту работы мультивибратора и наклон «пилы» в генераторе пилообразного напряжения?

14. Что обеспечивает матрица R—$2R$?

15. В чем сущность способа последовательного приближения в аналого-цифровых преобразователях?

16. За счет чего уменьшаются габариты источника вторичного электропитания с бестрансформаторным входом?

17. Чем мостовая схема выпрямителя лучше, чем схема с одним диодом?

18. Для чего используются трехфазные выпрямители?

19. Выстройте фильтры в порядке улучшения характеристик сглаживания пульсаций: *CRC*, *CLC*, *C*. За счет чего происходит улучшение?

20. Что является регулирующим элементом в компенсационном стабилизаторе?

21. Как происходит изменение степени открытия транзистора в линейном стабилизаторе?

22. В чем сущность широтно-импульсной модуляции?

23. За счет чего происходит изменение выходного напряжения в тиристорном выпрямителе?

24. Что такое инвертор в силовой электронике?

25. Какие фильтры и из каких элементов применяют для защиты от помех, создаваемых источниками вторичного электропитания с бестрансформаторным входом?

26. Для чего применяют экранирование?

27. Какие виды гальванической развязки применяются в электронных устройствах?

28. Назовите способы конструктивной защиты от помех.

Содержание

Глава 3
ОСНОВЫ СХЕМОТЕХНИКИ ЦИФРОВЫХ УСТРОЙСТВ

В результате изучения главы 3 студент должен:

знать

- арифметические и логические основы построения цифровой полупроводниковой электроники;
- основные виды комбинационных цифровых устройств;
- основные виды последовательностных цифровых устройств;

уметь

- разрабатывать принципиальные электрические схемы простейших цифровых устройств;
- выполнять анализ работы простейших цифровых устройств, синтезированных из элементарных логических элементов;

владеть

- методами синтеза цифровых электронных устройств с использованием булевой алгебры и карт Карно.

3.1. Двоичная система счисления

В отличие от аналоговых устройств, в которых информационный сигнал может принимать любые значения в некотором диапазоне, в цифровых устройствах носителем информации является цифровой двоичный код. Этот код состоит из множества двоичных разрядов, каждый из которых принимает только два фиксированных значения — «0» или «1». Двум двоичным цифрам соответствуют два состояния электронной схемы. Обычно это два фиксированных уровня напряжения, которые может принимать сигнал на выходе схемы. Поэтому все разнообразие математических операций, выполняемых цифровыми устройствами, основывается на двоичной системе счисления.

Напомним суть записи числа, означающего, например, количество «тысяча триста восемь» в десятичной системе счисления:

$$1308 \to 1 \cdot 10^3 + 3 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0;$$
$$1 \cdot 1000 + 3 \cdot 100 + 0 \cdot 10 + 8 \cdot 1;$$
$$1000 + 300 + 0 + 8 = 1308.$$

По сути, набор цифр «один—три—нуль—восемь» представляет собой код, т.е. перечень коэффициентов (множителей), стоящих перед числом десять (основанием системы счисления) в степени (справа налево): нуль, один, два, три и т.д. «Вес» каждого коэффициента (цифры) определяется разрядом — позицией, которую он занимает в коде (единицы, десятки, сотни, тысячи и т.д.). Поэтому такая система счисления называется *позиционной*.

В позиционной двоичной системе счисления основанием служит число два, а коэффициентами могут быть только две цифры — нуль (0) и единица (1), поэтому, например, количество «тринадцать» запишется кодом 1101:

$$1101 \to 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0;$$
$$1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1;$$
$$8 + 4 + 0 + 1 = 13.$$

Арифметические действия в двоичной системе выполняют аналогично десятичной.

Рассмотрим сущность выполнения действия сложения, например, чисел 45 и 87 в десятичной системе. Сначала складываем цифры младшего разряда — нулевого, соответствующего 10^0 — единицам: 5 + 7 = 12. В числе 12 уже две цифры, причем цифра младшего разряда суммы — 2, а цифра 1 принадлежит более старшему разряду — первому, соответствующего 10^1 — десяткам. Поэтому 2 записываем, 1 — запоминаем, т.е. в младшем разряде суммы, соответствующем единицам, будет 2, а 1, соответствующую более старшему разряду — десяткам, будем учитывать при сложении цифр первого разряда. Далее складываем цифры первого разряда 4 + 8 и прибавляем к ним 1, которую запомнили; получаем 13. Теперь цифра, соответствующая первому разряду суммы — 3, а 1 переносится в старший разряд, соответствующий второму разряду суммы — 10^2, т.е. сотням.

Аналогично выполняется сложение в двоичной системе, например, чисел «шесть» и «семь» (табл. 3.1). Числа шесть и семь в двоичной системе имеют вид:

$$6 \to 110\ (1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 4 + 2 + 0 = 6);$$
$$7 \to 111\ (1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4 + 2 + 1 = 7).$$

Сложение начинаем с нулевого разряда, соответствующего 2^0:

$0 + 1 = 1$ (младшая цифра суммы равна 1).

Переходим к сложению цифр первого разряда, соответствующего 2^1: $1 + 1 = 2$. Для записи числа 2 в двоичной системе требуется код: 10 ($1 \cdot 2^1 + 0 \cdot 2^0 = 2$), т.е. цифра первого разряда суммы равна 0, а 1 переносим в старший разряд (нуль записываем, один запоминаем). Таким образом, если сумма цифр больше единицы, происходит перенос в старший разряд.

Далее складываем 1 + 1 и прибавляем 1, которую запомнили, получаем 3. Число 3 в двоичной системе записываем кодом 11 ($1 \cdot 2^1 + 1 \cdot 2^0 = 3$), т.е. цифра второго разряда суммы равна 1, а старшая 1 переносится в следующий (третий) разряд.

Таким образом, сумма чисел 110 и 111 равна 1101. Нетрудно убедиться, что это двоичное число — 13 ($1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$).

Таблица 3.1

Сложение		Вычитание	
десятичная	двоичная	десятичная	двоичная
1↖ 4 5 + 8 7 1 3 2	1↖ 1 1 0 (6) + 1 1 1 (7) 1 1 0 1 (13)	• • 1 3 2 − 8 7 4 5	• • 1 1 0 1 (13) − 1 1 0 (6) 1 1 1 (7)
			1↖ 1 1 0 1 (13) + 0 0 1 (обратный код) 1 1 1 0 + ↘1 1 1 1 (7)

Примечание: 1↖ — перенос из младшего разряда в старший; ↘1 — перенос из старшего разряда в младший; • — заем из старшего разряда.

Рассмотрим теперь сущность действия вычитания в десятичной системе на примере: 132 — 87:

- сначала вычитаем цифры нулевого разряда, но так как из двух нельзя вычесть семь, занимаем 1 (один десяток — основание системы счисления) в старшем разряде, что отмечаем точкой над цифрой 3; получаем 10 + 2 = 12; 12 – 7 = 5;

• при вычитании цифр первого разряда необходимо занять 1 в следующем старшем разряде и, учитывая, что в первом разряде осталась цифра 2, получаем 10 + 2 — 8 = 4.

Аналогично выполняем вычитание из числа 1101 (13) числа 110 (6) в двоичной системе:

• нулевой (2^0) разряд: 1 – 0 = 1;

• первый (2^1) разряд: из нуля нельзя вычесть единицу, поэтому занимаем 1 (т.е. одно основание системы счисления, равное двум) и получаем 2 – 1 = 1;

• второй (2^2) разряд: из нуля, оставшегося после занимания единицы в этом разряде, нельзя вычесть единицу, поэтому занимаем 1 (основание, равное двум) в третьем (2^3) разряде: 2 – 1 = 1. Получим 111, т.е. двоичное число «семь».

Покажем, что операцию вычитания можно заменить сложением, но для этого уменьшаемое (т.е. 1101) надо складывать с обратным кодом вычитаемого, который получаем, заменив цифры на обратные (т.е. 1 на 0, а 0 на 1). Произведя сложение, переносим 1 из старшего разряда в младший, как показано стрелкой, и получаем двоичный код 111, соответствующий числу «семь».

Действия умножения и деления (а как доказано в математике, и любые другие операции) можно свести к сложению и вычитанию кодов, сдвинутых влево или вправо на то или иное число разрядов. Сдвиг числа влево на один разряд соответствует умножению его на 2, а вправо — делению на 2. Действительно, каждая цифра числа при его сдвиге влево будет иметь вес уже не i-го разряда, т.е. 2^i, а $(i + 1)$-го, т.е. 2^{i+1}. При сдвиге вправо вес каждой цифры будет уменьшаться в 2 раза, т.е. составит не 2^i, а 2^{i-1}.

Покажем, как с помощью операций сложения и сдвига можно выполнить умножение двоичных чисел. Вычисление произведения осуществляют от старших разрядов множителя к младшим по шагам. На каждом шаге анализируют очередную цифру множителя. Если она равна 1, то к промежуточному результату вычислений прибавляют множимое, если 0, то результат оставляют без изменений. При переходе к более младшему разряду промежуточный результат сдвигают влево на один разряд, т.е. умножают на 2. Последнему шагу вычислений соответствует нулевой разряд множителя. Обозначим через A множимое, через B — множитель, а через b_i — i-й разряд множителя. Тогда

в результате вычислений по описанному выше алгоритму будет получено число П:

$$\begin{aligned}\text{П} &= (\ldots((A \cdot b_n)\,2 + A \cdot b_{n-1})\,2 + \ldots + A \cdot b_1)\,2 + A \cdot b_0 = \\ &= A \cdot b_n \cdot 2^n + A \cdot b_{n-1} \cdot 2^{n-1} + \ldots + A \cdot b_1 \cdot 2 + A \cdot b_0 = \\ &= A\,(b_n \cdot 2^n + b_{n-1} \cdot 2^{n-1} + \ldots + b_1 \cdot 2 + b_0).\end{aligned}$$

Выражение в последних скобках есть не что иное, как число B, а число П — произведение A и B.

Поскольку вычитание сводится к сложению обратного кода, становится понятным, почему любые математические операции можно осуществить с помощью простейших операций сложения и сдвига, что и используется в процессорах ЭВМ.

Таким образом, если в десятичной системе для записи кодов используется десять цифр (от 0 до 9), то в двоичной системе — лишь две (0 и 1), которые называют битами (двоичными цифрами).

В любой системе кодирования должно выполняться неравенство $N < S^n$, где N — количество кодируемых объектов; S — основание кода (основание системы счисления); n — длина кода (количество разрядов в коде-числе).

Например, для кодирования количества «тринадцать» в десятичной системе счисления достаточно двухразрядного кода $13 < 10^2$, а в двоичной необходим четырехразрядный код $13 < 2^4 = 16$, как показано в табл. 3.1.

Для хранения и обработки информации в виде текстов, формул и чисел необходимо с помощью бит закодировать около 150 различных символов (заглавные и строчные буквы латинского и русского алфавитов, знаки препинания, математические знаки, десять цифр и т.п.), т.е. $N \approx 150$. Для этого необходимы восьмиразрядные коды ($150 < 2^8 = 256$). Восьмиразрядный код называют байтом. Емкость памяти ЭВМ оценивают в килобайтах (Кбайт), мегабайтах (Мбайт) и гигабайтах (Гбайт).

В табл. 3.2 приведены примеры двоичного кодирования знаков, латинских букв и десятичных цифр.

Таблица 3.2

Символ	Код	Символ	Код	Символ	Код
A	11000001	0	11110000	7	11110111
a	10000001	1	11110001	8	11111000

Окончание табл. 3.2

Символ	Код	Символ	Код	Символ	Код
B	11000010	2	11110010	9	11111001
в	10000010	3	11110011	Точка	01001011
C	11000011	4	11110100	Плюс	01001110
c	10000011	5	11110101	Минус	01101101
D	11000100	6	11110110		

В кодах табл. 3.2, например, величина –6.285 будет закодирована так:

01101101 11110110 01001011 11110010 11111000 11110101.
(минус) (шесть) (точка) (два) (восемь) (пять)

3.2. Алгебра логики

На первый взгляд цифровые устройства кажутся относительно сложными. Однако они основаны на принципе многократного повторения относительно простых базовых логических схем. Связи между этими схемами строятся на основе чисто формальных методов. Инструментом такого построения служит булева алгебра, названная по имени одного из ее разработчиков — английского математика Джорджа Буля. Применительно к цифровой технике она называется также алгеброй логики. В отличие от переменной в обычной алгебре логическая переменная имеет только два значения, которые обычно называются логическим нулем и логической единицей. В качестве обозначений используется «0» и «1» или просто 0 и 1. В дальнейшем по тексту учебника, чтобы отличить их от обычных 0 и 1, будем использовать обозначения лог. 0 и лог. 1, «0» и «1», а в данном параграфе — просто 0 и 1.

Существуют три основные операции между логическими переменными: *конъюнкция* (логическое И), *дизъюнкция* (логическое ИЛИ) и *инверсия* (логическое НЕ). В алгебре логики используются следующие обозначения операций:

- конъюнкция: $F = A \wedge B = A \cdot B = AB$;
- дизъюнкция: $F = A \vee B = A + B$;
- инверсия: $F = \overline{A}$.

Таблица 3.3

Таблица истинности для логических функций

И: $F = A \wedge B$			ИЛИ: $F = A \vee B$			НЕ: $F = \overline{A}$	
A	*B*	*F*	*A*	*B*	*F*	A	*F*
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Как следует из табл. 3.3, для конъюнкции — логического И — *F* только тогда равна 1, когда ее аргументы *A и B* равны 1. При дизъюнкции (логическом ИЛИ) *F* равна 1 тогда, когда *A или B* равны 1. Отсюда и следуют названия этих функций. Обе эти функции можно распространить на сколь угодно большое число переменных. Инверсия — логическая функция только одной переменной.

3.2.1. Основные теоремы и положения алгебры логики

Принцип двойственности. Запишем правила выполнения операций ИЛИ и И, расположив строчки И в обратном (снизу вверх) порядке:

ИЛИ	И
$0 \vee 0 = 0$	$1 \wedge 1 = 1$
$0 \vee 1 = 1$	$1 \wedge 0 = 0$
$1 \vee 0 = 1$	$0 \wedge 1 = 0$
$1 \vee 1 = 1$	$0 \wedge 0 = 0$

Сравним построчно операции ИЛИ и И. Нетрудно видеть, что если заменить в строках ИЛИ и И все 0 на 1, все 1 на 0 и знаки дизъюнкции на знаки конъюнкции, то правила меняются местами: строка ИЛИ превращается в строку И и наоборот.

В этом состоит принцип двойственности, который в общем виде записывается так:

$$\overline{A \vee B} = \overline{A} \cdot \overline{B}.$$

Для преобразования формул алгебры логики с целью их минимизации используются, как и в обычной алгебре, скобки, а если их нет, то сначала выполняется отрицание (инверсия) над отдельными переменными, затем логическое умножение (конъюнкция) и наконец логическое сло-

жение (дизъюнкция). Однако если черта (знак инверсии) стоит над совокупностью букв и знаков, то она выполняется в последнюю очередь.

В процессе преобразования формул используются также теоремы алгебры логики.

Теоремы для одной переменной (легко проверяемые подстановкой $A = 1$ и $A = 0$):

1. $A \vee 0 = A$;
2. $A \vee 1 = 1$;
3. $A \vee A = A$;
4. $A \vee \overline{A} = 1$;
5. $A \cdot 0 = 0$;
6. $A \cdot 1 = A$;
7. $A \cdot A = A$;
8. $A \cdot \overline{A} = 0$;
9. $\overline{\overline{A}} = A$.

Заметим, что эти теоремы (как и последующие) остаются справедливыми и для случая, если под A понимать не только одну переменную, но и целое выражение.

Теоремы для двух и более переменных:

10а. $A \vee B = B \vee A$; 10б. $AB = BA$ — *переместительный закон*;

11а. $A \vee B \vee C = A \vee (B \vee C) = (A \vee B) \vee C$;

11б. $ABC = A\,(BC) = (AB)\,C$ — *сочетательный закон*;

12а. $A\,(B \vee C) = AB \vee AC$; 12б. $A \vee BC = (A \vee B) \cdot (A \vee C)$ — *распределительный закон*.

Если теоремы 10, 11 очевидны и совпадают с правилами обычной алгебры, то очевидность теоремы 12б (как и ряда следующих теорем) следует из принципа двойственности.

В самом деле, заменим в левой и правой частях теоремы 12а все переменные их отрицаниями и поменяем знаки конъюнкции и дизъюнкции друг на друга, получим: $\overline{A} \vee \overline{B}\,\overline{C} = (\overline{A} \vee \overline{B}) \cdot (\overline{A} \vee \overline{C})$. Введем новые обозначения: $\overline{A} = D$; $\overline{B} = E$ и $\overline{C} = G$. Получим $D \vee EG = (D \vee E) \cdot (D \vee G)$, а это и есть теорема 12б;

13а. $A \vee AB = A$; 13б. $A\,(A \vee B) = A$ — *закон поглощения* (читается «A поглощает B»).

Доказательство теоремы 13а: $A \vee AB = A\,(1 \vee B) = A \cdot 1 = A$ (используя теоремы 2 и 6); доказательство теоремы 13б следует из принципа двойственности;

14а. $(A \vee \overline{B})\,B = AB$; 14.б $AB \vee B = A \vee B$.

Доказательство теоремы 14а:

$(A \vee \overline{B})B = AB\ \overline{B}B\ = AB\ 0 \vee AB$ (используя теоремы 8 и 1), доказательство теоремы 14б следует из принципа двойственности;

15а. $AB \vee \overline{A}B = B$; 15б. $(A \vee B)\cdot(\overline{A} \vee B) = B$ – *закон склеивания* (читается «склеивание по A»).

Доказательство теоремы 15а: $AB \vee \overline{A}B = B(A \vee \overline{A}) = B \cdot 1 = B$ (используя теоремы 4 и 6); доказательство теоремы 15б следует из принципа двойственности.

16а. $\overline{A \vee B \vee C \vee ...} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot ...$; 16б. $\overline{A \cdot B \cdot C ...} = \overline{A} \vee \overline{B} \vee \overline{C} \vee ...$ — *теорема де Моргана*, представляющая наиболее общую формулировку принципа двойственности.

3.2.2. Булевы функции

Значение булевой функции F, как результат выполнения логических операций над двоичными переменными — аргументами A, B, C, ..., зависит от значения аргументов. *Задать булеву функцию* — значит указать значения, которые принимает функция (т.е. 0 или 1) при всех возможных комбинациях значений аргументов. Таблица, в которой построчно указываются все возможные сочетания аргументов и значения, которые булева функция принимает при каждом сочетании, называется *таблицей истинности*. Так, например, функции ИЛИ и И имеют только два аргумента A и B, поэтому имеется четыре возможные комбинации их значений, которые представлены в их таблице истинности (см. табл. 3.3).

Каждую конкретную комбинацию аргументов называют *набором*. Для краткости набор записывают в виде двоичного числа, цифрами которого являются значения аргументов. Для n аргументов существует $N = 2^n$ наборов.

Если неизвестно, какие значения принимает функция на всех наборах аргументов, то она называется *недоопределенной*, или не полностью (частично) определенной, а комбинации аргументов, для которых функция не определена, — *запрещенными наборами*.

Значения функции на запрещенных наборах можно задать по своему усмотрению (*доопределить* функцию). Этот прием используется при минимизации функций.

После того как таблица истинности составлена, находят алгебраическую форму этой логической функции. На следующем этапе функцию преобразуют в простейшую форму, которую потом реализуют с помощью соответствующих электрических схем. Логические функции записывают, как правило, в дизъюнктивной нормальной форме. При этом поступают следующим образом.

1. В таблице истинности выделяют строки, в которых булева функция имеет значение 1.

2. Для каждой такой строки составляют конъюнкцию всех входных переменных, причем записывают сомножитель без инверсии, если переменная принимает значение 1 (например, A), в противном случае записывают сомножитель с инверсией (например, $\overline{A}$). Таким образом составляют столько конъюнкций всех аргументов, сколько имеется строк в таблице истинности, в которых функция равна 1. Каждая такая конъюнкция называется конституентой единицы.

Конституента единицы — это функция n аргументов, которая принимает значение, равное единице, только на одном наборе аргументов. На всех остальных наборах она равна нулю.

3. Наконец, записывая логическую сумму — дизъюнкцию всех найденных конституент единицы, получают искомую функцию.

Рассмотрим этот способ на примере таблицы истинности для функций F и F' (табл. 3.4).

Таблица 3.4

Набор	A	B	C	F'	F
0	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	1	1
3	0	1	1	0	0
4	1	0	0	1	1
5	1	0	1	0	0
6	1	1	0	1	1
7	1	1	1	–	0

Функция F' является недоопределенной. В строке 7 (правильнее сказать — на седьмом наборе) значение функции неопределенное. Доопределим эту функцию таким образом, чтобы она на этом наборе была равна 0 (функция F).

На наборах 2, 4, 6 функция равна единице. Прежде всего следует составить конъюнкции (конституенты единицы) для этих наборов.

Набор 2: $K_2 = \overline{A}B\overline{C}$.

Набор 4: $K_4 = A\overline{B}\overline{C}$.

Набор 6: $K_6 = AB\overline{C}$.

Искомая функция записывается в виде логической суммы — дизъюнкции полученных конституент единицы:

$$F = K_2 \vee K_4 \vee K_6;$$

$$F = \overline{A}B\overline{C} \vee A\overline{B}\overline{C} \vee AB\overline{C}.$$

Эта запись называется совершенной дизъюнктивной нормальной формой (СовДНФ) рассматриваемой логической функции.

3.2.3. Минимизация булевых функций

После получения алгебраической формы записи логической функции переходят к ее упрощению или, как говорят в булевой алгебре, к минимизации.

Минимизация булевых функций производится с помощью теорем алгебры логики. Наиболее эффективными приемами минимизации являются вынесение за скобки общих членов, применение двойного отрицания, теоремы де Моргана, законов поглощения и склеивания. При этом наряду с полным склеиванием часто применяют неполное склеивание, при котором оба члена, участвовавшие в склеивании, или один из них остаются и могут склеиваться с другими конституентами. Неполное склеивание можно рассматривать как сочетание теоремы 15 с теоремой 3.

Рассмотрим процесс минимизации на примере функции F. Сначала определим, какие конституенты имеют общие члены. Это конституенты K_2 и K_6, а также K_4 и K_6. Так как конституенты K_2 и K_6 имеют общие члены $B\overline{C}$, они могут быть вынесены за скобки:

$$K_2 \vee K_6 = \overline{A}B\overline{C} \quad AB\overline{C} \quad B\overline{C}(\overline{A} \quad A) \vee$$

Оставшееся в скобках выражение $(\overline{A} \vee A) = 1$. Таким образом, конституенты K_2 и K_6 склеиваются по A:

$$K_2 \vee K_6 = B\overline{C}.$$

Однако конституента K_6 может понадобиться для операции склеивания конституент K_4 и K_6, поэтому проведем неполное склеивание с сохранением конституенты K_6, т.е.:

$$K_2 \vee K_6 = B\overline{C} \vee K_6.$$

Таким образом,

$$F = K_2 \vee K_4 \vee K_6 = (K_2 \vee K_6) \vee K_4 = B\overline{C} \vee K_6 \vee K_4.$$

Теперь можно провести склеивание по B конституент K_4 и K_6:

$$K_4 \vee K_6 = A\overline{B}\overline{C} \quad A\overline{B}\overline{C} \quad A\overline{C}(\overline{B} \quad B) \vee A\overline{C} =$$

В результате имеем более простое выражение для F:

$$F = B\overline{C} \quad A\overline{C}.$$

Вынесем за скобки $\overline{C}$ и получим конечный результат минимизации:

$$F = \overline{C}(B \quad A).\vee$$

3.2.4. Минимизация булевых функций с помощью карт Карно

Важнейшим вспомогательным средством для определения наиболее простой логической функции является карта Карно. Это не что иное, как измененная запись таблицы истинности. В этом случае значения аргументов не просто записываются рядом друг с другом, а размещаются по горизонтали и вертикали таблицы, деля ее, наподобие шахматной доски, на отдельные квадраты. При четном количестве аргументов половину из них записывают по горизонтали, а половину — по вертикали. При нечетном числе аргументов по горизонтали размещается на один аргумент больше, чем по вертикали.

Порядок размещения различных комбинаций значений аргументов следует выбирать таким, чтобы при переходе от одной ячейки к соседней изменялся лишь один аргумент. В эти ячейки заносят те значения логической функции, которые соответствуют значениям аргументов. В качестве примера (рис. 3.1) приведена таблица истинности и соответствующая ей карта Карно для функции И (конъюнкции).

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

а

B \ A	0	1
0	0	0
1	0	1

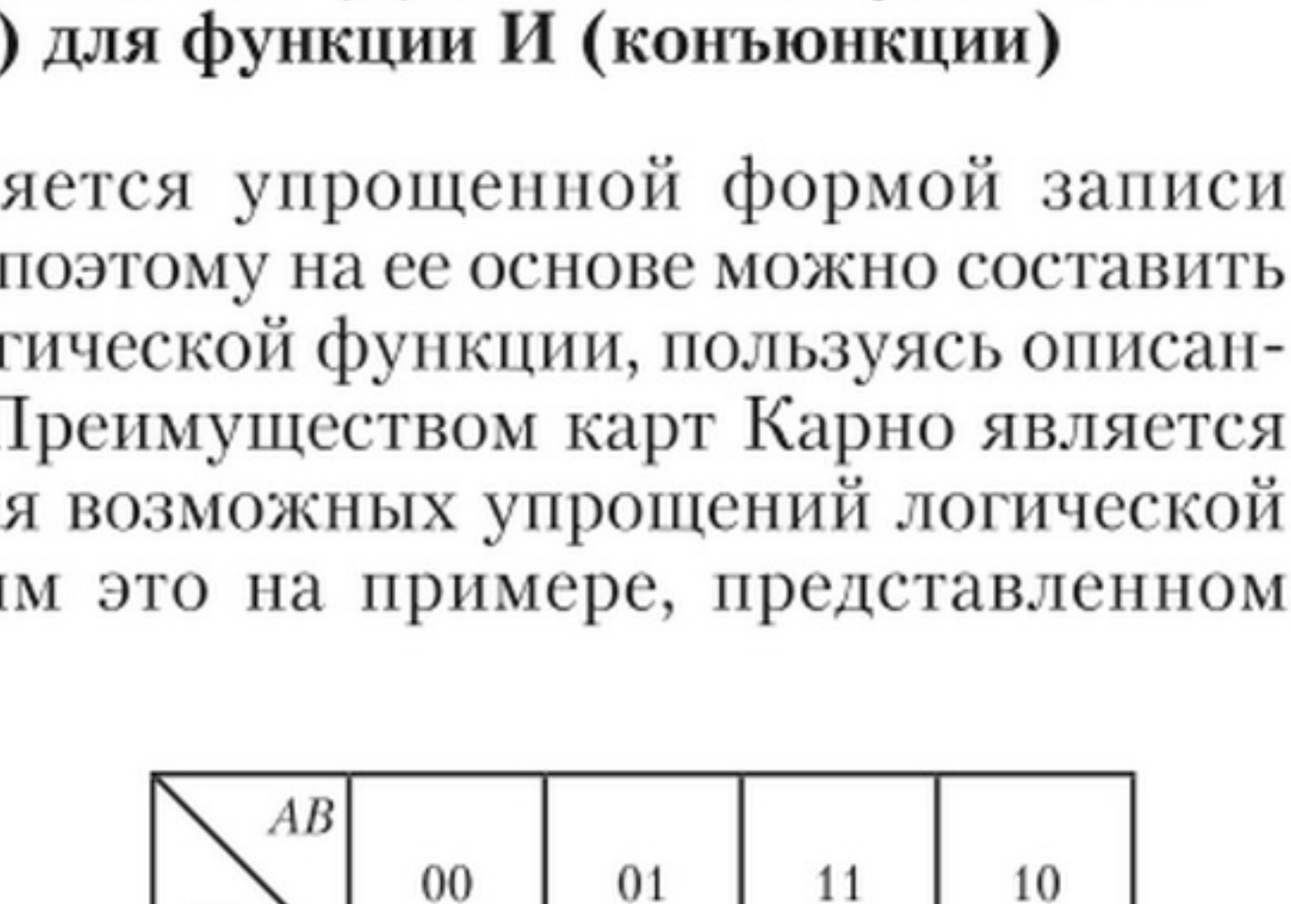
б

Рис. 3.1. **Таблица истинности (*а*) и соответствующая ей карта Карно (*б*) для функции И (конъюнкции)**

Карта Карно является упрощенной формой записи таблицы истинности, поэтому на ее основе можно составить СовДНФ искомой логической функции, пользуясь описанным выше методом. Преимуществом карт Карно является простота обнаружения возможных упрощений логической функции. Рассмотрим это на примере, представленном на рис. 3.2.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

а



б

Рис. 3.2. **Таблица истинности (*а*) и соответствующая ей карта Карно (*б*) для функции *F***

В первую очередь при составлении СовДНФ следует, как указывалось выше, составить конъюнкции всех аргументов для каждой конституенты единицы, т.е. для каждой ячейки, в которой стоит единица. Для ячейки, расположенной в левом верхнем углу, получаем:

$$K' = A\,B\,C\,D;$$

для ячейки, расположенной правее:

$$K'' = \overline{A}\,B\,\overline{C}\,\overline{D}.$$

Когда будет составлена дизъюнкция всех конституент единицы, помимо других в ней встретится и такой фрагмент:

$$K' \vee K'' = \overline{A}\,B\,C\,D \quad \overline{A}\,B\,\overline{C}\,\overline{D}.$$

Он упрощается следующим образом:

$$K' \vee K'' = \overline{A}\,C\,D(\overline{B} \quad B) \quad \vee \overline{A}\,C\,\overline{D}.$$

Отсюда следует общее правило упрощения логических функций для карт Карно: если в двух, четырех, восьми и т.д. ячейках, ограниченных прямоугольным или квадратным контуром, стоят только единицы, можно записывать непосредственно конъюнкцию для всей этой группы, причем в нее должны входить лишь те аргументы, которые остаются неизменными для всех ячеек данной группы.

Таким образом, в этом примере конъюнкция для группы *II*, состоящей из двух ячеек, равна

$$K_{II} = \overline{A}\,C\,\overline{D},$$

что соответствует ранее полученной функции. В одну группу связываются также те ячейки, которые находятся на левом и правом краях одной строки или в верхней и нижней частях одного столбца.

Для группы *I*, состоящей из четырех ячеек, можно записать:

$$K_I = \overline{A}\,\overline{B}.$$

Для группы *III*, имеющей квадратную форму и состоящей также из четырех ячеек, получим следующую конъюнкцию:

$$K_{III} = AC.$$

Еще одна единица осталась в правом верхнем углу. Она может быть связана, например, с единицей в нижней части рассматриваемого столбца в группу из двух ячеек. Другая возможность состоит в объединении единиц, находящихся на левом и правом краях первой строки. Однако если принять во внимание, что в каждом углу карты Карно находится единица, то можно найти простейшее решение. Связывая эти единицы в одну четырехэлементную группу, получим:

$$K_{IV} = \overline{B}\,\overline{D}.$$

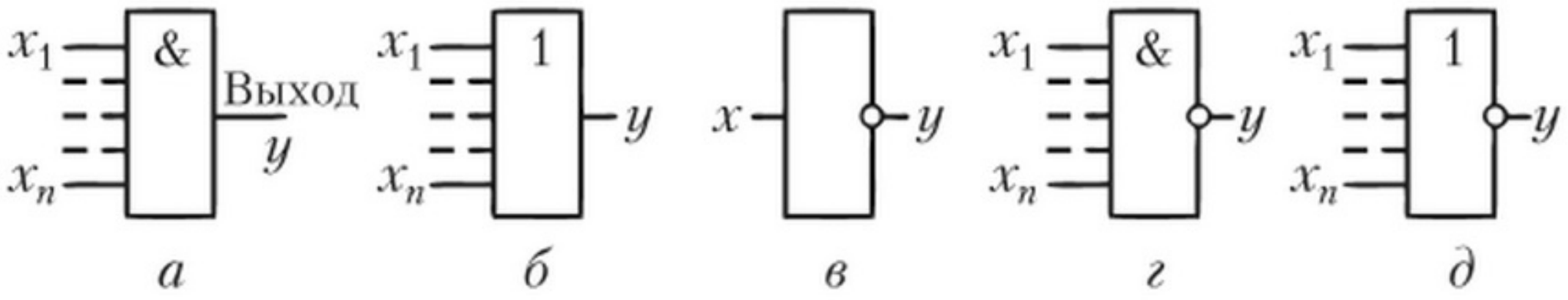
Таким образом, с помощью карты Карно можно найти наипростейший вариант логической функции F:

$$F = K_I \vee K_{II} \vee K_{III} \vee K_{IV} = AB \vee ACD \vee AC \vee BD.$$

3.2.5. Алгебра логики и цифровые электронные схемы

Возникает вопрос: как можно представить логические функции с помощью электрических схем? Так как логические переменные могут иметь только два дискретных значения, следует обратить внимание на схемы, которые могут находиться в двух легко различимых состояниях. Такими схемами являются электрические переключающие схемы, выполняемые на основе транзисторных ключей. Для представления логических переменных в цифровой схемотехнике используют электрическое напряжение, имеющее два различных уровня: высокий, близкий по уровню к напряжению питания (транзистор закрыт), и низкий, близкий к потенциалу корпуса (транзистор открыт). Этим уровням можно поставить в соответствие состояния логических «1» и «0». Если высокий уровень напряжения соответствует логической «1», а низкий — «0», такая система обозначений называется *позитивной логикой*. В противном случае (высокий — «0», низкий — «1») система называется *негативной логикой*.

В соответствии с тремя операциями алгебры логики в схеме цифровых устройств используют следующие логические элементы, входные переменные которых часто обозначают через x_i, а выходные через y:



а *б* *в* *г* *д*

Рис. 3.3. **Условные обозначения элементов цифровой логики:**

а – И; *б* – ИЛИ; *в* – НЕ; *г* – И-НЕ; *д* – ИЛИ-НЕ

1) элемент И — схема логического умножения, конъюнктор (рис. 3.3, *а*);

2) элемент ИЛИ — схема логического сложения, дизъюнктор (рис. 3.3, *б*);

3) элемент НЕ — схема логического отрицания, инвертор (рис. 3.3, *в*).

Этот набор элементов называют основным базисом или основной функционально полной системой элементов. Последнее означает, что с помощью этих элементов можно создать схему, осуществляющую любую сколь угодно сложную логическую операцию.

Помимо этих элементов в интегральной схемотехнике часто применяются логические схемы, выполняющие операции И-НЕ (рис. 3.3, *г*) и ИЛИ-НЕ (рис. 3.3, *д*).

Информация, поступающая в цифровое устройство, представляет дискретный (т.е. состоящий из нулей и единиц) сигнал (код). На передачу сигнала отводится конечный отрезок времени, называемый тактом работы устройства. Если за один такт в устройство передается один из разрядов двоичного числа, то устройство работает с последовательным кодом, если же за один такт передается все двоичное число одновременно, то устройство работает с параллельным кодом.

В общем случае на вход цифрового устройства поступает множество двоичных переменных X (x_1, x_2, ..., x_n), а с выхода снимается множество двоичных переменных Y (y_1, y_2, ..., y_s). При этом устройство осуществляет (реализует) определенную связь (логическую функцию) между входными и выходными переменными.

Цифровые устройства делят на *комбинационные* и *последовательностные*.

В комбинационных устройствах (рис. 3.4, *а*) значения Y в течение каждого такта определяются значениями X только

в этот же такт. Такие устройства состоят из логических элементов. В последовательностных устройствах (рис. 3.4, *б*) значения Y определяются значениями X как в течение рассматриваемого такта, так и существовавшими в ряде предыдущих тактов. Для этого в последовательностных устройствах кроме логических должны быть еще и запоминающие элементы. При этом память устройства может охватывать не бесконечно большое, а конечное число тактов. Поэтому цифровые (дискретные) устройства с памятью называют *конечными автоматами*, которыми являются все ЭВМ.

Подобно входным и выходным переменным, переменные, сохраняемые в памяти устройства, тоже двоичные и зависят от значений входных переменных в предыдущих тактах.

Любое дискретное устройство и составляющие его элементы и узлы осуществляют ту или иную булеву функцию над двоичными входными переменными.

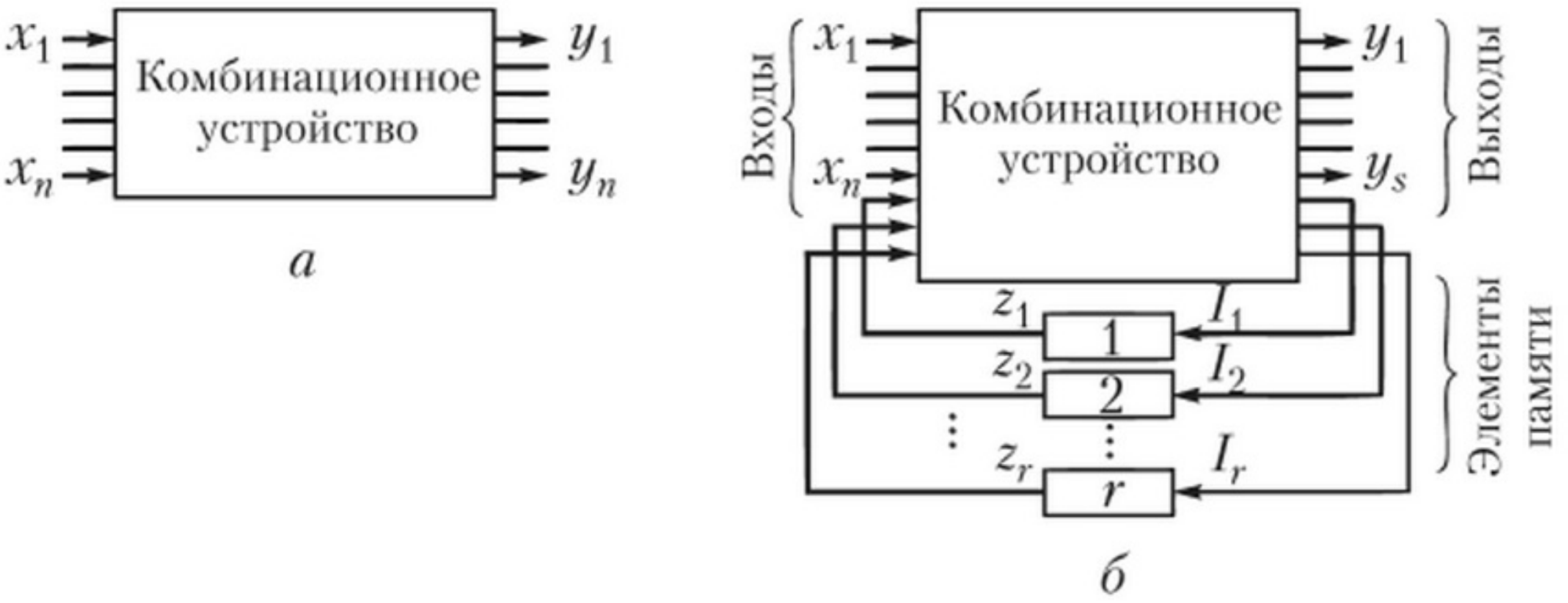
Известно, что булеву функцию можно задать тремя способами: *содержательно* (путем словесного описания), *таблично* и *алгебраически*. Наиболее часто для описания работы дискретных устройств пользуются табличной формой.

Таблицы, показывающие связь между входными и выходными переменными комбинационных устройств, так же как и в алгебре логики, называют *таблицами истинности*, а алгебраическая форма этих связей представляет систему алгебраических функций:

$$y_1 = y_1(x_1, x_2, ..., x_n),$$
$$.....$$
$$y_s = y_s(x_1, x_2, ..., x_n).$$

В последовательностных устройствах выходные переменные y_i зависят не только от входных сигналов x_k, но и от сигналов элементов памяти, поступающих в этот же такт. При анализе и синтезе последовательностные устройства делят на комбинационную часть и элементы памяти.

Обозначим два следующих друг за другом такта работы автомата как t и $(t + 1)$. Состояние элементов памяти в $(t + 1)$-й такт определяется множествами как входных сигналов, так и сигналов на выходах элементов памяти в предыдущий такт t, т.е. $z_i^{t+1} = \varphi_i(x_1, x_2, ..., x_n, I_1, I_2, ..., I_r)^t$. Это выражение называют *функцией переходов*.



а *б*

Рис. 3.4. **Структура комбинационного (*а*) и последовательностного (*б*) цифровых устройств**

Функции переходов и выходов последовательностных устройств могут выражаться *таблицами переходов и выходов*. Поскольку эти таблицы описывают работу последовательностного устройства, т.е. процесс его переключения, они называются также *таблицами переключений*.

Реальные электрические схемы всегда инерционны, и между моментом изменения сигналов на входе схемы и моментом появления соответствующего сигнала на выходе всегда проходит пусть и небольшое (наносекунды), но конечное время. Таблицы и алгебраические функции описывают лишь установившийся режим, т.е. в статике. В динамической же части такта связь между переменными может отличаться от режима статики. Это явление называют переходным состоянием (гонками в автоматах). Если его не учитывать, то спроектированное устройство может работать с ошибками. Для борьбы с гонками в автоматах используют синхронизацию работы электронных схем. В соответствии с этим цифровые устройства разделяются на асинхронные и синхронные. В асинхронных изменение входных сигналов сразу влечет за собой соответствующее изменение выходных сигналов (конечно, после окончания переходных процессов в электронных схемах). В синхронных изменение выходных сигналов происходит только после подачи синхронизирующих (тактовых) импульсов, управляющих работой автомата.

Комбинационные части автоматов являются асинхронными. Значит, на их выходе могут появляться гонки, которые приводят к сбоям (ошибкам), если элементы памяти автомата будут управляться непосредственно выходными

сигналами комбинационной части. Такие автоматы называют асинхронными, и в них существует опасность сбоев.

В синхронных автоматах элементы памяти изменяют свое состояние только с приходом внешнего тактового импульса, т.е. когда все переходные процессы в комбинационной части закончатся и на ее выходах установятся сигналы, соответствующие входным. Поэтому опасности сбоев из-за гонок в таких автоматах нет.

3.3. Ключевые схемы

Транзисторные ключевые схемы являются основой всей полупроводниковой цифровой схемотехники. Ключевая схема предназначена для коммутации (переключения) тока в нагрузке и содержит источник напряжения питания, нагрузку (резистор R) и ключ ($Кл$). Полупроводниковый ключ подобен механическому выключателю (рис. 3.5, *а*). Если ключ идеален, т.е. его сопротивление в разомкнутом состоянии бесконечно велико, а в замкнутом равно нулю, то ток в цепи при разомкнутом ключе $I = 0$, а при замкнутом $I = E/R$. Пусть за выходное напряжение $U_{вых}$ принято напряжение между корпусом и точкой $К$. Тогда при разомкнутом ключе, когда падение напряжения на сопротивлении R отсутствует, потенциал точки $К$ будет равен потенциалу источника E и, следовательно, $U_{вых} = E$.

Когда же ключ замкнут, то падение напряжения на сопротивлении от протекающего по нему тока полностью уравновешивает напряжение источника $IR = E$, потенциал точки $К$ равен потенциалу корпуса, принятому за нулевой уровень, и, следовательно, $U_{вых} = 0$.

Построив в координатах $I—U$ по точкам E и E/R нагрузочную прямую (рис. 3.5, *б*) и рассматривая ось абсцисс как вольт-амперную характеристику ключа в разомкнутом состоянии, а ось ординат — в замкнутом, придем к выводу, что точка A определяет состояние схемы при замкнутом, а точка B — при разомкнутом ключе. Для идеального ключа коэффициент использования напряжения питания $K_и = (U_B - U_A)/E = 1$.

Если же ключ неидеальный, то в замкнутом состоянии он обладает хоть небольшим, но внутренним сопротивлением $R_{вн}$, а в разомкнутом состоянии — не бесконечно большим, а конечным сопротивлением изоляции $R_{из}$

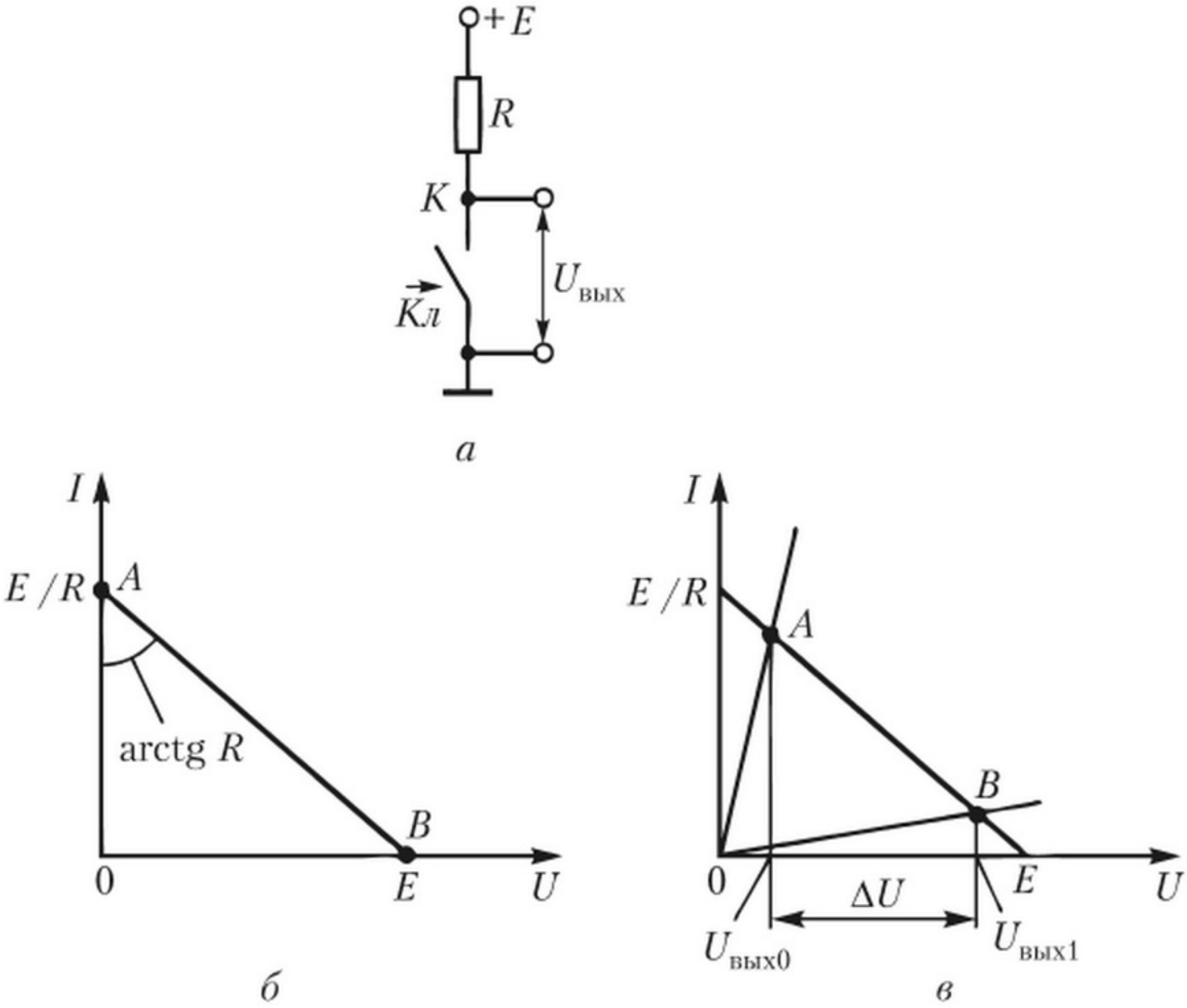


Рис. 3.5. **Ключевая схема (*а*); ее идеальная (*б*) и реальная (*в*) характеристики**

(рис. 3.5, *в*). Точки A и B пересечения нагрузочной прямой с вольт-амперными характеристиками реального ключа сместятся по сравнению с идеальным ключом, диапазон изменения выходного напряжения уменьшится до значения ΔU и, значит, коэффициент использования напряжения $K_и = \Delta U / E < 1$.

3.3.1. Ключевая схема на биполярном транзисторе

В полупроводниковой ключевой схеме роль ключа выполняет биполярный транзистор, включенный по схеме с общим эмиттером, или полевой — по схеме с общим истоком.

Рассмотрим ключ на биполярном транзисторе (рис. 3.6, *а*). Управление таким ключом осуществляет сигнал $U_{вх}$. При $U_{вх} = 0$ ток базы $I_Б$ тоже равен нулю и состояние схемы определяется точкой B (рис. 3.6, *б*) пересечения нагрузочной прямой с выходной характеристикой транзистора при

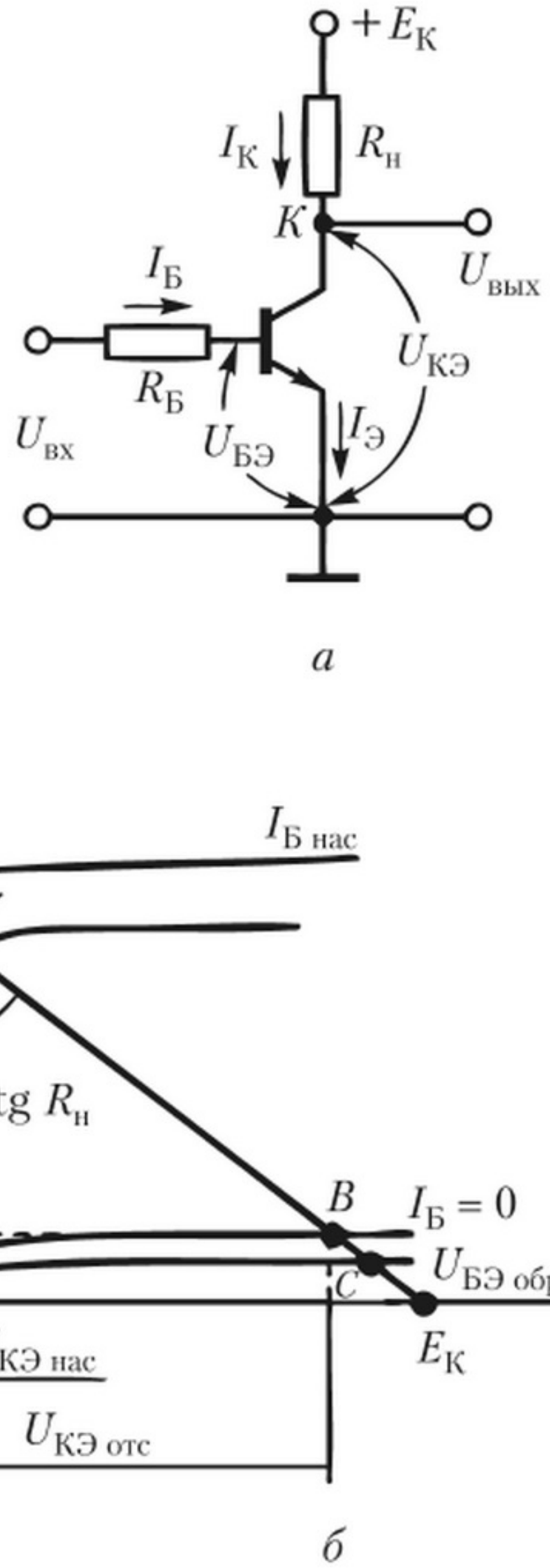


Рис. 3.6. **Ключевая схема на биполярном транзисторе (*а*), ее выходные вольт-амперные характеристики (*б*)**

$I_Б = 0$. Транзистор находится в состоянии отсечки, что равносильно разомкнутому ключу, и выходное напряжение, определяемое потенциалом точки K, равно $U_{КЭ\ отс}$, т.е. несколько меньше, чем $E_К$. При $U_{вх}$, достаточном для создания базового тока $I_{Б\ нас}$, переводящего транзистор в режим насыщения, напряжение $U_{КЭ}$ составляет доли вольт. В этом случае состояние схемы определяется точкой A, что равносильно замкнутому ключу, и выходное напряжение равно $U_{КЭ\ нас}$, т.е. несколько выше нулевого уровня.

Содержание

Таким образом, транзистор ведет себя как *неидеальный ключ* и коэффициент

$$K_{и} = (U_{КЭ\ отс} - U_{КЭ\ нас})/E_{К} < 1.$$

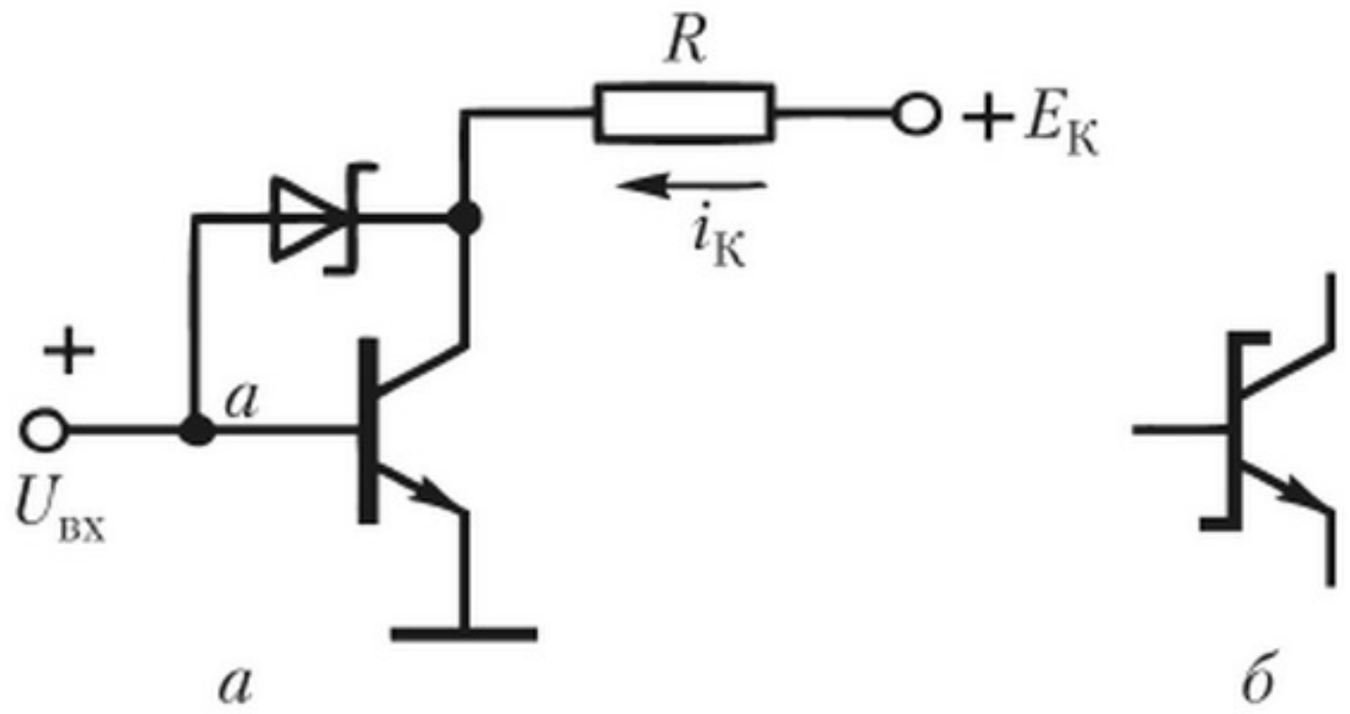
На подобных ключевых схемах реализуются различные функции алгебры логики. Выясним, какую логическую функцию реализует ключевая схема на рис. 3.6, *а*. Если на входе ключа высокий потенциал, то транзистор открыт, ток коллектора создает на сопротивлении нагрузки $R_{н}$ падение напряжения, так что потенциал коллектора — низкий, соответствующий точке *A*. Напротив, когда на входе ключа низкий потенциал, транзистор закрыт, а на его коллекторе высокий потенциал, соответствующий точке *B*. Таким образом, ключевая схема *инвертирует уровень входного сигнала независимо от того, каким уровнем закодированы нуль и единица.*

Следовательно, простейшая ключевая схема на транзисторе с нагрузкой в цепи коллектора, с которого снимается выходное напряжение, является *инвертором, реализующим функцию* НЕ *как в позитивной, так и в негативной логике.*

Переключение транзистора из одного состояния в другое происходит не мгновенно, для этого требуется пусть небольшое, но конечное время. Именно это время определяет быстродействие всех цифровых устройств.

При протекании тока через биполярный транзистор в базе происходит накопление неосновных носителей. Причем чем больше коллекторный ток, тем больше носителей к этому моменту должно быть накоплено в базе, т.е. в базе создается заряд $q_{Б}$. После того как управляющее напряжение $U_{БЭ}$ становится запирающим, коллекторный ток $I_{К}$ продолжает еще некоторое время оставаться неизменным за счет избытка носителей в базе. Этот отрезок времени $t_{р}$ называют *временем рассасывания* неосновных носителей из области базы. Лишь после времени $t_{р}$ происходит переход транзистора из состояния насыщения в состояние отсечки и ток $I_{К}$ снижается до уровня $I_{К0}$, соответствующего запертому состоянию ключевой схемы (точка *B* на вольт-амперных характеристиках).

Этап рассасывания можно устранить, если транзистору после отпирания создать режим, когда он находится на границе между состоянием насыщения и активным режимом работы. Для этого в интегральных схемах используется диод Шоттки, включенный параллельно база-коллектор-



а *б*

Рис. 3.7. **Транзистор Шоттки: принципиальная схема (*а*), условное обозначение (*б*)**

ному переходу транзистора. Такая структура называется *транзистором Шоттки* (рис. 3.7, *а*, *б*).

Напомним, что в активном режиме переход база—коллектор транзистора смещен в обратном направлении (см. параграф 1.3, рис. 1.8, *а*), т.е. потенциал базы, подключенной к $U_{ВХ}$, много ниже потенциала коллектора. С увеличением тока транзистора потенциал коллектора понижается и, приближаясь к насыщению, становится ниже потенциала базы. При этом в базе начинают накапливаться избыточные неосновные носители, увеличивая заряд $q_Б$ и повышая степень насыщения транзистора. Если бы в схеме на рис. 3.7, *а* отсутствовал диод, то с увеличением тока потенциал точки *b* понизился бы настолько, что наступило насыщение транзистора. Этого в схеме не происходит, так как при незначительном (менее 0,1 В) понижении потенциала точки *b* относительно точки *a* отпирается диод Шоттки и избыточный заряд $q_Б$ удаляется из области базы в коллектор.

Диод Шоттки в интегральном исполнении представляет собой контакт металла с коллекторной областью транзистора и составляет единую структуру. В настоящее время транзистор Шоттки является основой для биполярных цифровых интегральных схем (ИС).

3.3.2. Ключевая схема на комплементарных транзисторах

Полевые транзисторы очень широко используются в цифровых ИС. Так же как и биполярный, полевой транзистор может быть положен в основу транзисторного ключа, реализующего функцию НЕ, т.е. инвертора. Однако если ключ на биполярном транзисторе почти не потребляет мощности только в закрытом состоянии, то на полевых транзи-

сторах можно реализовать ключ, практически не потребляющий мощности от источника питания как в закрытом, так и в открытом состоянии.

Создать на МОП-транзисторах инвертор, не потребляющий мощности при любом сигнале на выходе, позволяют так называемые *комплементарные* (взаимодополняющие) транзисторы (КМОП), представляющие два МОП-транзистора *VT*1 и *VT*2 с *каналами противоположного типа*, затворы и стоки которых соединены параллельно (рис. 3.8). Схема симметрична: когда один из транзисторов открыт и выполняет роль замкнутого ключа, другой закрыт и служит нагрузочным сопротивлением.

Рассмотрим работу КМОП-транзисторов при позитивной логике и положительной полярности напряжения питания и сигналов. Исток и подложка транзистора *VT*1 подключены к нулевому потенциалу корпуса, а исток и подложка транзистора *VT*2 — к потенциалу $+E$ источника питания. Поэтому при подаче на вход схемы логического нуля ($U_{вх} \approx 0$) разность потенциалов между затвором и истоком первого транзистора $U_{ЗИ1} = U_{вх} - 0 \approx 0$ и *VT*1 заперт. Напряжение между затвором и истоком второго транзистора $U_{ЗИ2} = U_{вх} - E \approx 0 - E \approx -E$, и транзистор *VT*2 оказывается в состоянии глубокого насыщения. Через него, как через замкнутый ключ, потенциал $+E$ подан на выход ($U_{вых} \approx +E$), реализуя тем самым на выходе логическую единицу. При этом общий для обоих транзисторов ток I_C близок к нулю, так как сопротивление запертого транзистора *VT*1 достаточно велико.

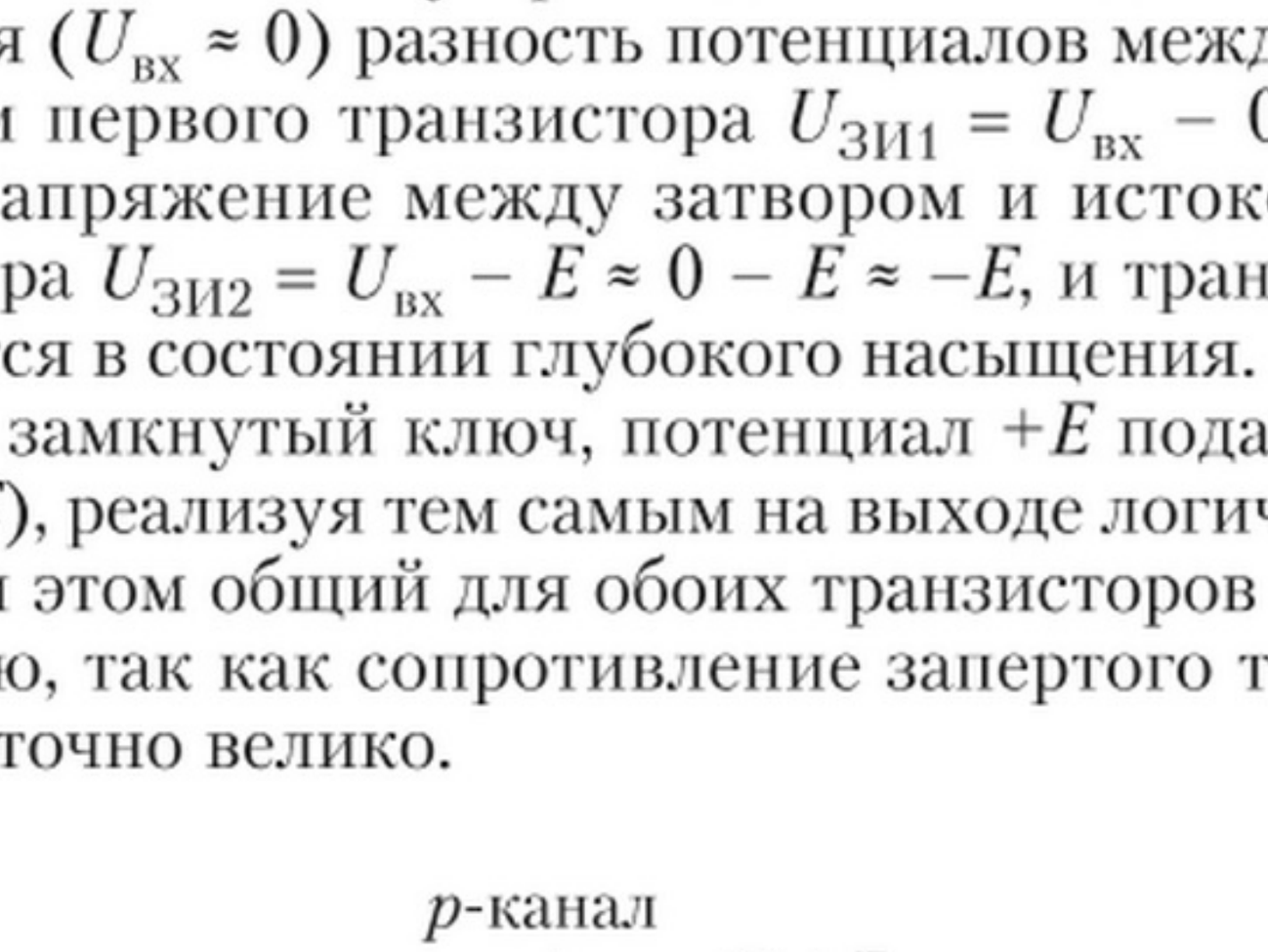


Рис. 3.8. **Инвертор на КМОП-транзисторах**

Содержание

Если же на вход подана логическая единица (т.е. $U_{вх} \approx +E$), то $U_{ЗИ1} = U_{вх} - 0 \approx +E - 0 \approx +E$ и транзистор $VT1$ окажется открытым, а транзистор $VT2$ — закрытым, так как $U_{ЗИ2} = U_{вх} - E \approx +E - E \approx 0$. Значит, через замкнутый ключ $VT1$ на выход будет подан нулевой потенциал корпуса $U_{вых} \approx 0$, реализуя на выходе логический нуль. При этом общий ток I_C останется близким к нулю, потому что будет достаточно большим сопротивление запертого транзистора $VT2$.

Таким образом, в любом статическом состоянии схема практически не потребляет мощности от источника питания. Ток через оба транзистора будет протекать только во время переключения, когда один из транзисторов еще не закроется, а другой — уже приоткроется.

Ключевая схема на КМОП-транзисторах близка к идеальному ключу, и коэффициент использования напряжения источника питания в таком инверторе $K_и = U_{вых}/E \approx 1$.

Работоспособность такого инвертора не зависит от напряжения питания при условии, что оно не меньше удвоенного значения порогового напряжения транзистора ($E \geqslant 2U_{ЗИ\ пор}$), поэтому схема может работать при больших разбросах напряжения питания.

Благодаря высокому значению коэффициента использования напряжения питания, малой потребляемой мощности и высокому быстродействию ключевые схемы на КМОП-транзисторах нашли широкое применение в интегральной схемотехнике.

3.4. Логические элементы интегральных микросхем

3.4.1. Транзисторно-транзисторные логические элементы

Основой для биполярных цифровых ИС является многоэмиттерный транзистор, обеспечивающий реализацию логических операций И и ИЛИ. Он используется во входных цепях микросхем, а вместе с обычным транзистором—инвертором образует базовую логическую схему. Цифровые ИС, в которых для логических операций И и ИЛИ используются многоэмиттерные транзисторы, а для операции НЕ — транзисторный инвертор, получили название *транзисторно-транзисторной логики* или сокращенно ТТЛ.

На рис. 3.9 показана простейшая схема ТТЛ. Если на все входы транзистора $VT1$ подан высокий потенциал, то все

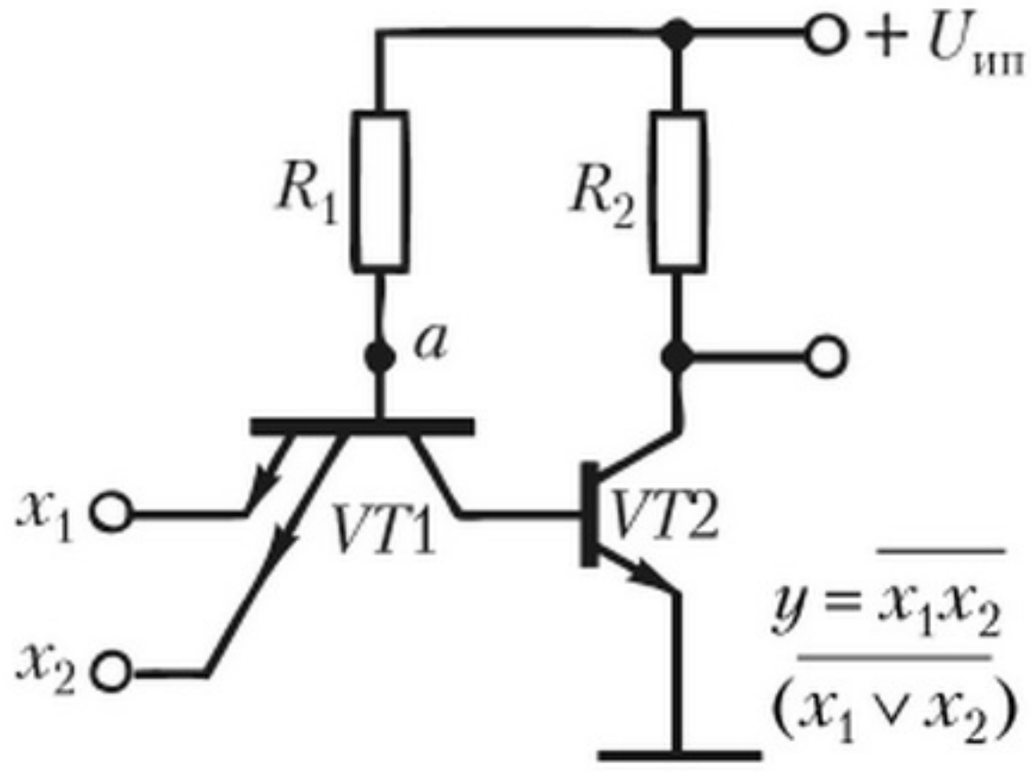


Рис. 3.9. **Простейший элемент ТТЛ**

его эмиттеры заперты и эмиттерные токи отсутствуют. Заметим, что полярность источника питания $U_{ип}$ – прямая для p-n-перехода база–коллектор транзистора $VT1$, и этот переход остается открытым. Поэтому по цепи «$U_{ип}$–R_1–база $VT1$–коллектор $VT1$–база $VT2$–эмиттер $VT2$–корпус» течет ток $I_{Б\ нас}$, который открывает и вводит в насыщение транзистор $VT2$.

Так как $VT2$ и R_2 представляют инвертор (см. рис. 3.6, *а*), при насыщенном $VT2$ на выходе схемы образуется сигнал 0, потому что потенциал $U_{вых}$ окажется близким к низкому потенциалу корпуса. Сопротивление резистора R_1 подобрано так, чтобы за счет падения напряжения на нем от тока $I_{Б\ нас}$ транзистора $VT2$ потенциал точки a (базы транзистора $VT1$) был бы ниже, чем $+U_{вх}$, и эмиттеры $VT1$ оставались бы запертыми.

При подаче низкого потенциала 0 хотя бы на один из входов открывается p-n-переход этого эмиттера, появляется значительный ток $I_Э$, который создает на R_1 падение напряжения $I_Э R_1$, почти целиком уравновешивающее $U_{ип}$. Потенциал точки a приближается к нулевому потенциалу корпуса, в результате чего разность потенциалов между базой и эмиттером транзистора $VT2$ становится близкой к нулю, ток $I_Б$ транзистора $VT2$ прекращается и он переключается в режим отсечки. В результате потенциал $U_{вых} \approx U_{ип}$. Таким образом, в позитивной логике, когда за логическую единицу принят высокий уровень напряжения, многоэмиттерный транзистор $VT1$ выполняет операцию И, а транзистор $VT2$ с резистором R_2 – операцию НЕ, реализуя таким образом базис И-НЕ. В негативной логике, когда за логическую единицу принимается низкий уровень напряжения,

схема реализует базис ИЛИ-НЕ, что и обозначают в скобках на выходе схемы y.

В простейшей схеме ТТЛ транзисторы входят в насыщение, а это ведет к увеличению времени переключения схемы, поэтому в интегральной схемотехнике эта схема в чистом виде не используется. Для устранения режима насыщения транзисторов в настоящее время в ИС используются транзисторы Шоттки, отчего в названии логики появилась буква «Ш» — ТТЛШ.

Базовый элемент ТТЛШ представлен на рис. 3.10. Вместо инвертора, состоящего из транзистора $VT2$ и резистора R_2 в схеме рис. 3.9 применяют сложный инвертирующий каскад, который обеспечивает высокий ток нагрузки и значительно меньшую потребляемую мощность. Сравним эти схемы.

При синтезе различных цифровых устройств одного логического элемента, конечно, недостаточно. Поэтому нагрузкой для логического элемента являются входы аналогичных логических элементов — эмиттеры многоэмиттерных транзисторов. Пусть на выходе схемы обыч-

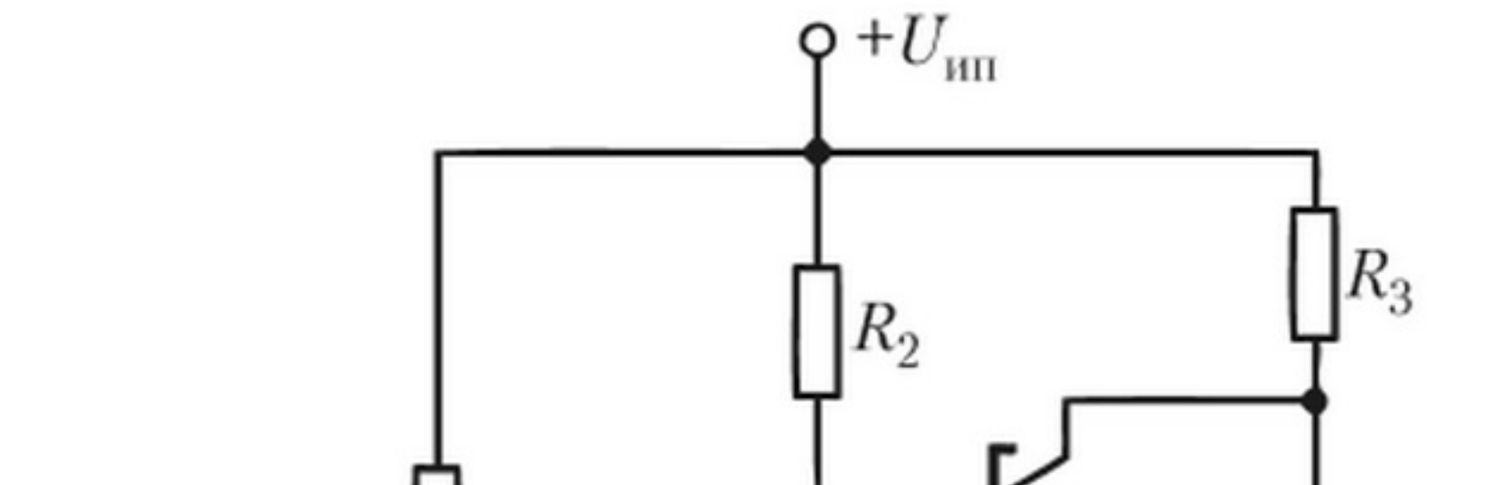


Рис. 3.10. **Базовый элемент ТТЛШ**

Содержание

ного инвертора логический нуль. При этом ток в нагрузку не течет, зато через резистор R_2 будет течь значительный коллекторный ток насыщения, который просто нагревает схему. Когда на выходе схемы логическая единица, транзистор *VT*2 закрыт. Ток в нагрузку будет течь от источника питания через резистор R_2. Чем больше ток нагрузки, тем больше будет падение напряжения на резисторе. При этом мощность, рассеиваемая на резисторе, возрастает, а выходное напряжение, соответствующее логической единице, падает и может стать недопустимо низким.

В схеме сложного инвертора в выходном каскаде применен составной эмиттерный повторитель, состоящий из транзисторов *VT*3 и *VT*5. Теперь ток нагрузки создается транзистором *VT*5. Ток эмиттера транзистора *VT*3 является током базы для транзистора *VT*5, поэтому он может быть в β_5 раз меньше тока нагрузки (β_5 — коэффициент передачи тока транзистора *VT*5). Ток через резистор R_2 теперь является током базы для транзистора *VT*3 и может быть меньше его эмиттерного тока в β_3 раз (β_3 — коэффициент передачи тока транзистора *VT*3). Таким образом, ток через резистор R_2 может быть меньше тока нагрузки в $\beta_3 \cdot \beta_5$ раз. Ток открытого транзистора *VT*2 в схеме сложного инвертора, так же как и в схеме обычного инвертора, обеспечивает уровень логического нуля. Но теперь он может быть гораздо меньше. Он должен лишь обеспечивать открытое состояние транзистора *VT*4, который обеспечивает на выходе схемы потенциал, близкий к потенциалу корпуса. Каскад, состоящий из транзистора *VT*6 и резисторов R_4, R_6, необходим для ускоренного включения транзистора *VT*4. В момент переключения схемы из состояния логической единицы в состояние логического нуля транзисторы *VT*2, *VT*4 и *VT*6 закрыты и переход база—эмиттер транзистора *VT*6 имеет высокое сопротивление. Сопротивление транзистора *VT*6, высокое в начале переходного процесса, складывается с сопротивлением R_4, и эмиттерный ток транзистора *VT*2 на их общем сопротивлении создает увеличенное падение напряжения, которое быстрее открывает транзистор *VT*4. В момент выключения транзисторов при переходе из логического нуля к логической единице транзистор *VT*6 открыт, что позволяет транзистору *VT*2 быстрее перейти в запертое состояние.

Для повышения помехоустойчивости эмиттеры *VT*1 соединены с корпусом через диоды *VD*1, *VD*2, запертые для

входных сигналов положительной полярности. Они открываются только при отрицательной полярности напряжений на входах, что наблюдается при переходных процессах, когда из-за паразитных индуктивностей и емкостей в цепях, подключенных к входам базового элемента, возникают затухающие колебания. Их значительная начальная амплитуда может вызвать ложное срабатывание входной логики. При включении диодов этого не происходит, так как первая же отрицательная полуволна помехи открывает диод и замыкается через него. При этом на соответствующем эмиттере создается напряжение, равное ЭДС отпирания диода, не превышающее долей вольт. Следующие затухающие как отрицательные, так и положительные полуволны помехи имеют еще меньшую амплитуду, не превышающую сигнала логической единицы и, значит, не вызывающие ложного срабатывания элемента.

Первоначально на основе элементов ТТЛШ разрабатывались быстродействующие цифровые вычислительные устройства и, в частности, ЭВМ. Однако в дальнейшем совершенствование технологии КМОП привело к вытеснению элементов ТТЛШ, поскольку элементы КМОП обладают гораздо меньшей потребляемой мощностью. Тем не менее элементы ТТЛШ в настоящее время продолжают использоваться главным образом во входных и выходных цепях вычислительных устройств. Это объясняется их высокой помехоустойчивостью, способностью работать на емкостную нагрузку, которая всегда присутствует там, где выходные линии связи имеют протяженность более нескольких дециметров.

3.4.2. Логические элементы на КМОП-транзисторах

Как было показано на примере инвертора, выполненного на КМОП-транзисторах, отличительной особенностью таких МОП-структур является их свойство не потреблять мощности в статическом режиме. При любом сочетании сигналов открывание МОП-транзистора одного типа сопровождается запиранием МОП-транзистора противоположного типа.

Общая закономерность построения таких структур состоит в том, что параллельное соединение одного типа транзисторов сопровождается последовательным соединением транзисторов противоположного типа. Проиллюстри-

руем это положение на базовых элементах, реализующих функции ИЛИ-НЕ и И-НЕ.

В схемах (рис. 3.11) пары транзисторов *VT*1, *VT*3 и *VT*2, *VT*4 образуют комплементарные структуры: когда один из них заперт, то другой открыт. Пусть в схеме на рис. 3.11, *а* на оба входа x_1 и x_2 подан логический нуль. Тогда транзисторы *n*-типа *VT*1и *VT*2 заперты, так как разность потенциалов между их затворами и истоками близка к нулю. Транзисторы *p*-типа *VT*3 и *VT*4 открыты, потому что нулевой потенциал, подведенный к затворам относительно их подложек, создаст отрицательную разность потенциалов, необходимую для индуцирования в них *p*-канала. Но так как через эти каналы протекают лишь пренебрежительно малые токи запертых транзисторов *VT*1 и *VT*2, падения напряжения на транзисторах *VT*3 и *VT*4 незначительны и, значит, выходное напряжение, почти равное $+U_{\text{ип}}$, соответствует логической единице.

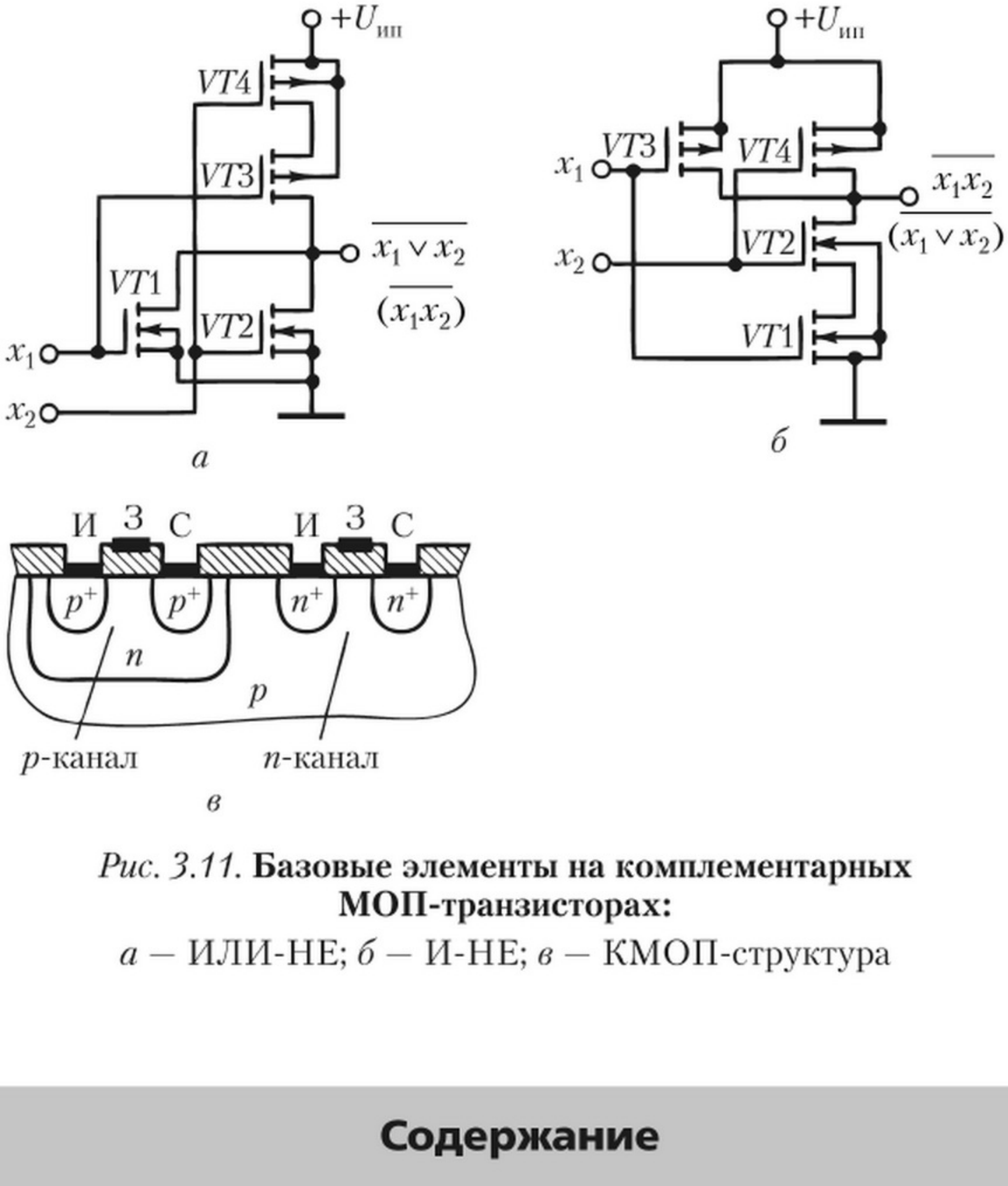


Рис. 3.11. **Базовые элементы на комплементарных МОП-транзисторах:**
а — ИЛИ-НЕ; *б* — И-НЕ; *в* — КМОП-структура

Содержание

Если хотя бы на один из входов, например x_1, подать логическую единицу, т.е. потенциал, близкий к $+U_{ип}$, то соответствующий p-транзистор (в данном случае $VT3$) закроется, отключая $+U_{ип}$ от выхода, на котором через открывшийся транзистор n-типа (в рассматриваемом примере $VT1$) подан нулевой потенциал корпуса, т.е. логический нуль. Таким образом, схема реализует функцию ИЛИ-НЕ. Как и в случае инвертора, в рассмотренной схеме перепад выходных напряжений близок к напряжению питания (коэффициент использования напряжений ключевой схемы близок к единице). Поэтому помехоустойчивость логических ИМС на КМОП-структурах высока.

Аналогично можно разобрать работу схемы, реализующей функцию И-НЕ (рис. 3.11, *б*). Как и в логике ТТЛШ, в рассматриваемых базовых элементах реализуется принцип двойственности, т.е. при смене позитивной логики на отрицательную одна и та же схема может выполнять функции как ИЛИ-НЕ, так и И-НЕ.

К преимуществам элементов на КМОП-структурах (рис. 3.11, *в*) относится способность работать без сбоев при больших разбросах напряжения питания.

Благодаря своим уникальным параметрам — малой потребляемой мощности, высокой помехоустойчивости, широким допускам на величину питающих напряжений, высоком быстродействии при небольших емкостных нагрузках — КМОП-элементы получили широкое распространение в интегральной схемотехнике. Эти элементы являются доминирующими в микропроцессорных БИС/СБИС, полупроводниковых запоминающих устройствах и СБИС программируемой логики. В современных БИС/СБИС на основе КМОП-структур выполняются внутренние области микросхем, в то время как на ТТЛШ-структурах — область периферийных схем, где требуется передача сигналов по внешним (по отношению к микросхеме) цепям, испытывающим значительную емкостную нагрузку.

3.5. Дешифраторы и шифраторы

Как отмечалось в параграфе 3.2, цифровые устройства делятся на комбинационные и последовательностные. К комбинационным относятся такие цифровые устройства,

выходные сигналы которых зависят только от текущего значения входных сигналов. Эти устройства, в отличие от последовательностных, не обладают памятью. После завершения переходных процессов в этих устройствах на их выходах устанавливаются выходные величины, на которые характер переходных процессов влияния не оказывает.

Любое сложное цифровое устройство может быть разделено на комбинационную часть, выполняющую логические операции, и элементы памяти. В принципе комбинационная часть может быть выполнена на логических элементах, однако это слишком сложно и дорого. Гораздо проще для этого использовать готовые комбинационные устройства. К основным комбинационным устройствам относят дешифраторы, шифраторы, мультиплексоры (распределители), демультиплексоры и сумматоры.

3.5.1. Дешифраторы

Дешифратор (*decoder*) — *это комбинационное устройство, позволяющее распознавать числа, представленные позиционным n-разрядным кодом*. Если на входе дешифратора n-разрядный двоичный код, то на его выходе код «1 из N». В кодовой комбинации этого кода только одна позиция занята единицей, а все остальные — нулевые. Например, код «1 из N», содержащий 4 кодовые комбинации, будет представлен следующим образом:

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Такой код называют *унитарным*, поэтому дешифратор является преобразователем позиционного двоичного кода в унитарный. Так как возможное количество чисел, закодированных n-разрядным двоичным кодом, равно количеству наборов из n аргументов ($N = 2^n$), то дешифратор, имеющий n входов, должен иметь 2^n выходов. Такой дешифратор называют полным. Если часть входных наборов не используется, то дешифратор называют неполным, и у него число выходов меньше 2^n. Таким образом, в зависимости от входного двоичного кода на выходе дешифратора возбуждается только одна из выходных цепей, по номеру которой можно распознать входное число.

Дешифраторы применяют для расшифровки адресов ячеек запоминающих устройств, высвечивания букв и цифр на мониторах, индикаторах и других устройствах. Чаще всего они являются встроенными в БИС, как, например, в полупроводниковых запоминающих устройствах, однако они выпускаются и в виде ИС среднего уровня интеграции.

Проиллюстрируем реализацию дешифраторов на примере полного дешифратора трехразрядных чисел. Таблица истинности дешифратора представлена в табл. 3.5.

Таблица 3.5

Номера наборов	Входы			Выходы							
	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	0	**1**	0	0	0	0	0	0	0
1	0	0	1	0	**1**	0	0	0	0	0	0
2	0	1	0	0	0	**1**	0	0	0	0	0
3	0	1	1	0	0	0	**1**	0	0	0	0
4	1	0	0	0	0	0	0	**1**	0	0	0
5	1	0	1	0	0	0	0	0	**1**	0	0
6	1	1	0	0	0	0	0	0	0	**1**	0
7	1	1	1	0	0	0	0	0	0	0	**1**

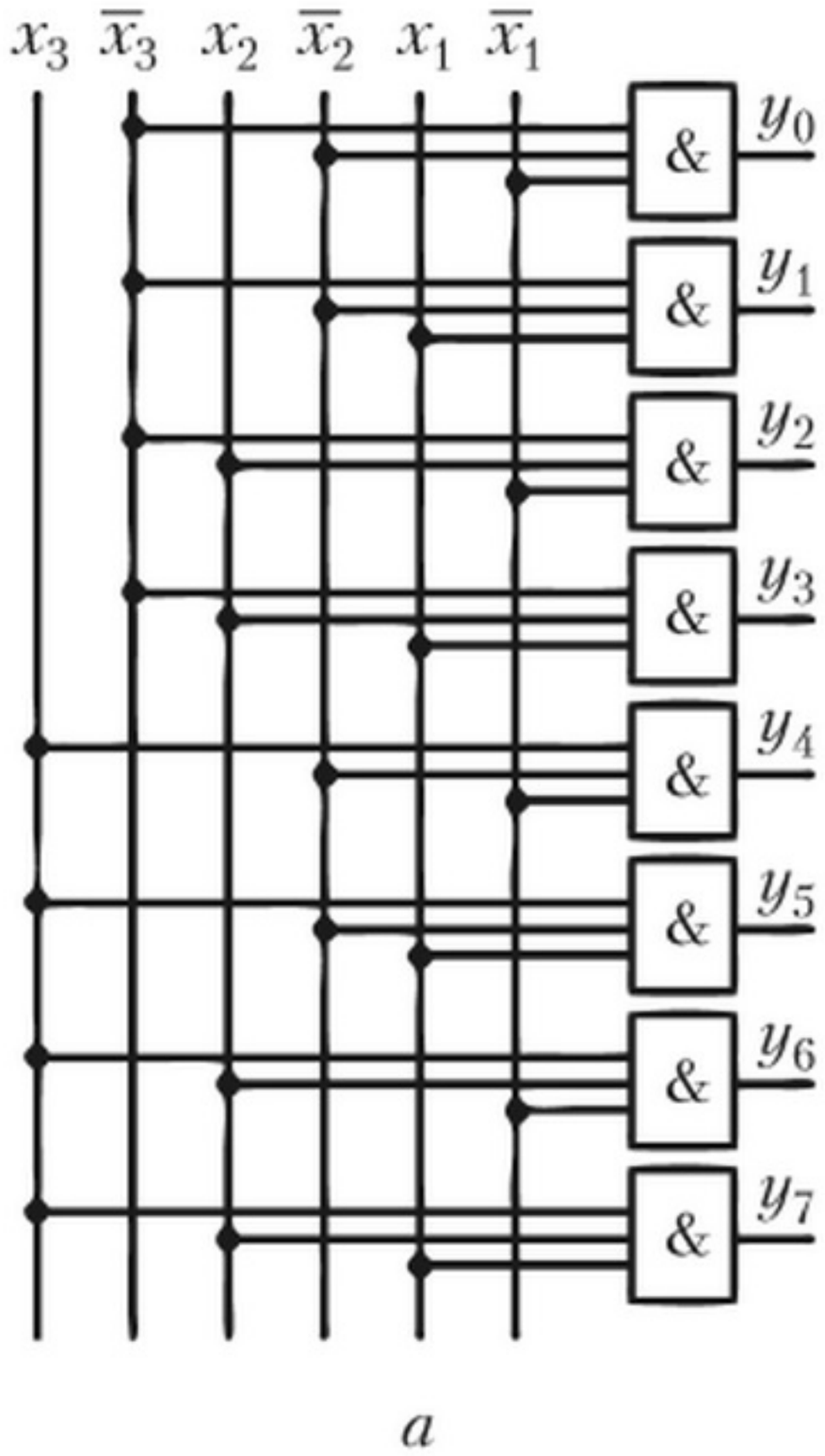
Как видно, каждый выход y_i равен единице только на одном наборе, поэтому работа дешифратора описывается восемью функциями — по числу выходов дешифратора, каждая из которых является конъюнкцией (логическим И) трех аргументов:

$$y_0 = \overline{x_3}\,\overline{x_2}\,\overline{x_1};\quad y_1 = \overline{x_3}\,\overline{x_2}\,x_1;\quad y_2 = \overline{x_3}\,x_2\,\overline{x_1};$$

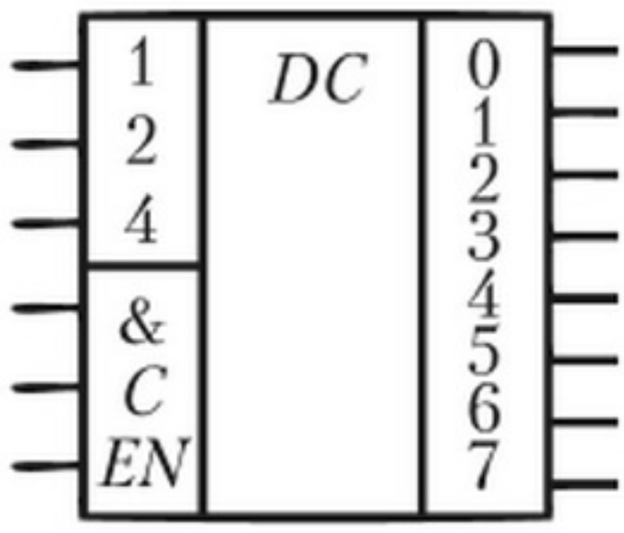
$$y_3 = \overline{x_3}\,x_2\,x_1;\quad y_4 = x_3\,\overline{x_2}\,\overline{x_1};\quad y_5 = x_3\,\overline{x_2}\,x_1;$$

$$y_6 = x_3\,x_2\,\overline{x_1};\quad y_7 = x_3\,x_2\,x_1.$$

Схема трехразрядного полного дешифратора показана на рис. 3.12. Для реализации одной функции y_i нужен один трехвходовый конъюнктор. Так как на входах конъюнкторов присутствуют как прямые значения аргументов, так



а



б

Рис. 3.12. **Линейный дешифратор на три входа:**
а — логическая схема; *б* — условное обозначение дешифратора с входами синхронизации и разрешения

и инверсные, в схеме дешифратора необходимы три инвертора (см. рис. 3.12, *а*).

Часто дешифраторы выполняют с управляемой синхронизацией, при которой дешифрация кода будет произведена во время подачи синхронизирующего импульса, поступившего на вход *C*, лишь при условии, что на вход *EN* подан разрешающий единичный сигнал (см. рис. 3.12, *б*). Для реализации такого условия необходимы конъюнкторы с четырьмя входами, на четвертый вход которых поступает сигнал разрешения. Этот сигнал формируется двухвходовым конъюнктором при совпадении сигналов *C* и *EN*.

Число контактов у стандартного корпуса несложной ИС ограничено (14, 16 или 24), поэтому дешифраторы, выпускаемые в виде ИС, имеют небольшую разрядность входного кода (три, реже четыре). Так, например, в 16-контактном корпусе может быть помещен лишь трехразрядный полный дешифратор. Если требуется создать дешифратор большей разрядности, используют каскадное соединение дешифраторов небольшой разрядности.

Пример 3.1. Пусть на основе трехразрядных дешифраторов необходимо создать пятиразрядный (рис. 3.13).

Решение. Пятиразрядный дешифратор должен иметь $2^5 = 32$ выходов. Разделим пять разрядов на младшие x_2, x_1, x_0 и старшие x_4, x_3. Тогда младшие можно подать на входы четырех 3-разрядных дешифраторов второго каскада и сформировать $8 \cdot 4 = 32$ выхода. Используя входы разрешения *EN*, можно выбирать один из четырех дешифраторов второго каскада, на котором должен сформироваться единичный сигнал. Для этого старшие два разряда подадим на входы управляющего дешифратора первого каскада, а его выходы подключим к входам разрешения *EN* дешифраторов первого каскада.

Пусть, например, входной код равен $11011 = 27_{10}$. Так как старшие разряды — «11», то управляющий дешифратор разрешит работу 4-го дешифратора второго каскада. При

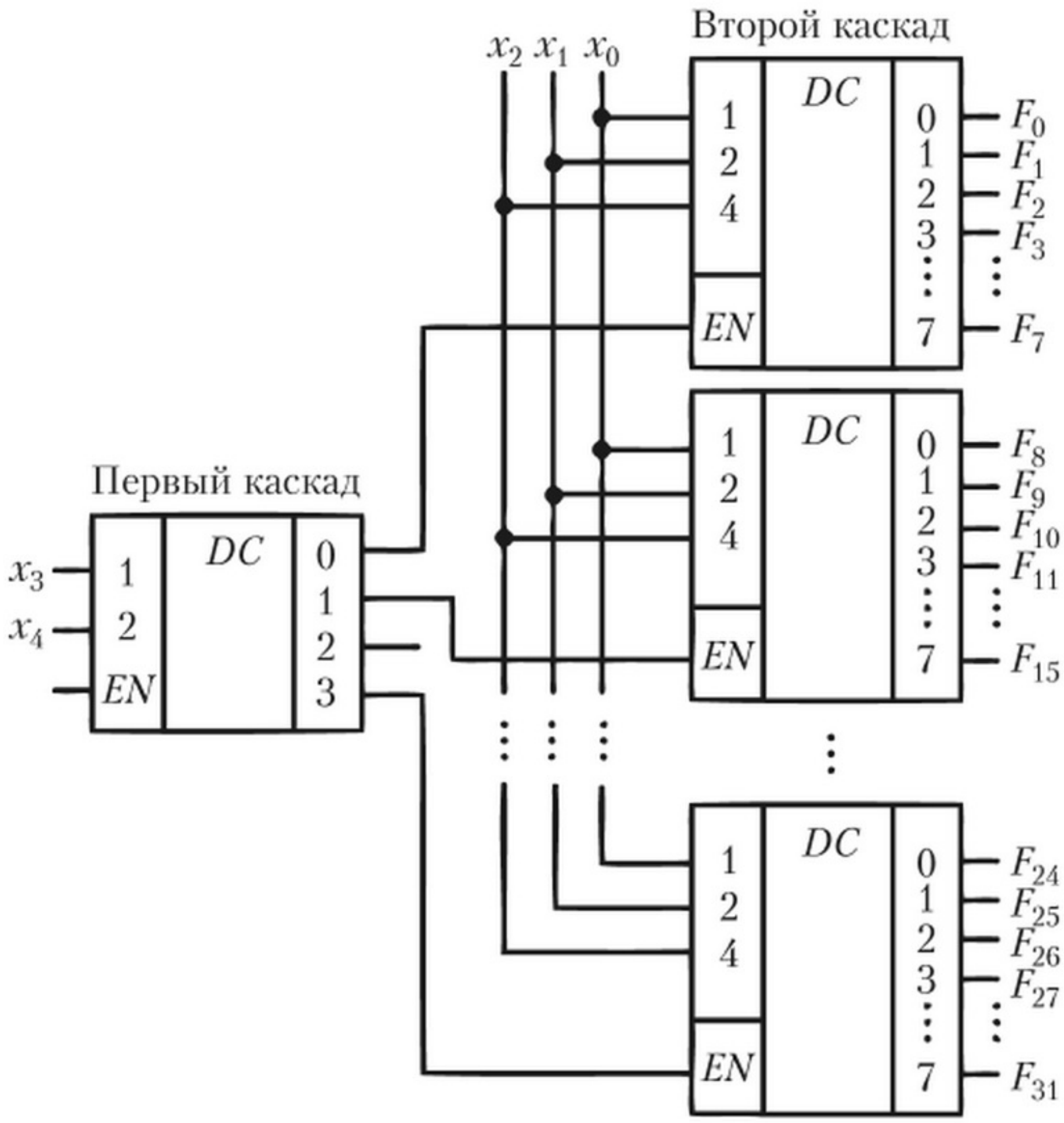


Рис. 3.13. **Схема наращивания размерности дешифратора**

этом на выходах первых трех дешифраторов будут нули, а на выходе «3» четвертого дешифратора, т.е. F_{27}, будет логическая единица.

Дешифраторы широко применяются в системах управления технологическими процессами. Многие исполнительные устройства, такие, как электродвигатель, исполнительный механизм на основе электромагнита, могут управляться всего двумя командами: «включить» и «выключить». При этом команде «включить» удобно сопоставить логическую «1», а команде «выключить» — логический «0». Для управления такими устройствами используют унитарные коды, в которых каждый разряд жестко связан с конкретным устройством. Количество управляемых устройств может составлять несколько десятков, и дешифратор должен иметь соответствующее число выходов.

На рис. 3.14 показана схема управления восемью исполнительными устройствами на основе дешифратора. Схема содержит восемь аналогичных цепей, обеспечивающих включение/отключение исполнительного устройства. Состояние исполнительного устройства фиксируется элементом памяти, в качестве которого чаще всего используется триггер (см. параграф 3.9). Верхний вход обеспечивает включение элемента, а нижний — выключение. Сигнал, определяющий включенное или выключенное состояние, поступает на соответствующие схемы И (верхние или нижние) всех элементов памяти, но воспринимается этот сигнал только тем элементом, который выбирается дешифратором. Для этого на схему управления вместе с сигналами ВКЛ/ВЫКЛ одновременно подается код, поступающий

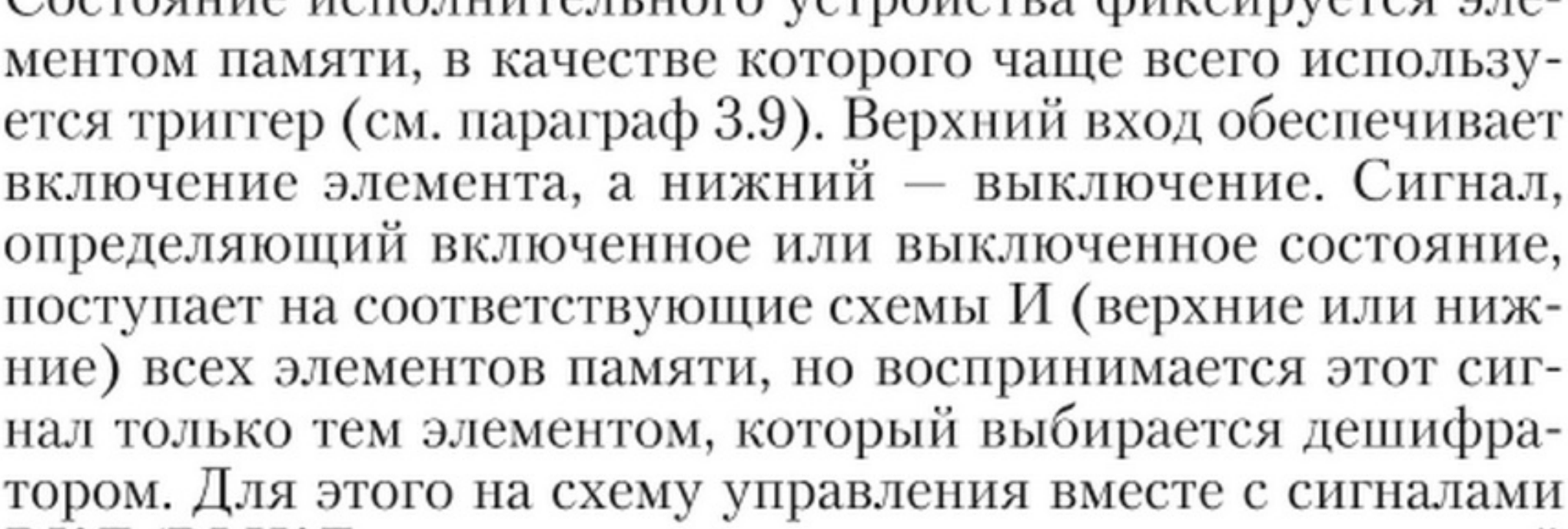


Рис. 3.14. **Схема управления исполнительными устройствами технологической системы на основе дешифратора**

Содержание

на дешифратор и определяющий номер исполнительного устройства. Сигнал с выхода элемента памяти усиливается и поступает в цепь включения исполнительного устройства. Здесь возможна установка оптронной гальванической развязки (см. параграф 2.10), электромагнитного реле, обеспечивающего подачу высокого включающего напряжения, например ≈220 В, электромагнитного пускателя, подающего трехфазное напряжение на электродвигатель.

3.5.2. Шифраторы

Шифратор (*coder*) — *это комбинационное устройство, выполняющее функции, обратные дешифратору*. При подаче сигнала на один из его входов (унитарный код) на выходе должен образоваться соответствующий двоичный код.

Если число входов шифратора равно 2^n, то число выходов, очевидно, должно быть равным n, т.е. числу разрядов двоичного кода, которым можно закодировать 2^n ситуаций.

Проиллюстрируем синтез схемы шифратора при $n = 3$. Таблица истинности имеет вид, приведенный в табл. 3.6.

Таблица 3.6

Входы x	Выходы			Входы x	Выходы		
	y_3	y_2	y_1		y_3	y_2	y_1
0	0	0	0	4	1	0	0
1	0	0	1	5	1	0	1
2	0	1	0	6	1	1	0
3	0	1	1	7	1	1	1

Работа шифратора описывается тремя функциями y_3, y_2, y_1, каждая из которых равна единице на четырех наборах (номер набора соответствует номеру входа). СовДНФ функций выхода равны:

$$y_1 = x_1 \vee x_3 \vee x_5 \vee x_7;\ \ y_2 = x_2 \vee x_3 \vee x_6 \vee x_7;\ \ y_3 = x_4 \vee x_5 \vee x_6 \vee x_7.$$

Три функции реализуются тремя дизъюнкторами (рис. 3.15), на выходах которых формируется трехразрядный двоичный код.

При этом аргумент x_0 не входит ни в одну из логических функций и шина x_0 остается незадействованной. Действительно, входному сигналу x_0 должен соответствовать код

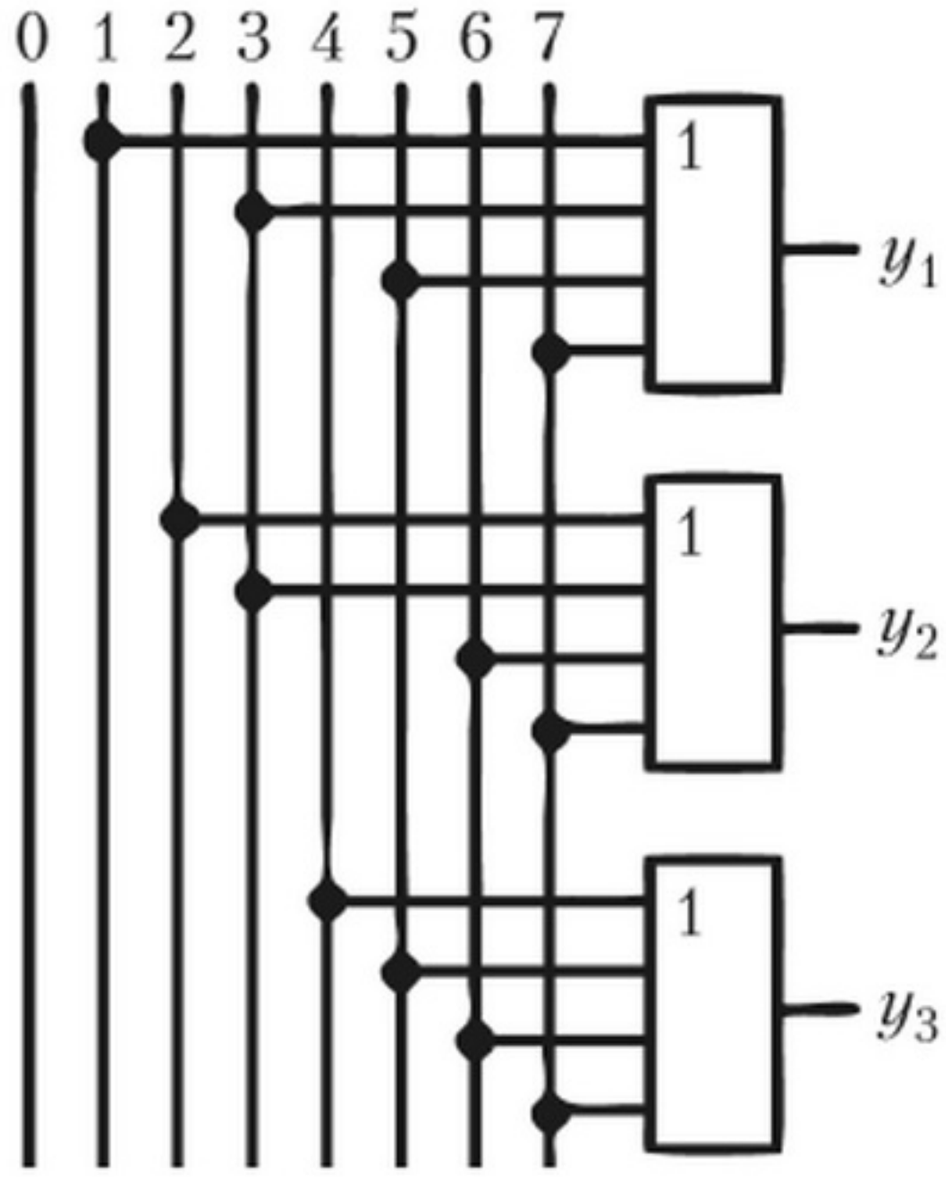


Рис. 3.15. **Шифратор на три выхода**

«000», который все равно будет на выходе шифратора, если все остальные аргументы равны нулю.

Кроме обычных шифраторов существуют также *приоритетные шифраторы*. Такие шифраторы выполняют более сложную операцию. При работе ЭВМ и других устройств часто решается задача определения приоритетного претендента на обслуживание. Несколько конкурентов выставляют свои запросы на обслуживание, которые не могут быть удовлетворены одновременно. Нужно выбрать, кому предоставляется право первоочередного обслуживания. Простейший вариант задачи — присвоение каждому источнику запросов фиксированного приоритета. Например, группа из восьми запросов R_7, ..., R_0 (R — от англ. *request* — запрос) формируется так, что высший приоритет имеет источник номер семь, а далее приоритет уменьшается от номера к номеру. Самый младший приоритет у нулевого источника — он будет обслуживаться только при отсутствии всех других запросов. Если имеются одновременно несколько запросов, обслуживается запрос с наибольшим номером.

Приоритетный шифратор вырабатывает на выходе двоичный номер старшего запроса. При наличии всего одного возбужденного входа приоритетный шифратор работает так же, как и двоичный. Поэтому в сериях ИС двоичный шифратор как самостоятельный элемент может отсутство-

вать. Режим его работы — частный случай работы приоритетного шифратора.

3.6. Распределители и мультиплексоры

3.6.1. Распределители

Распределитель — это устройство, передающее импульс, поступивший на его вход x, на один из выходов в зависимости от управляющего сигнала, заданного двоичным кодом.

Распределитель выполняется на основе дешифратора (рис. 3.16). Управляющий сигнал УС в виде параллельного двоичного кода подается на входы дешифратора, выходы которого соединены со средними входами конъюнкторов, составляющих выходную часть схемы. На входах *C* и *EN* поддерживается логическая единица. На верхние входы конъюнкторов поступает входной импульс, который пройдет лишь через тот конъюнктор, на который от дешифратора подается разрешающий единичный сигнал. Пусть, например, единичный сигнал поддерживается на выходе 0 дешифратора. Тогда импульсы линии x будут передаваться через верхний конъюнктор на выход y_1. Если управляющий код изменится, и единичный сигнал появится, например, на выходе 7 дешифратора, то входные импульсы x будут передаваться через нижний конъюнктор на выход 8 распределителя.

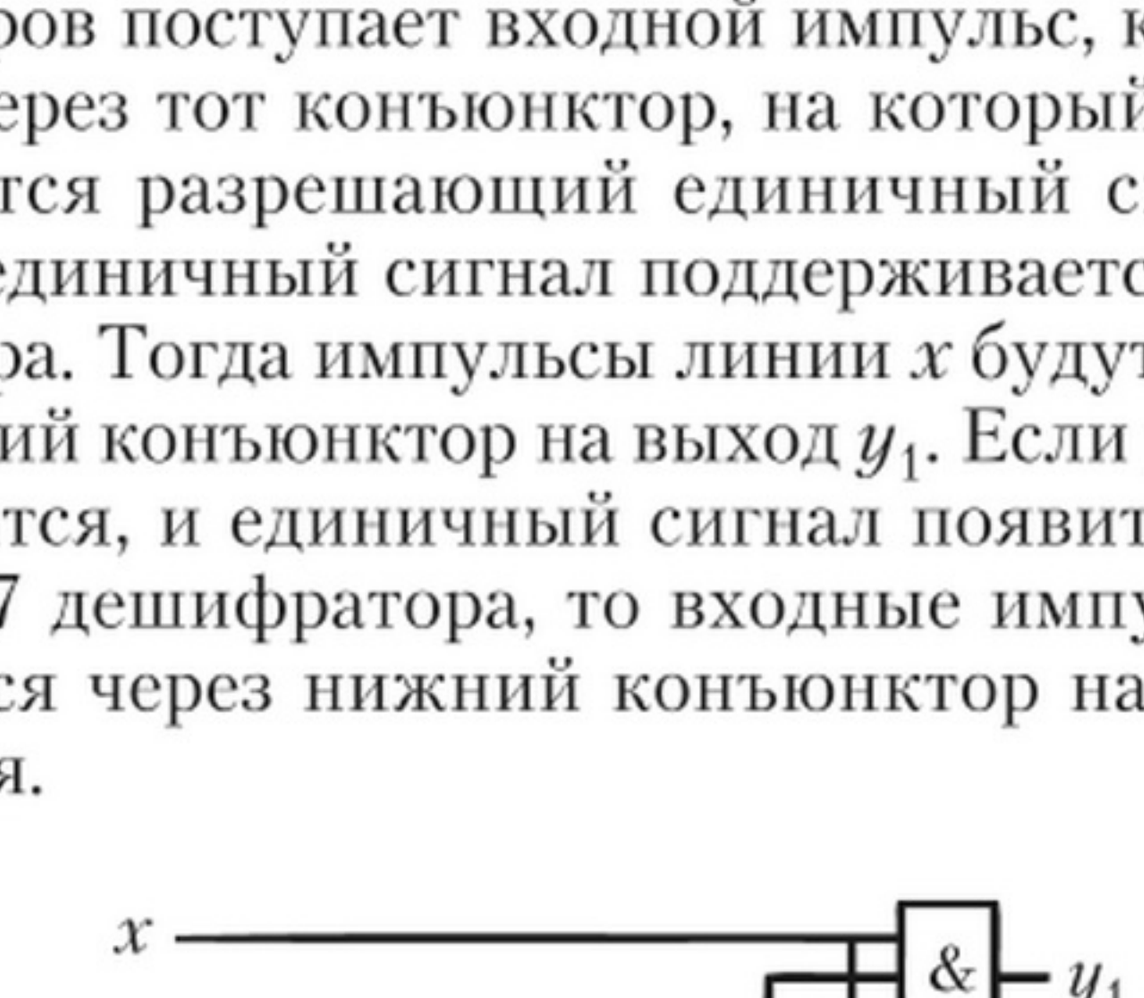


Рис. 3.16. **Распределитель на восемь выходов**

Такой распределитель может быть выполнен синхронным. Для этого могут быть использованы либо входы *C* и *EN* дешифратора, либо третьи (на рис. 3.17 — нижние) входы конъюнкторов, на которые подаются сигналы синхронизации СИ.

3.6.2. Мультиплексоры

Мультиплексор — это устройство, подключающее единственный выходной канал к одному из входов в зависимости от управляющего сигнала, заданного двоичным кодом. Таким образом, мультиплексор решает задачу, обратную распределителю. Иными словами, мультиплексор позволяет производить прием сигналов с различных направлений.

Разрядность n управляющего сигнала определяет количество входов 2^n, с которых мультиплексор позволяет осуществлять прием информации.

На рис. 3.17, *а* приведена функциональная схема мультиплексора на четыре входа x_0—x_3, управляемая двухразрядным кодом a_0a_1 — адресом выбираемого входа.

Управляющий сигнал, распознаваемый дешифратором, формирует единичный сигнал на одном из его выходов, который, поступая на нижний вход одной из схем И, пропускает на выход y (через дизъюнктор ИЛИ) импульсы с той входной линии, которая подключена к верхнему входу

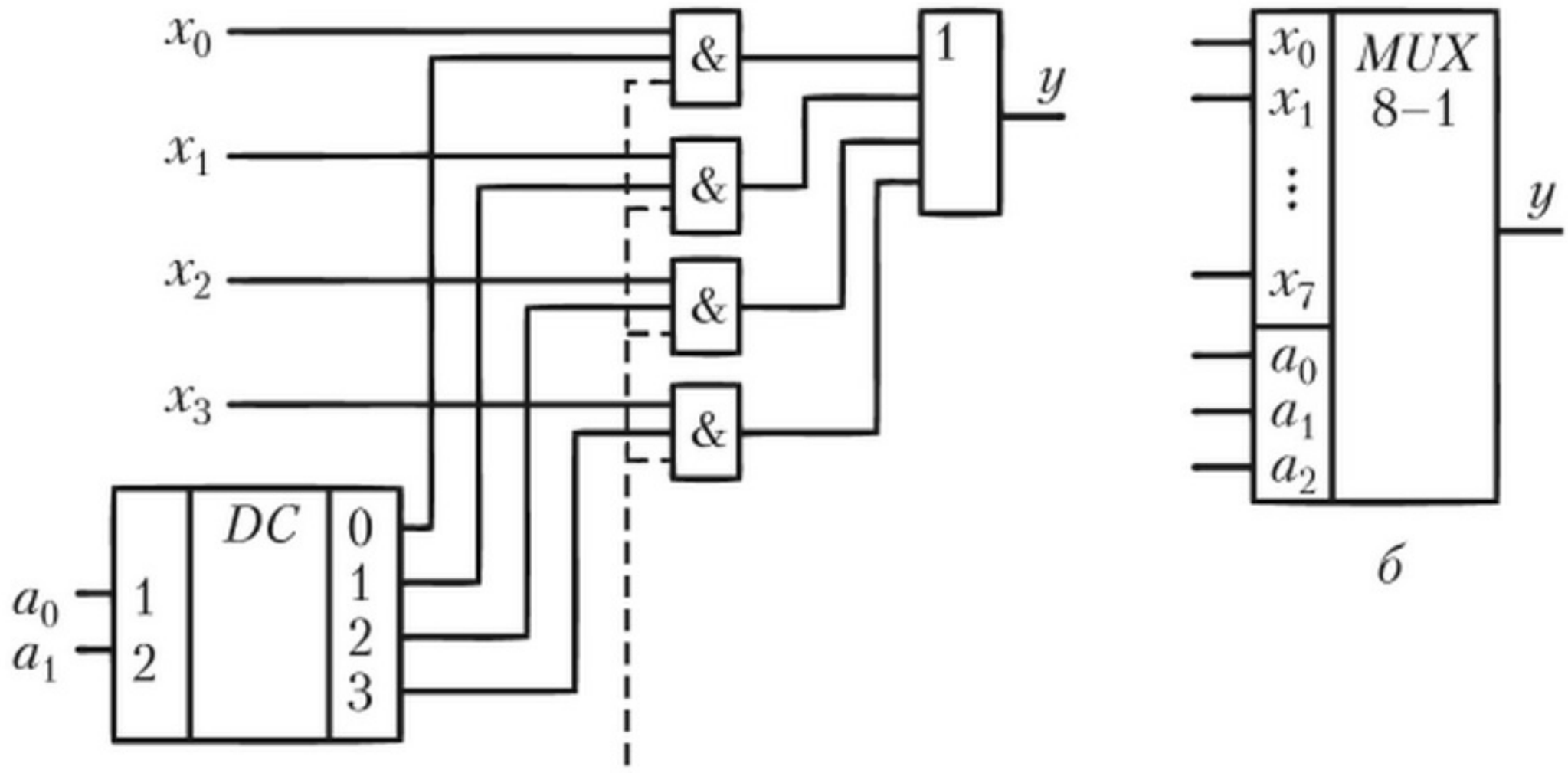


Рис. 3.17. **Мультиплексор:**
а — функциональная схема; *б* — условное обозначение

данной схемы И. Мультиплексор может быть синхронизирован подачей на конъюнкторы синхроимпульсов, как показано пунктиром на рис. 3.17, *а*. Условное обозначение мультиплексора на восемь входов приведено на рис. 3.17, *б*.

3.6.3. Реализация логических функций на основе мультиплексоров

На основе мультиплексоров может быть реализована любая логическая функция, благодаря чему встроенные в микросхему мультиплексоры широко применяются в СБИС программируемой логики.

Покажем, как на основе мультиплексора можно реализовать логическую функцию «сумма по модулю 2», представленную таблицей истинности, приведенной в табл. 3.7.

Таблица 3.7

Номера набора	x_1	x_0	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Так как мультиплексор может пропустить на выход сигнал с любого входа, адрес которого установлен на соответствующих адресных входах, подадим на входы 0—3 сигналы «0» и «1» в соответствии с F. Номер входа при этом будет соответствовать номеру набора (рис. 3.18). Теперь если на адресные входы мультиплексора a_1a_0 подавать аргу-

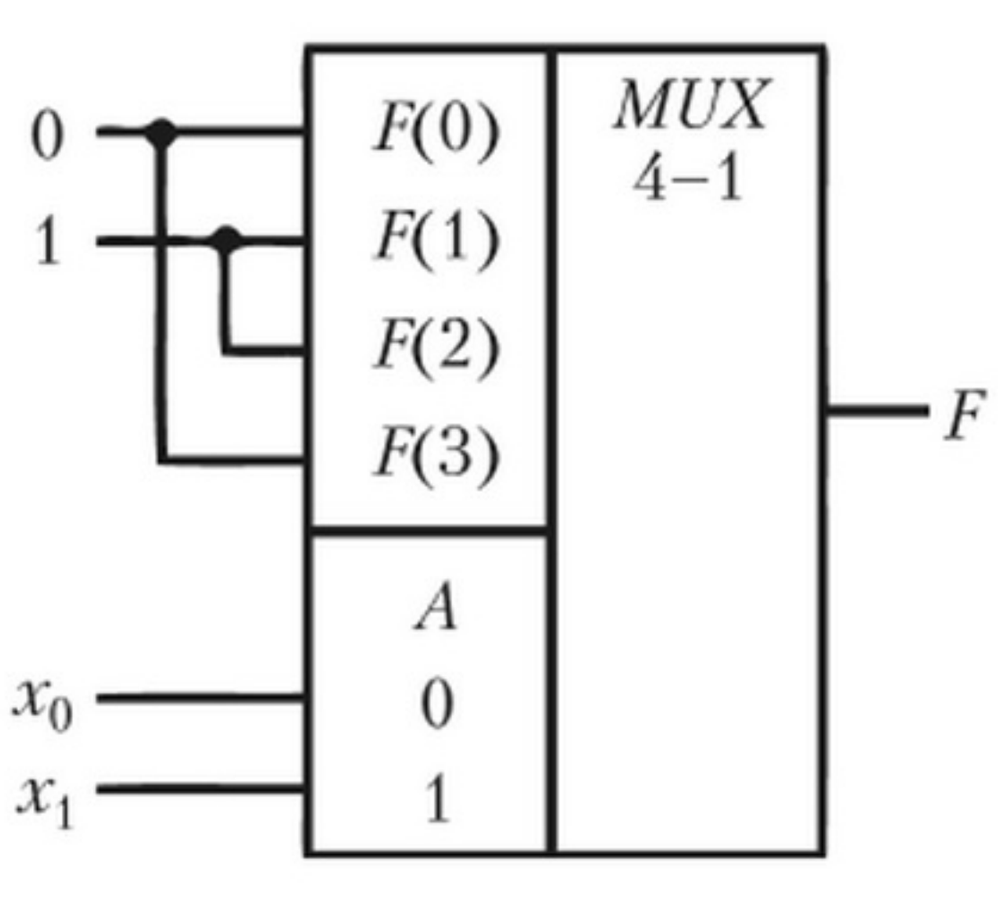


Рис. 3.18. **Реализация функции «сумма по модулю два»**

менты функции x_1x_0, значения функции F будут формироваться на выходе мультиплексора в соответствии с таблицей истинности.

Таким образом, чтобы мультиплексор выполнял логическую функцию, на его информационные входы необходимо подать логические «1» или «0» в соответствии с таблицей истинности, а на адресные входы — аргументы функции.

Рисунок 3.19 иллюстрирует возможность воспроизведения с помощью мультиплексора любой функции n аргументов. Действительно, каждому набору аргументов соответствует передача на выход одного из сигналов настройки. Если этот сигнал есть значение функции на данном наборе аргументов, то задача решена. Разным функциям будут соответствовать разные коды настройки. Информационные входы становятся входами настройки булевой функции, а адресные — входами аргументов.

В примере с реализацией функции «сумма по модулю 2» используется мультиплексор с двумя адресными входами, число которых равно числу аргументов функции. Однако возможны ситуации, когда с помощью такого мультиплексора можно реализовать и более сложные функции.

Пусть, например, булева функция трех аргументов имеет вид:

$$F = x_1x_2x_3 \quad \overline{x_2}\,\overline{x_3} \vee$$

Аргумент x_1 входит в функцию только один раз (в конъюнкцию $x_1x_2x_3$), в то время как x_2 и x_3 — два раза. Это

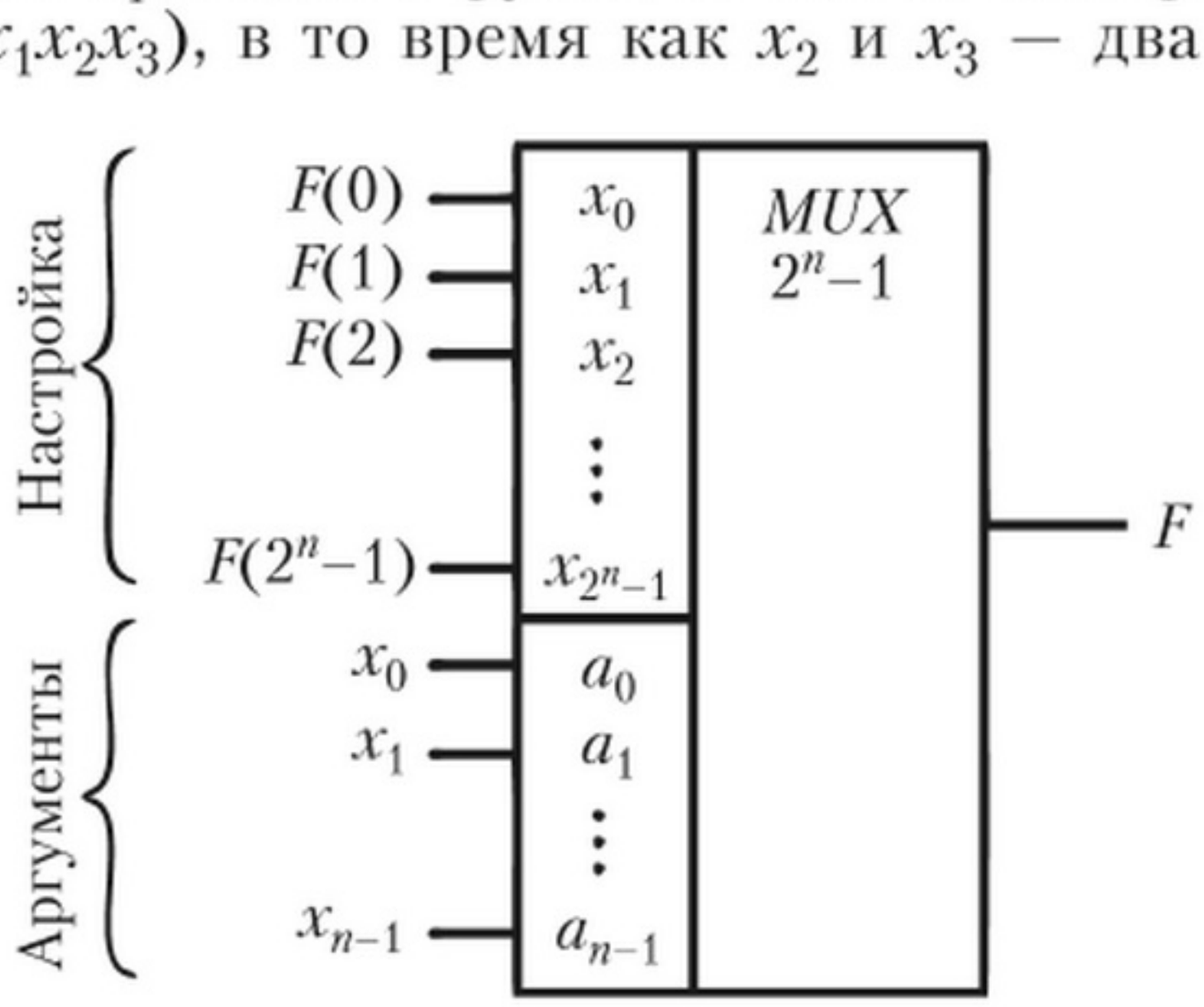


Рис. 3.19. **Реализация любой логической функции *n* аргументов**

значит, что если x_2 и x_3 равны единице, то значение функции F однозначно определяется значением x_1. Это дает возможность перехода к сокращенной таблице истинности функции F, которая в этом случае называется остаточной (табл. 3.8).

Таблица 3.8

x_3	x_2	$F_{ост}$
0	0	1
0	1	0
1	0	0
1	1	x_1

Таким образом, на входы настройки $F(0)$, $F(1)$, $F(2)$ мультиплексора, соответствующие наборам $x_3x_2 = 00$, 01 и 10, подаются логические «1» или «0» в соответствии с $F_{ост}$. На вход $F(3)$ подается аргумент x_1. Схема включения мультиплексора, реализующего функцию F, изображена на рис. 3.20. Переменная x_1 в этом случае переводится из аргументов на входы настройки.

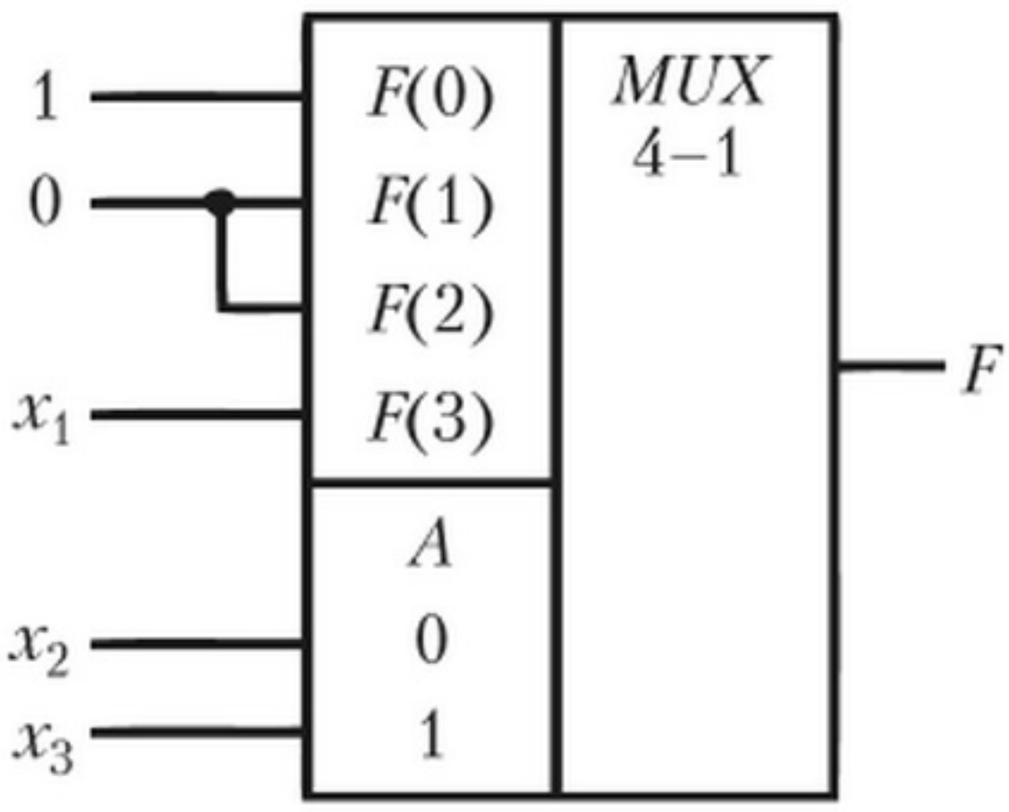


Рис. 3.20. **Реализация функции трех аргументов на мультиплексоре с двумя адресными входами**

Если использовать дополнительные элементарные логические схемы, то на основе мультиплексора можно реализовать функции более чем трех аргументов. При этом два аргумента будут подаваться на адресные входы, а остальные два — на входы настройки после соответствующей логической обработки.

Пример 3.2. Пусть на основе мультиплексора с двумя адресными входами необходимо реализовать функцию

$$F = x_1 x_2 \quad x_3 \overline{x_4}.$$

Решение. Данная функция равна единице, когда $x_2x_1 = 11$ или $x_4x_3 = 01$. Если на адресные входы подать аргументы x_1 и x_2, то на вход настройки $F(3)$, соответствующий набору $x_2x_1 = 11$, в соответствии с таблицей истинности $F_{ост}$ необходимо подать единицу (табл. 3.9).

Таблица 3.9

x_2	x_1	$F_{ост}$
0	0	$x_3\overline{x_4}$
0	1	$x_3\overline{x_4}$
1	0	$x_3\overline{x_4}$
1	1	1

Для всех остальных сочетаний x_1 и x_2 результат конъюнкции x_1x_2 равен нулю и функция F будет однозначно определяться конъюнкцией $x_3\overline{x}_4$. Поэтому на остальные входы настройки нужно подать этот сигнал, сформированный с помощью логических схем.

Если же в качестве адресных взять аргументы x_3 и x_4, то единицу нужно будет подавать на вход настройки $F(1)$, соответствующий набору $x_4x_3 = 01$ (табл. 3.10). На все остальные входы настройки нужно будет подавать результат конъюнкции x_1x_2 (рис. 3.21).

Таблица 3.10

x_4	x_3	$F_{ост}$
0	0	x_1x_2
0	1	1
1	0	x_1x_2
1	1	x_1x_2

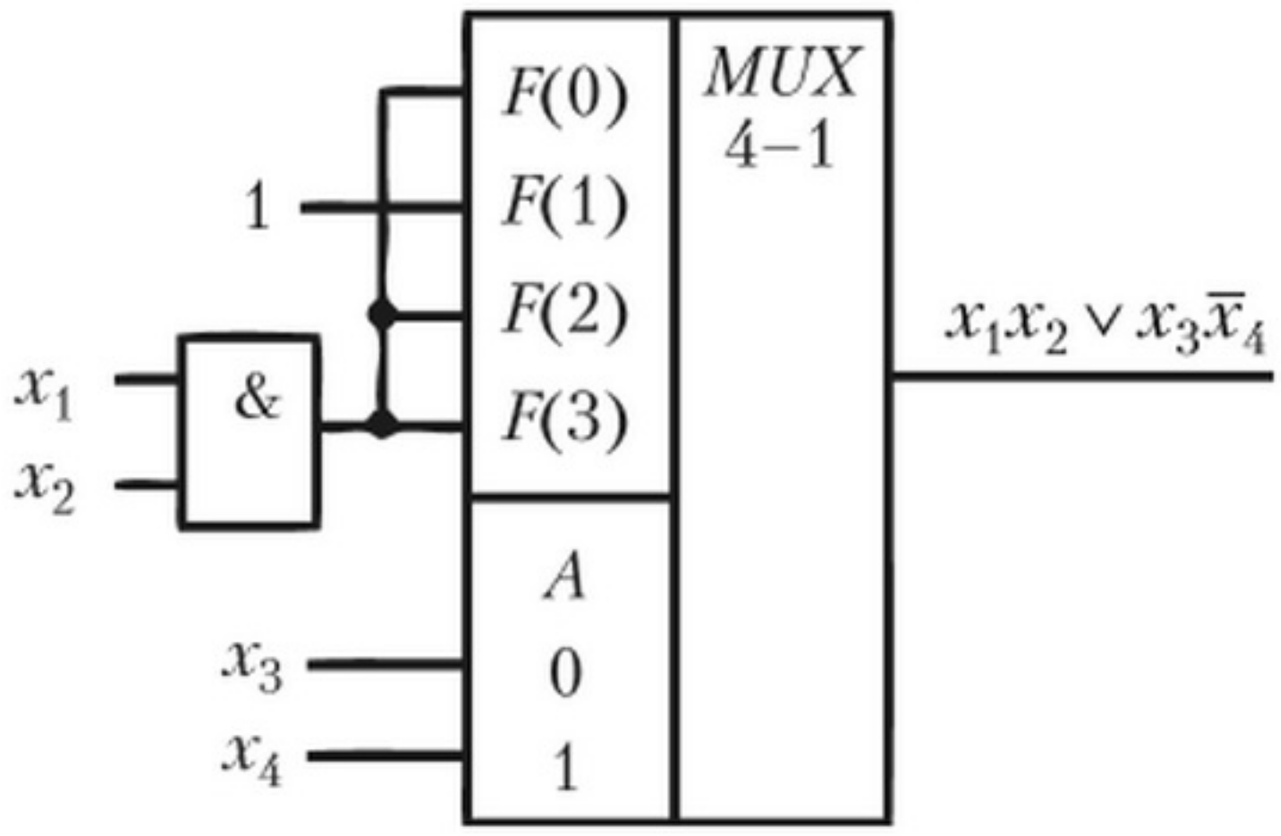


Рис. 3.21. **Реализация заданной функции *F***

Однако для реализации конъюнкции x_1x_2 нужна всего одна схема И на два входа, в то время как для конъюнкции $x_3\overline{x}_4$ требуется еще один инвертор. Поэтому в качестве адресных выбираем аргументы x_1 и x_2.

Еще более сложные логические функции можно реализовать на основе нескольких мультиплексоров. Именно они являются основой для выполнения логических функций в СБИС программируемой логики. Наборы мультиплексоров и вспомогательных логических схем в таких СБИС объединяются в универсальные логические блоки.

3.7. Сумматоры

3.7.1. Синтез одноразрядного сумматора

Как известно, все многообразие математических операций можно свести к единственной операции сложения прямых и обратных кодов чисел, сдвинутых влево или вправо на то или иное число разрядов. Поэтому одним из основных узлов вычислительных устройств является сумматор (*SM*). Любые сумматоры многоразрядных чисел могут быть построены на основе одноразрядного сумматора, который выполняет сложение трех одноразрядных двоичных чисел — двух слагаемых a и b с учетом переноса из более младшего разряда суммы. Результатом его работы является один разряд суммы и сигнал переноса в более старший разряд. В табл. 3.11 приведены все возможные варианты значений i-го разряда суммы s_i и переноса в старший разряд p_{i+1}, возникающие при сложении слагаемых a_i, b_i с учетом переноса p_i.

Таблица 3.11

Перенос предыдущего разряда p_i	Слагаемое		Сумма s_i	Перенос в старший разряд p_{i+1}
	a_i	b_i		
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

Составленная СовДНФ для сигнала переноса имеет вид:

$$p_{i+1} = \overline{p}_i a_i b_i \vee p_i \overline{a}_i b_i \vee p_i a_i \overline{b}_i \vee p_i a_i b_i.$$

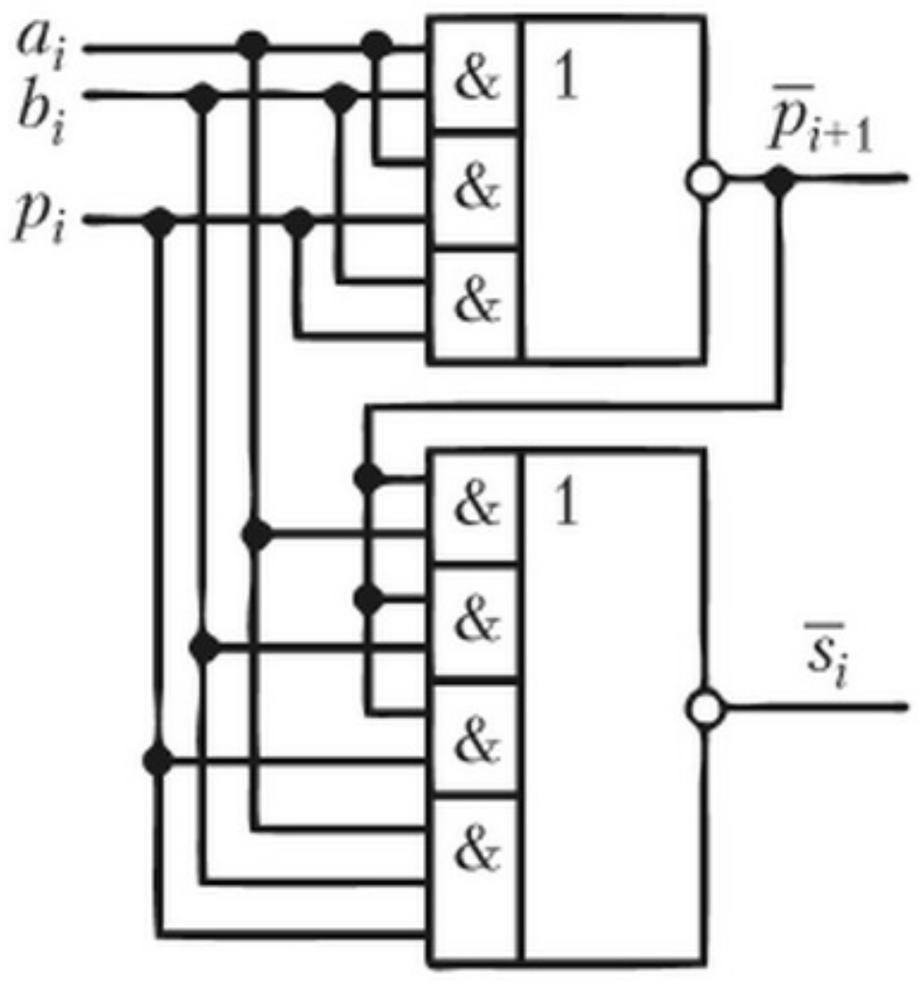
Произведя операции неполного склеивания первой конъюнкции с четвертой, второй с четвертой и третьей с четвертой, получим:

$$p_{i+1} = a_i b_i \vee p_i b_i \vee p_i a_i. \qquad (3.1)$$

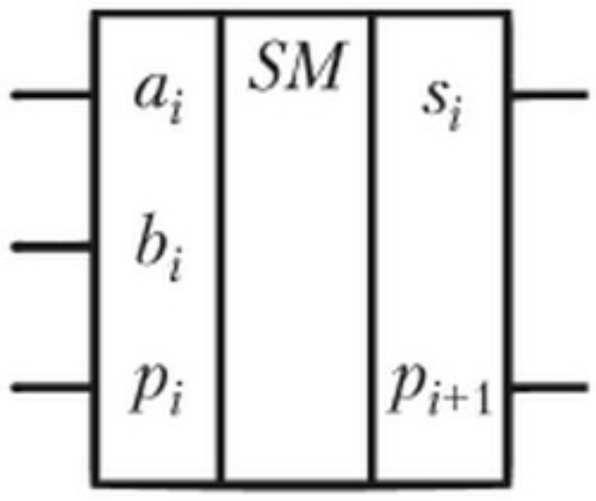
Для получения булевой функции сигнала суммы s_i можно воспользоваться обычным способом на основе СовДНФ, однако в интегральной схемотехнике для этого используют другой вид функции s_i, использующий полученное выражение для сигнала p_{i+1}. Надо обратить внимание на то, что во всех строках таблицы истинности, кроме первой и последней, $s_i = \overline{p}_{i+1}$. Чтобы использовать сигнал $\overline{p}_{i+1}$, необходимо получить из него выражение, удовлетворяющее первой и последней строкам. Выражение $a_i \vee b_i \vee p_i$ равно нулю только в первой строке таблицы, а во всех остальных равно единице. Значит, конъюнкция $\overline{p}_{i+1}$ с выражением $a_i \vee b_i \vee p_i$ сохранит все единичные значения $\overline{p}_{i+1}$, кроме первой строки. Чтобы добавить единицу в последней строке, выполним операцию ИЛИ полученного выражения с конъюнкцией $a_i\, b_i\, p_i$. Таким образом, выражение для сигнала суммы s_i примет окончательный вид:

$$s_i = \overline{p}_{i+1}(a_i \quad b_i \quad \vee p_i) \vee a_i b_i p_i.$$

Поскольку в интегральной схемотехнике базовый элемент обычно обеспечивает инвертирование, то схема сумматора имеет вид, приведенный на рис. 3.22, *а*. Чтобы получить прямые значения сигналов s_i и p_{i+1}, необходимы еще два инвертора. Условное обозначение одноразрядного сумматора приведено на рис. 3.22, *б*.



а



б

Рис. 3.22. **Сумматор:**
а — функциональная схема; *б* — условное обозначение

3.7.2. Последовательный многоразрядный сумматор

Используя одноразрядный сумматор, можно построить суммирующее устройство для сложения многоразрядных двоичных чисел. Различают многоразрядные последовательные и параллельные сумматоры.

Последовательный сумматор состоит из одноразрядного сумматора, на входы которого из сдвигающих регистров, хранящих слагаемые A и B, подаются по тактам разряд за разрядом коды этих чисел, начиная с младшего разряда (рис. 3.23). Если время элемента задержки равно одному такту (например, использован D-триггер), то сигнал переноса от предыдущего разряда, сформированный в предыдущем такте, поступит на нижний вход сумматора только в следующем такте, когда на входы a и b будут поданы значения следующего разряда слагаемых. В результате на выходе s разряд за разрядом будет формироваться дво-

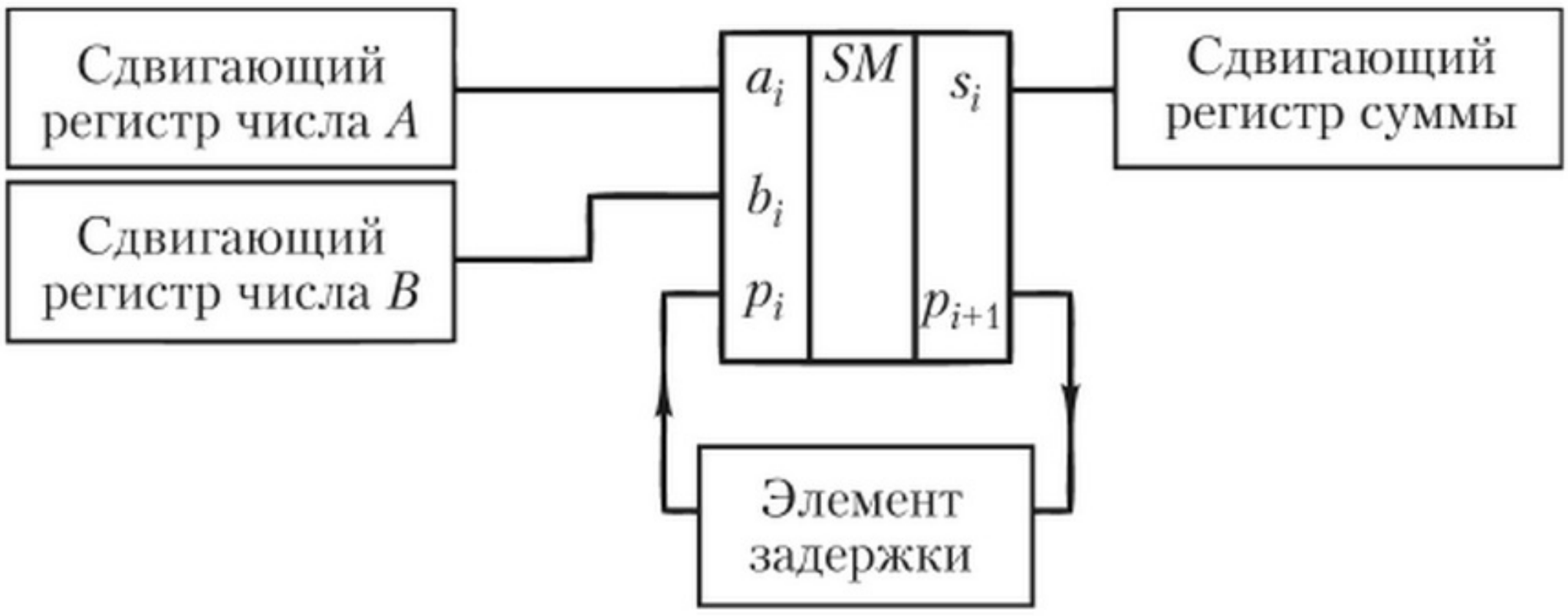


Рис. 3.23. **Последовательный многоразрядный сумматор**

ичное число, равное сумме чисел A и B, которое будет восприниматься сдвигающим регистром суммы.

3.7.3. Параллельные сумматоры

Недостатком последовательного сумматора является то, что выполнение операции сложения растягивается на множество тактов, которое тем больше, чем больше разрядность чисел. Значительно меньшее время выполнения операции имеет *параллельный сумматор*. В этом устройстве операция сложения производится одновременно во всех разрядах чисел A и B, поступающих в параллельном коде. Для этого схему составляют из n одноразрядных сумматоров, соединяя выход переноса i-го разряда со входом переноса соседнего $(i + 1)$-го разряда (рис. 3.24). Такой сумматор называют сумматором с последовательным переносом. После того как сформируется результат на выходах всех одноразрядных сумматоров, он запоминается в параллельном регистре.

Для того чтобы на выходах одноразрядных сумматоров сформировался результат, необходимо, чтобы на входах присутствовали все три сигнала — a_i, b_i и p_i. Но сигнал переноса из разряда в разряд формируется предыдущими одноразрядными сумматорами, причем каждый последующий сумматор вынужден «ждать», пока не сформируется результат во всех предыдущих. В крайнем случае возможна ситуация, когда сигнал переноса будет последовательно передаваться через всю цепочку от самого младшего разряда до самого старшего, например, при сложении чисел $A = 11...11$ и $B = 00...01$. Поэтому время формирования

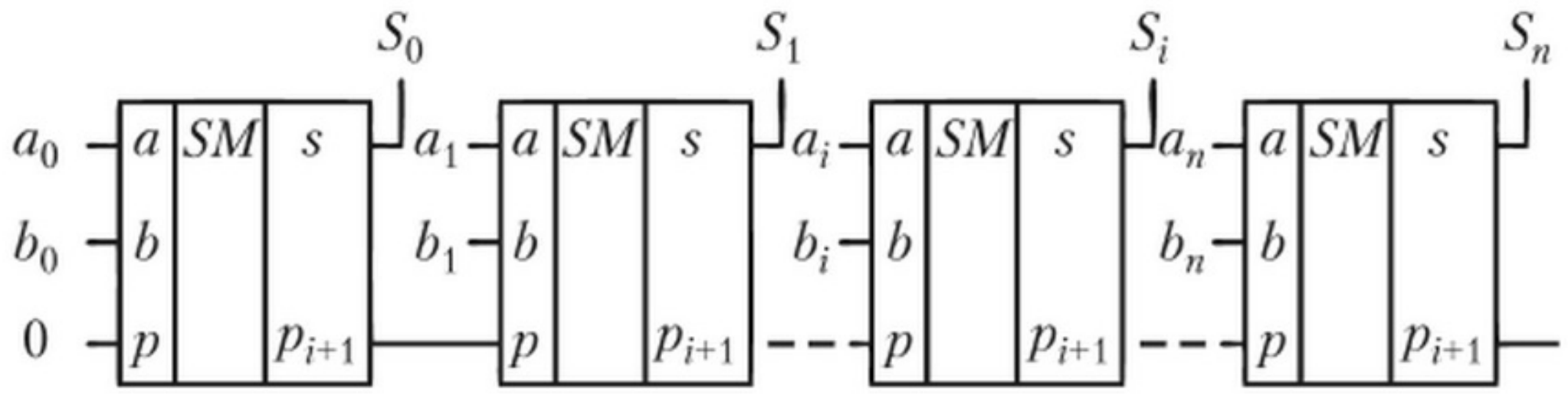


Рис. 3.24. **Параллельный сумматор с последовательным переносом**

результата в наихудшем случае включает в себя временные задержки, вносимые всеми одноразрядными сумматорами.

Исключить длительность распространения переноса позволяет *сумматор с параллельным переносом*. Идея заключается в том, чтобы сигналы переноса для всех разрядов формировались чисто логически на основе слагаемых A и B. Рассмотрим, как это можно сделать.

Сигнал переноса в 1-й разряд $p_1 = a_0 b_0$. Сигнал переноса во второй разряд можно определить по формуле (3.1):

$$p_2 = a_1 b_1 \vee a_1 p_1 \vee b_1 p_1 = a_1 b_1 \vee p_1 (a_1 \vee b_1) = a_1 b_1 \vee a_0 b_0 (a_1 \vee b_1).$$

Как видно, сигнал переноса p_2 может быть получен по значениям предыдущих разрядов чисел A и B без сигнала переноса p_1, формируемого сумматором. Подставляя в формулу (3.1) выражение для сигнала переноса p_2, можно получить логическое выражение для сигнала переноса третьего разряда, в котором также будут значения только предыдущих разрядов чисел A и B, и т.д. Таким образом, сигналы переноса для любого разряда могут быть сформированы чисто логическим путем по значениям слагаемых. Значит, нет необходимости ожидать, пока будут сформированы сигналы переноса одноразрядными сумматорами. Очевидно, однако, и другое. Чем больше разрядность чисел, тем выше сложность булевых функций сигналов переноса, особенно для самого старшего разряда. Поэтому сумматоры с параллельным переносом делают для чисел небольшой разрядности — не более восьми.

Структура сумматора с параллельным переносом приведена на рис. 3.25. Сигналы переноса формируются специальными логическими схемами ускоренного переноса CR (англ. *carry* — перенос), вычисляющими соответствующие булевы функции.

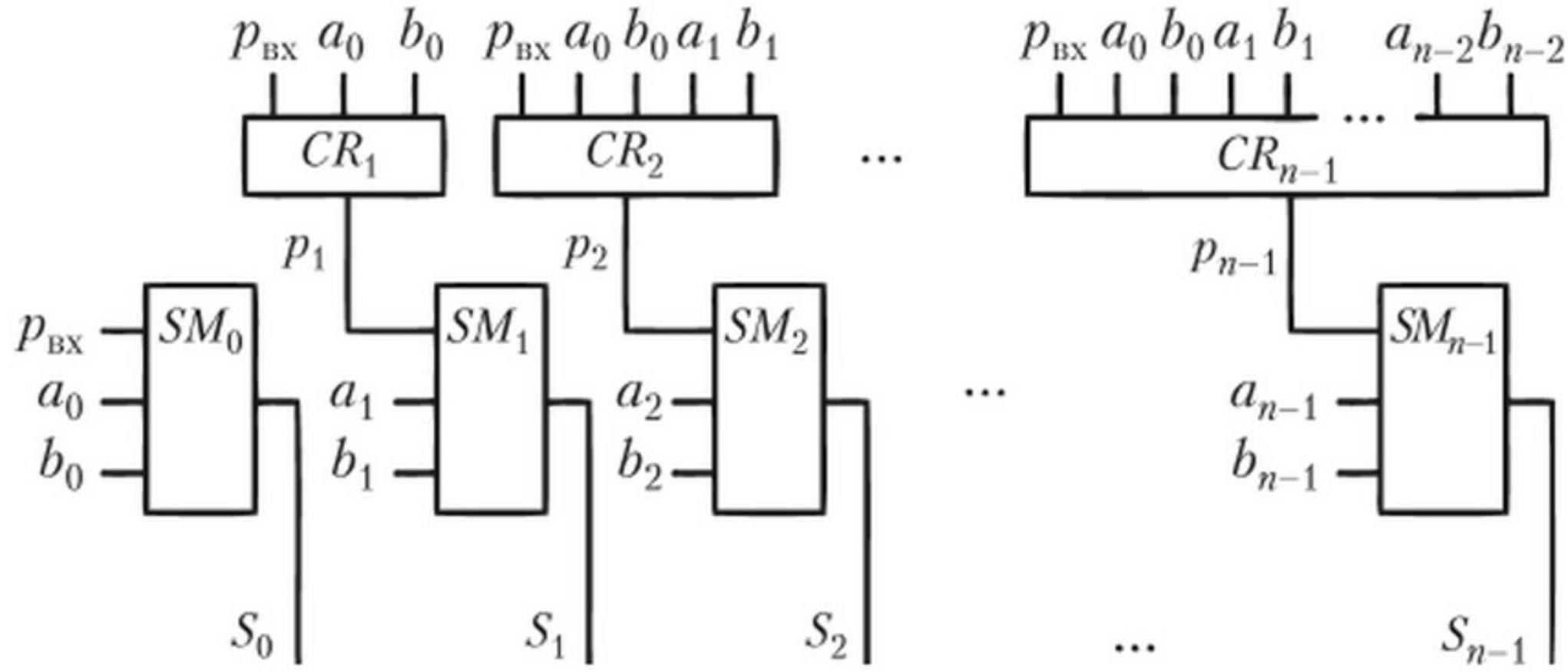


Рис. 3.25. **Сумматор с параллельным переносом**

Для чисел большой разрядности применяют *сумматоры с групповым переносом* (рис. 3.26). Схема такого сумматора разбивается на l групп разрядности m: например, четыре группы по восемь разрядов для сложения 32-разрядных чисел. (Обозначение $\underset{\diagup}{m}$ эквивалентно изображению группы из m проводников и используется для указания разрядности цифровых устройств и линий связи между ними.) Каждая группа представляет собой параллельный сумматор, подобный изображенному на рис. 3.25. На вход $p_{вх}$ поступает сигнал переноса от старшего разряда предыдущей группы к младшему разряду последующей группы. Этот сигнал формируется блоком переноса, анализирующим m разрядов слагаемых A и B «своей» группы, «не дожидаясь», когда в ней произойдет сложение всех разрядов. Блоки переноса (БП) разных групп включены последовательно и образуют тракт передачи переноса.

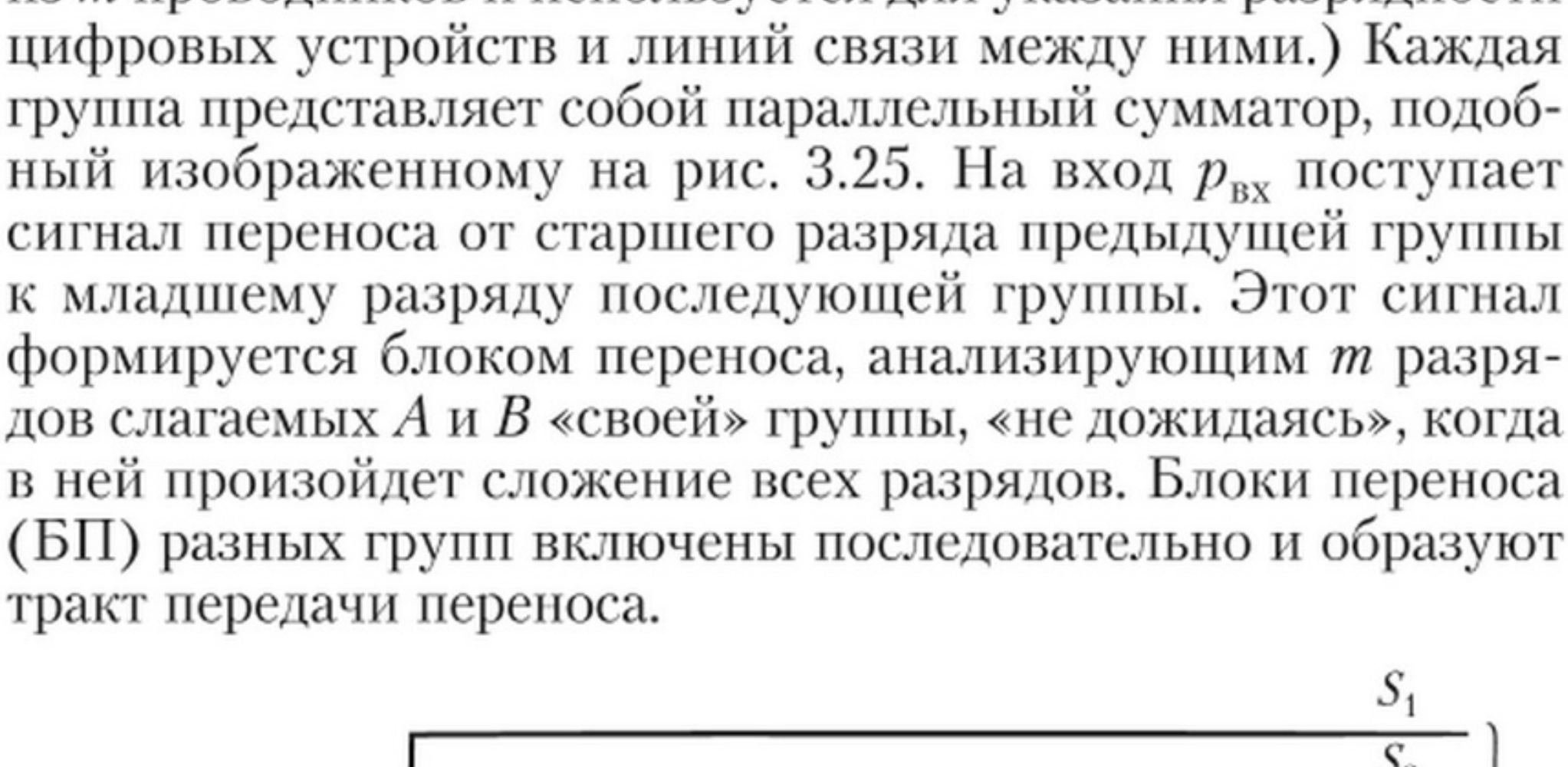


Рис. 3.26. **Групповой сумматор с цепным переносом**

Содержание

Эта структура сумматора с групповым переносом аналогична структуре сумматора с последовательным переносом, где вместо одноразрядных сумматоров включаются групповые. Такая структура получила название группового сумматора с цепным переносом.

Кроме структуры группового сумматора с цепным переносом возможна структура с параллельными межгрупповыми переносами. Структура этого сумматора аналогична структуре сумматора с параллельным переносом, в котором роль одноразрядных сумматоров играют группы. Аппаратная сложность сумматоров с параллельными межгрупповыми переносами выше, чем сложность предыдущего варианта, но при больших разрядностях они дают преимущества по быстродействию.

3.8. Арифметико-логические устройства и матричные умножители

Арифметико-логические устройства (АЛУ) выполняют арифметические и логические операции над двумя операндами. Основой АЛУ служит сумматор, схема которого дополнена логикой, расширяющей функциональные возможности АЛУ и обеспечивающей перестройку с одной операции на другую. Обычно АЛУ четырехразрядны и для наращивания разрядности объединяются.

Рассмотрим типовое АЛУ (рис. 3.27). В ИС имеются входы чисел A и B, входы выбора операций S, вход переноса $\overline{C}_i$ из младшего разряда и вход M, сигнал которого задает тип выполняемых операций: логические ($M = 1$) или арифметико-логические ($M = 0$). Результат операции вырабатывается на выходах F. Выходы G и H используются для организации параллельных переносов при наращивании разрядности обрабатываемых слов. Сигнал $\overline{C}_0$ — выходной перенос в старший разряд, а выход «$A = B$» — выход, равный «1» при условии $A = B$.

Перечень выполняемых АЛУ операций дан в табл. 3.12. Для краткости двоичные числа $s_3s_2s_1s_0$ представлены их десятичными эквивалентами. Все логические операции выполняются поразрядно, т.е. между одноименными (i-ми) разрядами чисел A и B. Межразрядные переносы возникают только в арифметических операциях. В арифметико-

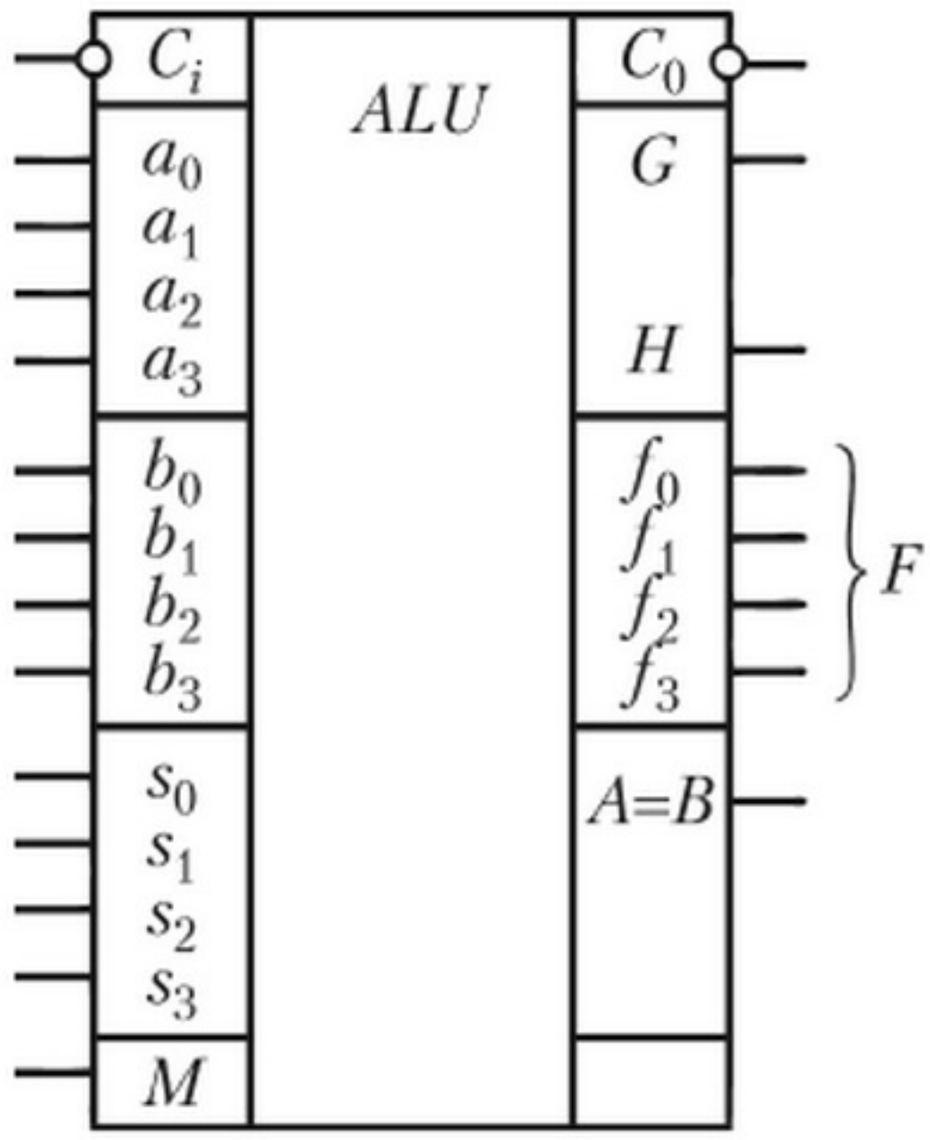


Рис. 3.27. **Условное обозначение АЛУ**

логических операциях сначала выполняются логические операции, а затем полученные коды складываются арифметически. Знак $\oplus$ обозначает логическую операцию «сумма по модулю два».

Таблица 3.12

S	Логические функции ($M = 1$)	Арифметико-логические функции ($M = 0$)
0	$\overline{A}$	$A + C_i$
1	$\overline{A \vee B}$	$A \vee B + C_i$
2	$\overline{A}B$	$A \vee \overline{B} + C_i$
3	0	$1 + C_i$
4	$\overline{AB}$	$A + A\overline{B} + C_i$
5	$\overline{B}$	$A \vee B + A\overline{B} + C_i$
6	$A \oplus B$	$A + \overline{B} + C_i$
7	$A\overline{B}$	$A\overline{B} + 1 + C_i$
8	$\overline{A} \vee B$	$A + AB + C_i$

Окончание табл. 3.12

S	Логические функции ($M = 1$)	Арифметико-логические функции ($M = 0$)
9	$\overline{A \oplus B}$	$A + B + C_i$
10	B	$A \vee \overline{B} + AB + C_i$
11	AB	$AB + 1 + C_i$
12	1	$A + A + C_i$
13	$A \vee \overline{B}$	$A \vee B + A + C_i$
14	$A \vee B$	$A \vee \overline{B} + A + C_i$
15	A	$A + 1 + C_i$

При операциях над словами большой разрядности АЛУ соединяются друг с другом с организацией последовательных и параллельных переносов. Соединение микросхем АЛУ в первом случае аналогично соединению сумматоров в схеме с последовательным переносом: выход $\overline{C}_0$ предыдущей микросхемы соединяется с входом $\overline{C}_i$ последующей. Если используется организация с параллельными переносами, то совместно с АЛУ применяются специальные микросхемы — блоки ускоренного переноса.

Один блок ускоренного переноса обеспечивает анализ специальных вспомогательных сигналов G и H, формируемых всеми микросхемами АЛУ, и формирование для них сигналов переноса $\overline{C}_i$. Сигналы переноса в старший разряд $\overline{C}_0$ при этом не используются.

Матричные умножители выполняют операцию умножения двоичных чисел не традиционным путем последовательных сдвигов и сложений, а параллельно. Схема выполнения операции умножения соответствует обычному «умножению столбиком». Но поскольку в операции используют двоичные числа, то арифметическое умножение цифры множимого на одну из цифр множителя заменяется конъюнкцией. Для трехразрядных чисел A и B схема умножения имеет следующий вид:

$$
\begin{array}{cccccc}
 & & a_2 & a_1 & a_0 & \\
 & & \times & & & \\
 & & b_2 & b_1 & b_0 & \\
\hline
 & & a_2b_0 & a_1b_0 & a_0b_0 & \\
+ & a_2b_1 & a_1b_1 & a_0b_1 & & \\
 & a_2b_2 & a_1b_2 & a_0b_2 & & \\
\hline
P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$

Все конъюнкции $a_i\, b_i$ вырабатываются параллельно схемами И, а для операций сложения используются одноразрядные сумматоры. Матричные умножители могут быть просто множительными блоками либо множительно-суммирующими. Последние обеспечивают удобство наращивания разрядности перемножаемых двоичных чисел.

В виде ИС средней степени интеграции реализуются умножители малой разрядности — не более 4×4. Умножители большей разрядности — до 16×16 выполняются уже в виде БИС.

3.9. Триггеры

Триггеры являются простейшими, элементарными конечными (цифровыми) автоматами, обладающими памятью. В более сложных дискретных устройствах они служат для хранения одного разряда двоичных чисел (одного бита информации). Практически все последовательностные устройства выполняются на основе триггеров.

Триггер, как конечный автомат, характеризуется следующими свойствами:

1) возможное число внутренних состояний — два (единица или нуль), что соответствует одной внутренней переменной, обозначаемой для триггеров обычно буквой Q;

2) число выходных переменных y — одно; значение переменной y совпадает с значением Q, т.е. $y^{t+1} = Q^{t+1}$;

3) число входных переменных x зависит от типа триггера.

Наряду с выходом Q, называемым *прямым*, триггер имеет инверсный выход $\overline{Q}$. Состояние триггера определяется значением прямого выхода Q. Говорят, что триггер находится в единичном состоянии, если $Q = 1$, и в нулевом,

если $Q = 0$. Изменение состояния триггеров под действием входных сигналов происходит скачкообразно.

Само название «триггер» происходит от английского слова *trigger* — спусковой крючок (курок) огнестрельного оружия, поэтому к триггерам часто применяют выражение «взводить», т.е. устанавливать его в единичное состояние, и «спускать, сбрасывать» при переводе его в нулевое состояние.

Разработано большое число типов триггеров, которые по функциональному признаку можно разделить на четыре основных типа:

- триггеры с двумя установочными входами — *RS*-триггеры. Вход *R* сбрасывает триггер в нулевое состояние (*reset* — сбрасывать), а вход *S* — устанавливает в единицу (*set* — устанавливать);
- триггеры задержки с одним входом — *D*-триггеры (*delay* — задержка);
- триггеры с одним счетным входом — *T*-триггеры;
- универсальные триггеры с несколькими входами.

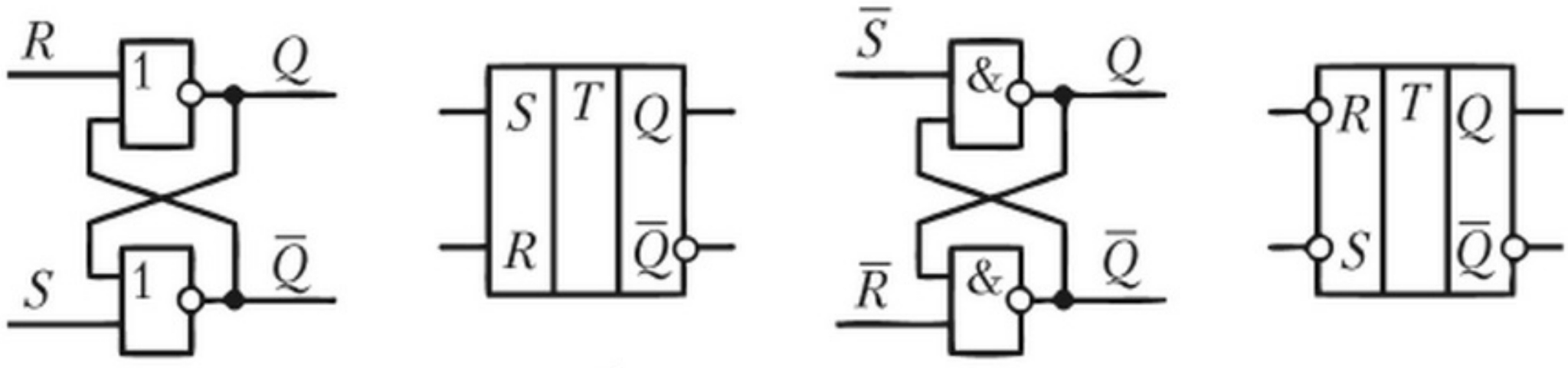
Как и любые конечные автоматы, триггеры могут быть асинхронными и синхронными (тактируемыми). В *асинхронных* изменение состояния происходит непосредственно с приходом входного сигнала, а в *синхронных* — лишь при подаче синхронизирующего (тактового) сигнала в соответствии со значениями сигналов на входах.

При этом различают триггеры со статическими и динамическими входами. Входы, управляемые потенциалами (уровнями напряжения), называют *статическими* (включая и сигнал синхронизации), а управляемые перепадами потенциалов (фронтами импульсов напряжения) — *динамическими*.

3.9.1. Триггер с установочными входами (*RS*-триггер)

Асинхронный *RS*-триггер. Асинхронный *RS*-триггер — это самый элементарный автомат с памятью. Если на триггер не подавать никаких внешних управляющих сигналов, он обеспечивает хранение своего состояния Q, равного единице или нулю, сколь угодно долгое время. Триггер строится на основе двух элементов ИЛИ-НЕ с двумя входами каждый (рис. 3.28, *а*, *б*).

Схема имеет два входа и два выхода. Входными для схемы являются сигналы установки триггера в единицу — *S* и сброса триггера в нуль — *R*. Выходами схемы являются



а *б* *в* *г*

Рис. 3.28. ***RS*-триггер на элементах ИЛИ-НЕ (*а*) и его условное обозначение (*б*); *RS*-триггер на элементах И-НЕ с инверсными входами (*в*) и его условное обозначение (*г*)**

прямой выход состояния триггера Q и инверсный выход $\overline{Q}$. Входные сигналы подаются на один из входов схем ИЛИ-НЕ. На другие входы схем подаются сигналы с выходов Q и $\overline{Q}$. Запоминающие свойства триггера обеспечивает перекрестное соединение выходов Q и $\overline{Q}$ и со вторыми входами элементов ИЛИ-НЕ. Предположим, что на входах R и S триггера поддерживались сигналы лог. 0 и в некоторый момент на вход S триггера была подана лог. 1. Тогда на выходе нижнего элемента ИЛИ-НЕ, соответствующего выходу $\overline{Q}$, будет лог. 0, который по перекрестной связи будет подан на второй вход верхнего элемента ИЛИ-НЕ. Так как на обоих входах верхнего элемента ИЛИ-НЕ будут лог. 0, на его выходе Q установится лог. 1 и триггер установится в единицу. Эта лог. 1 по перекрестной связи будет подана на второй вход нижней схемы ИЛИ-НЕ, поэтому даже если на входе S будет лог. 0, она принудительно будет поддерживать на выходе нижнего элемента лог. 0, а тот, в свою очередь, лог. 1 на выходе верхнего элемента. Таким образом, триггер «запомнит» единичное состояние. Если теперь на вход R подать лог. 1, верхний элемент сформирует на своем выходе лог. 0, который установит на выходе нижнего элемента лог. 1. Теперь выходы верхнего и нижнего элементов ИЛИ-НЕ будут взаимно поддерживать лог. 0 на выходе Q и лог. 1 на выходе $\overline{Q}$. Таким образом, триггер «запомнит» нулевое состояние. Если на входы R и S одновременно подать лог. 1, состояние на выходе триггера будет неопределенным, поэтому эта комбинация является для асинхронного *RS*-триггера запрещенной.

Аналогичная схема асинхронного *RS*-триггера может быть синтезирована на элементах И-НЕ (рис. 3.28, *в*). Осо-

бенностью схемы являются инверсные входы $\overline{S}$ и $\overline{R}$. Это означает, что для перевода триггера, например, в единичное состояние нужно на вход $\overline{S}$ подать не единичный, а нулевой сигнал, а в состоянии покоя на входах $\overline{S}$ и $\overline{R}$ лог. 1. В остальном работа схемы аналогична предыдущей. Условное изображение *RS*-триггера с инверсными входами показано на рис. 3.28, *г*.

Асинхронный *RS*-триггер является базовым элементом памяти, на основе которого в интегральной схемотехнике строятся многочисленные ИС триггерных и других устройств, обладающих памятью. При этом часто входы *R* и *S* выводятся на контакты микросхемы наряду с основными управляющими входами. В этом случае входы *S* и *R* называются входами асинхронной установки в единицу и в нуль.

Синхронный *RS*-триггер. Как указывалось в параграфе 3.2, из-за задержки переключения логических элементов в цифровых автоматах возможны режимы *состязаний*, в результате которых выходной сигнал на некоторое время принимает ложные значения. Устранить такие ошибки позволяет способ временно́го стробирования. Вырабатываемый устройством сигнал в этом случае передается на последующие устройства не непрерывно, а только в такой отрезок времени, когда искажение правильных значений выходного сигнала за счет состязаний заведомо исключено.

Временно́е стробирование обеспечивается синхронизирующими импульсами, поэтому синхронный *RS*-триггер, кроме информационных входов *R* и *S*, имеет вход *C*, на который подается синхронизирующий сигнал (рис. 3.29).

Синхронизация осуществляется включением на каждом входе асинхронного *RS*-триггера дополнительного элемента И-НЕ. Теперь сигналы *S* и *R* смогут пройти на входы

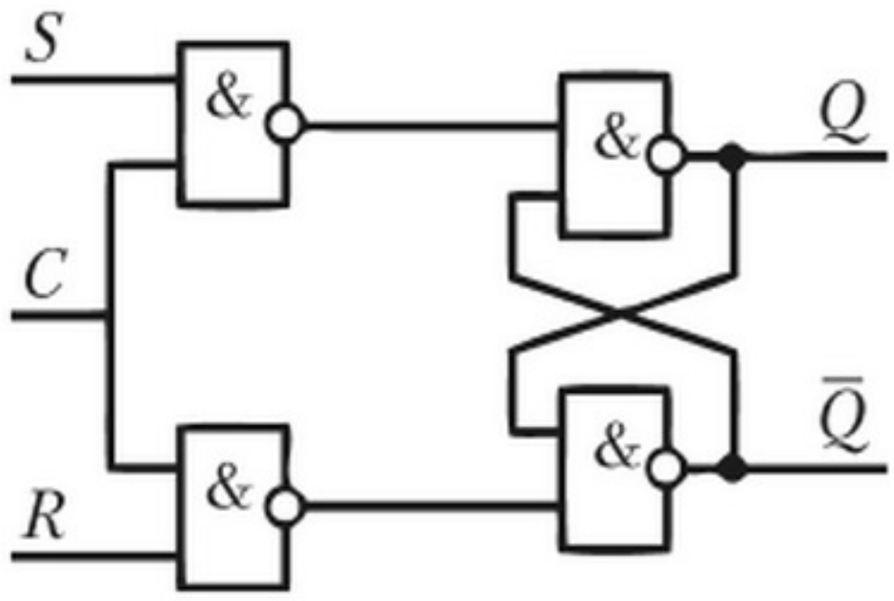


Рис. 3.29. **Синхронный *RS*-триггер на элементах И-НЕ**

Содержание

триггера только при наличии синхронизирующего сигнала *C*. Заметим, что за счет дополнительного инвертирования во входных элементах И-НЕ сигналы *R* и *S* становятся не инверсными, как это было в асинхронном триггере, а прямыми.

3.9.2. Триггер задержки (*D*-триггер)

Триггер задержки имеет единственный входной сигнал, обозначенный буквой *D*. Значение сигнала на выходе *Q* такого триггера в такт $(t + 1)$ равно значению сигнала на входе *D* в предыдущем такте *t*. Таким образом, входной сигнал передается на выход с задержкой на один такт, т.е. $Q^{t+1} = D^t$.

Асинхронный *D*-триггер не применяется, так как его выход просто повторяет входной сигнал после окончания режима состязаний. Синхронный же *D*-триггер функционирует в соответствии с табл. 3.13, из которой видно, что при отсутствии синхроимпульсов $(C = 0)$ состояние триггера остается неизменным. При условии $C = 1$ триггер передает на выход сигнал, поступивший на его вход в предыдущем такте.

Для синтеза синхронного *D*-триггера воспользуемся асинхронным *RS*-триггером. Как видно из таблицы, сигнал на входе *D* должен воздействовать на триггер только в том случае, когда имеется единичный уровень сигнала *C*. Следовательно, как и в случае синхронного *RS*-триггера, на входе *D*-триггера должны быть двухвходовые элементы И, пропускающие сигнал с входа *D*, если $C = 1$.

Таблица 3.13

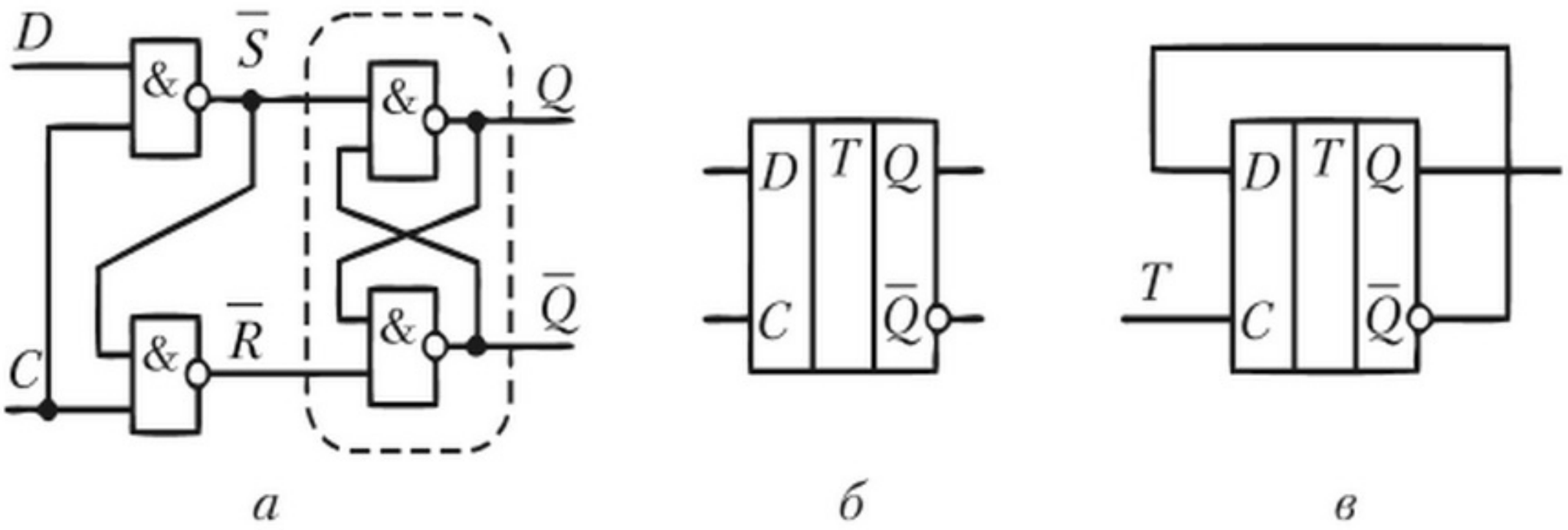
Такт t			Q^{t+1}	Такт t			Q^{t+1}
C^t	D^t	Q^t		C^t	D^t	Q^t	
0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1

Теперь обратимся к правой части таблицы. Переход триггера в состояние единицы должен происходить, когда на входах воздействуют сигналы $C = 1$ и $D = 1$. Значит, при таком сочетании сигналов на входе $\overline{S}$ асинхронного

RS-триггера должен быть нуль, т.е. $\overline{S} = \overline{CD}$. Переход триггера в состояние нуля должен происходить, когда на входах имеется комбинация $C = 1$, $D = 0$. Значит, на входе $\overline{R}$ асинхронного *RS*-триггера при таком сочетании сигналов также должен быть нуль, т.е. $\overline{R} = \overline{C\overline{D}}$. Поскольку сигнал на входе $\overline{S}$ формируется в момент, когда $C = 1$, и соответствует инверсному значению D, его можно использовать для формирования сигнала $\overline{R}$, т.е. $\overline{R} = \overline{C\overline{S}}$. Это же рассуждение можно получить и чисто формально с помощью алгебры логики:

$$\overline{R} = \overline{C\overline{S}} = \overline{C\overline{CD}} = \overline{C(\overline{C} \vee \overline{D})} = \overline{C\overline{C} \vee C\overline{D}} = \overline{C\overline{D}}.$$

Следовательно, *D*-триггер может быть синтезирован на основе асинхронного *RS*-триггера (рис. 3.30, *а*, *б*). Пунктиром обведена схема асинхронного *RS*-триггера, верхняя схема И-НЕ формирует сигнал на входе $\overline{S}$ асинхронного *RS*-триггера, а нижняя — на входе $\overline{R}$.



а *б* *в*

Рис. 3.30. **Синхронный *D*-триггер (*а*), его условное обозначение (*б*) и его использование в качестве *T*-триггера (*в*)**

3.9.3. *T*-триггер

Булеву функцию, отражающую работу ***T*-триггера**, содержательно можно представить так: триггер инвертирует свое предыдущее состояние при подаче по его вход *T* единичного сигнала, т.е. $Q^{t+1} = Q^t$ при $T = 1$. Соответствующая таблица переключений *T*-триггера представлена в табл. 3.14. Вход *T* называют *счетным входом* триггера. При подаче на триггер первого единичного сигнала он устанавливается в единицу, при подаче второго — сбрасыва-

ется в нуль, третьего — опять в единицу и т.д. Таким образом, изменение выходного сигнала происходит с частотой вдвое меньшей, чем входного. Это позволяет использовать *T*-триггер для синтеза двоичных счетчиков, в которых каждый триггер соответствует одному двоичному разряду.

Таблица 3.14

Такт t		Q^{t+1}
T^t	Q^t	
0	0	0
1	0	1
0	1	1
1	1	0

T-триггер может быть синтезирован на логических элементах И-НЕ или ИЛИ-НЕ, однако чаще всего для этого используют имеющийся в составе серий интегральных микросхем *D*-триггер. Превращение *D*-триггера в *T*-триггер осуществляется с помощью соответствующего соединения выводов микросхемы.

Так, если в синхронном *D*-триггере вход *D* соединить с инверсным выходом $\overline{Q}$, то он превращается в *T*-триггер. Действительно, если просмотреть в таблице переключений *D*-триггера строки, в которых состояние триггера *Q* меняется на противоположное (см. строки 2 и 3 правой части табл. 3.13), увидим, что значение сигнала на входе *D* противоположно значению *Q*. Поскольку сигнал на выходе $\overline{Q}$ всегда противоположен состоянию *Q*, то, подавая счетные сигналы *T* вместо синхроимпульсов на вход *C* из *D*-триггера получаем *T*-триггер (рис. 3.30, *в*).

3.9.4. *JK*-триггер

***JK*-триггер** имеет два входа *J* и *K*. Его работа описывается таблицей переключений (табл. 3.15), из которой видно, что при комбинации сигналов $J = 1$ и $K = 1$ (третий и седьмой наборы) триггер инвертирует свое предыдущее состояние, т.е. при конъюнкции $JK = 1$ $Q^{t+1} = \overline{Q^t}$. В остальных случаях *JK*-триггер аналогичен *RS*-триггеру. При этом вход *J* эквивалентен входу *S*, а вход *K* — входу *R*.

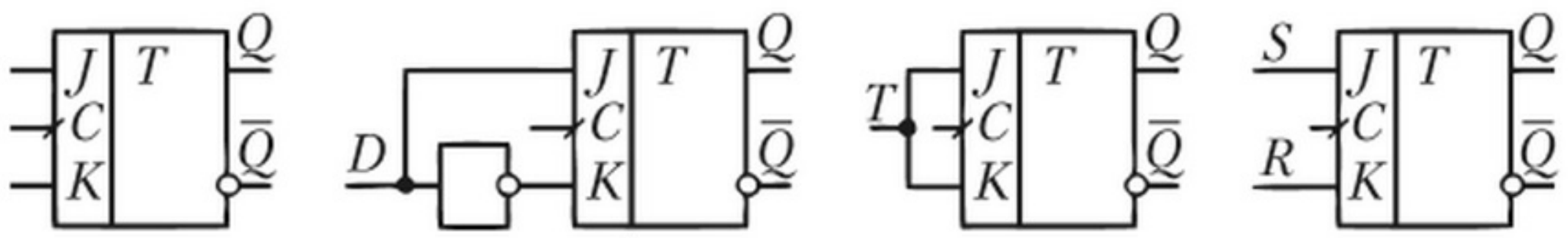
Таблица 3.15

Такт t				Q^{t+1}	Такт t				Q^{t+1}
Номера наборов	K^t	J^t	Q^t		Номера наборов	K^t	J^t	Q^t	
0	0	0	0	0	4	0	0	1	1
1	0	1	0	1	5	0	1	1	1
2	1	0	0	0	6	1	0	1	0
3	1	1	0	1	7	1	1	1	0

В интегральной схемотехнике распространены синхронные *JK*-триггеры с динамическим входом синхроимпульсов *C* (рис. 3.31, *а*). Эти триггеры путем несложной коммутации (соединения) внешних выводов можно превратить в схемы, выполняющие функции *RS*-, *D*- и *T*-триггера. Поскольку *T*-триггер меняет свое состояние на противоположное при подаче каждого единичного сигнала на счетный вход *T*, используем аналогичное свойство *JK*-триггера инвертировать свое состояние при одновременной подаче единичных сигналов на входы *J* и *K*. Таким образом, для превращения *JK*-триггера в *T*-триггер достаточно соединить входы *J* и *K* и подавать на них счетные сигналы *T* (рис. 3.31, *в*).

Чтобы из *JK*-триггера получить *D*-триггер, достаточно на вход *J* подавать прямое значение сигнала *D*, а на вход *K* — инверсное, т.е. необходим лишь один инвертор. При этом единичное значение сигнала *D* будет взводить триггер через вход *J*, а нулевое — сбрасывать через вход *K* (рис. 3.31, *б*).

Наконец, если на вход *J* подавать сигналы *S*, а на вход *K* — сигналы *R*, то *JK*-триггер будет выполнять функции *RS*-триггера (рис. 3.31, *г*).



а *б* *в* *г*

Рис. 3.31. **Условное обозначение динамического *JK*-триггера (*а*) и его использование в качестве *D*-триггера (*б*), *T*-триггера (*в*) и *RS*-триггера (*г*)**

Содержание

Рассмотренные синхронные триггеры являются одноступенчатыми, т.е. содержат один элемент памяти. В их обозначении используется одна буква *T* (см. рис. 3.31). Устойчивая работа таких триггеров в схемах с передачей информации между триггерами возможна только при условии, что перевод триггера в новое состояние осуществляется после завершения передачи информации о прежнем его состоянии в следующий за ним триггер. Выполнение этого условия обеспечивается в *двухступенчатых* триггерах, состоящих из двух одноступенчатых триггеров. Сначала информация записывается в первый триггер (обычно по переднему фронту сигнала *C*), а затем переписывается во второй (по заднему фронту). Такой принцип построения триггеров иногда называют *MS*-принципом (*master-slave* — хозяин-раб). Все рассмотренные типы триггеров могут быть построены по *MS*-принципу (на схемах их обозначают *TT*).

Триггеры выполняются в виде ИС средней степени интеграции. Обычно они имеют дополнительную логику во входных цепях, благодаря чему обычный триггер может иметь уже не один, а несколько входов. Часть этих входов может использоваться как информационные, а часть — как управляющие (разрешающие или запрещающие запись). ИС триггеров могут иметь входы асинхронной установки *R*, *S*, которые воздействуют на элемент памяти триггера непосредственно, минуя входные цепи, и являются доминирующими по отношению к другим входам. В одном корпусе ИС размещается до четырех триггеров, число которых ограничивается количеством выводов микросхемы.

3.10. Счетчики

Счетчиком называется узел для подсчета числа входных сигналов и хранения двоичного кода числа подсчитанных сигналов.

Счетчики — это конечные автоматы, внутреннее состояние которых определяется только количеством сигналов «1», пришедших на выход. Сигналы «0» не изменяют их внутреннего состояния.

Счетчики делятся на *суммирующие*, *вычитающие*, *реверсивные*, т.е. позволяющие как прибавить, так и вычесть очередную пришедшую на вход единицу.

Модулем счета (или коэффициентом пересчета) $K_{сч}$ называют количество поступивших на вход единиц, которое возвращает счетчик в исходное состояние.

Элементарными конечными автоматами в счетчиках являются триггеры. Количество триггеров, необходимое для реализации счетчика или пересчетной схемы, равно:

$$m = \log_2 K_{\text{сч}},$$

где m — ближайшее большее целое число.

Как и всякие дискретные автоматы, счетчики могут быть асинхронными и синхронными (тактируемыми).

Рассмотрим работу суммирующего счетчика с $K_{\text{сч}} = 8$. Для синтеза необходимы $m = \log_2 K_{\text{сч}} = \log_2 8 = 3$ триггера, соответствующие трем разрядам двоичного выходного числа. Обозначим Q_3 — старший, а Q_1 — младший разряды. Выходные сигналы кодируются предельно просто: $y_3 = Q_3$, $y_2 = Q_2$ и $y_1 = Q_1$. Входной сигнал может быть равен только «1», так как нуль не изменяет состояния счетчика, поэтому таблица переключений (табл. 3.16) содержит только $x = 1$.

Таблица 3.16

x^t	Q_3^t	Q_2^t	Q_1^t	Q_3^{t+1}	Q_2^{t+1}	Q_1^{t+1}
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	1	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1
1	1	1	1	0	0	0 — исходное состояние

Из табл. 3.16 видно, что триггер младшего разряда переключается от каждого входного сигнала $x^t = 1$ (нули и единицы в столбце Q_1^{t+1} чередуются один за другим). В столбце следующего разряда Q_2^{t+1} нули и единицы чередуются парами, затем (разряда Q_3^{t+1}) — четверками и т.д. (если увеличивать $K_{\text{сч}}$ и число разрядов). Таким образом, частота переключений каждого следующего триггера уменьшается вдвое по сравнению с частотой переключения предыдущего. Следовательно, счетчик можно построить как

цепочку последовательно включенных счетных триггеров (см. *T*-триггер).

На рис. 3.32, *а* приведена схема суммирующего счетчика, синтезированная на двухступенчатых *JK*-триггерах, работающих в счетном режиме. Подсчитываемые единицы поступают на вход *C* первого триггера, а на выходных шинах формируется двоичный код $y_3y_2y_1$ числа сосчитанных единиц. Триггеры соединены так, что сигнал с прямого выхода предыдущего триггера поступает на вход *C* последующего. Так как на входы *J* и *K* всех триггеров постоянно поданы единицы, каждый триггер переключается в противоположное состояние в момент спада импульса, поданного на его вход *C*, но после задержки $t_{зд i}$, определяющейся временем переходных процессов в триггере (рис. 3.32, *б*).

Представление счетчика цепочкой счетных триггеров справедливо как для суммирующего, так и для вычитающего вариантов, поскольку закономерность по отношению частот переключения разрядов сохраняется при просмотре таблицы как сверху вниз (прямой счет), так и снизу вверх (обратный счет). Различия при этом состоят в направлении переключения предыдущего разряда, вызывающего переключение следующего. При прямом счете следующий разряд переключается при переходе предыдущего в направлении 1—0, а при обратном — при переключении 0—1. Следовательно, различие между вариантами заключается в разном подключении входов триггеров к выходам предыдущих. Для реализации вычитающего счетчика на входы *C* последующих триггеров необходимо подавать сигналы с инверсных выходов предыдущих триггеров.

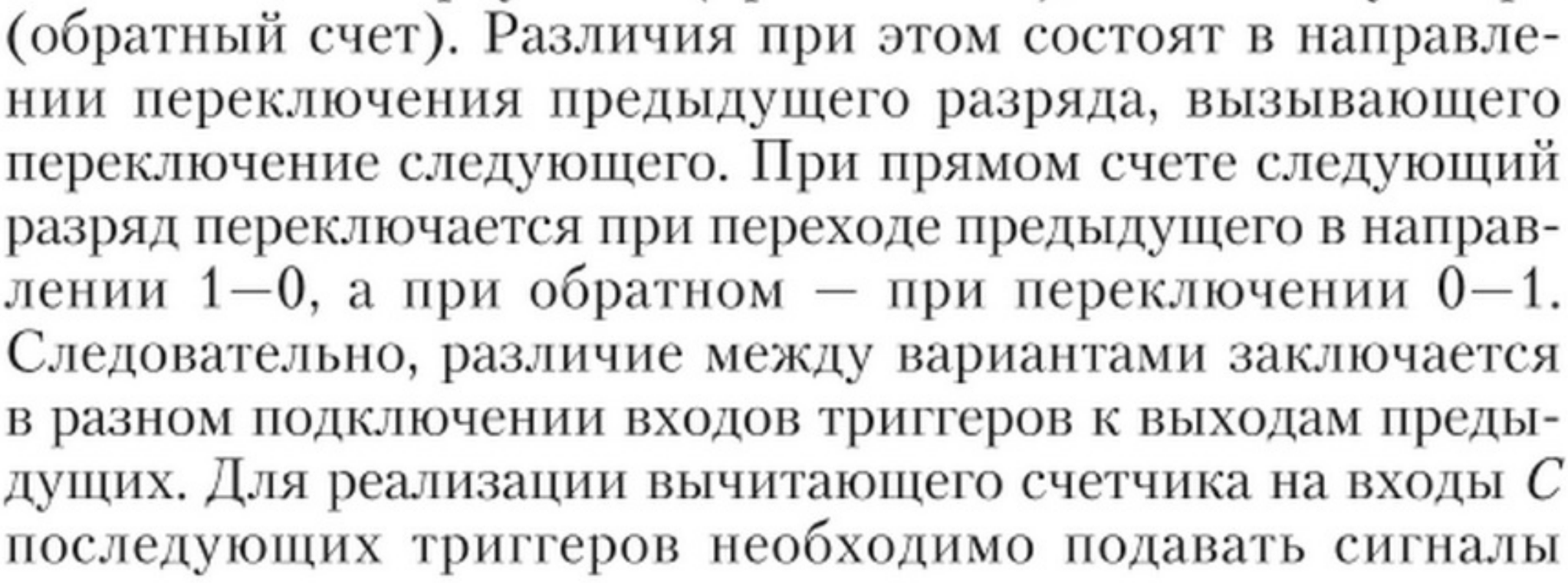


Рис. 3.32. **Асинхронный счетчик с последовательным переносом (*а*) и временная диаграмма его работы (*б*)**

Схема реверсивного счетчика должна работать в двух режимах — суммирующем и вычитающем. Чтобы сделать счетчик реверсивным, необходимо обеспечить подачу на входы триггеров импульсов либо с прямого, либо с инверсного выхода предыдущего триггера в зависимости от управляющего сигнала. Это легко можно сделать с помощью схемы И-ИЛИ (рис. 3.33).

Счетчики, рассмотренные на рис. 3.32 и 3.33, называют счетчиками с последовательным переносом. Переключение каждого последующего триггера происходит с некоторой задержкой $t_{зд}$ только после того, как переключится предыдущий. В результате задержка накапливается и триггер самого старшего разряда m переключается спустя время $t = mt_{зд}$. Таким образом, время установления кода на выходе счетчика прямо пропорционально его разрядности.

Для повышения быстродействия счетчики выполняют синхронными с параллельным (сквозным) переносом. Особенностью таких счетчиков является то, что выходы всех предшествующих разрядов подаются на входы триггера последующего (старшего) разряда, поэтому длительность переходного процесса (время установки кода) определяется длительностью переходного процесса одного разряда и не зависит от количества m триггеров.

Для синтеза счетчика с параллельным переносом обратим внимание на такую закономерность двоичных чисел: изменение старшего разряда на противоположное происходит тогда, когда все предыдущие разряды переходят из состояния «1» в состояние «0». Например, число «семь» в двоичном коде — 0111, а число «восемь» — 1000, число «тридцать один» — 011111, а число «тридцать два» —

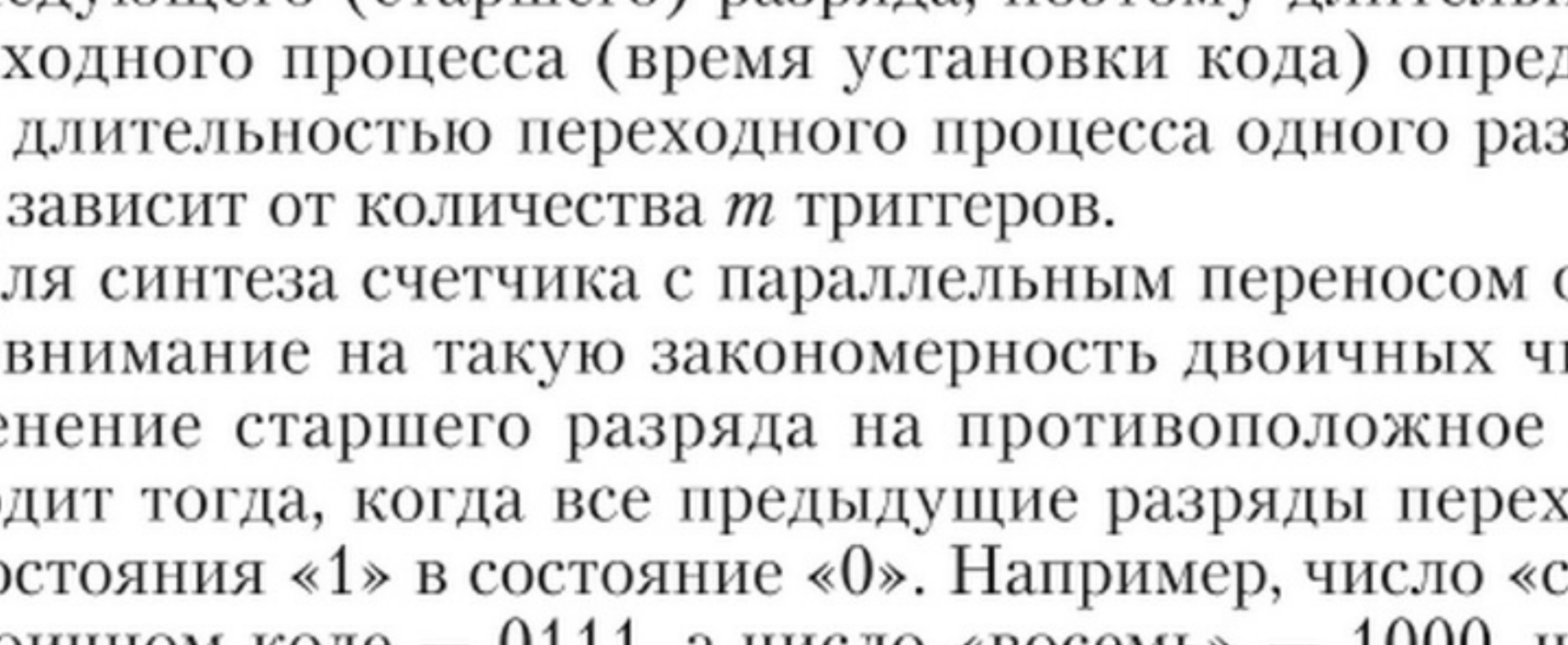


Рис. 3.33. **Реализация реверсивного счетчика с использованием схем И-ИЛИ для межразрядных связей**

Содержание

100000. Таким образом, чтобы сформировать параллельный перенос, достаточно на информационные входы триггеров подавать конъюнкцию сигналов с выходов всех предыдущих триггеров (если использовать JK-триггеры, то одновременно на два входа J и K), а на синхронизирующие входы всех триггеров — импульсы синхронизации, т.е. $Q_i = Q_0Q_1...Q_{i-1}$. Схема синхронного четырехразрядного суммирующего счетчика с параллельным переносом представлена на рис. 3.34. Импульсы синхронизации в этом счетчике становятся счетными импульсами. На входы J и K первого триггера, соответствующего младшему разряду, подается «1».

Рассмотренные счетчики были двоичными, т.е. имели модуль счета, кратный 2^n. Так, например, все триггеры счетчика на рис. 3.34 перейдут в исходное нулевое состояние после прихода шестнадцатого импульса. Значит, модуль счета этого счетчика $K_{сч} = 16 = 2^4$. У двоичных счетчиков сброс всех триггеров происходит с приходом очередного счетного импульса после того, как триггеры установятся в «1». Очевидно, что могут быть построены счетчики с любым модулем счета. Для этого необходимо обеспечить сброс всех триггеров в «0» после достижения максимально возможного числа. Это возможно осуществить двумя путями. Первый связан с традиционным синтезом счетчика на основе его таблицы переключений, второй — с формированием сигнала управляемого сброса всех триггеров в зависимости от модуля счета. *В первом случае* схема счетчика будет оригинальной для каждого модуля счета, *во втором* — типовой, построенной на основе обычных двоичных счетчиков. Для управляемого сброса может быть использован, например, вход R асинхронной установки триггера в нуль. Сигнал управляемого сброса легко сформировать с помо-

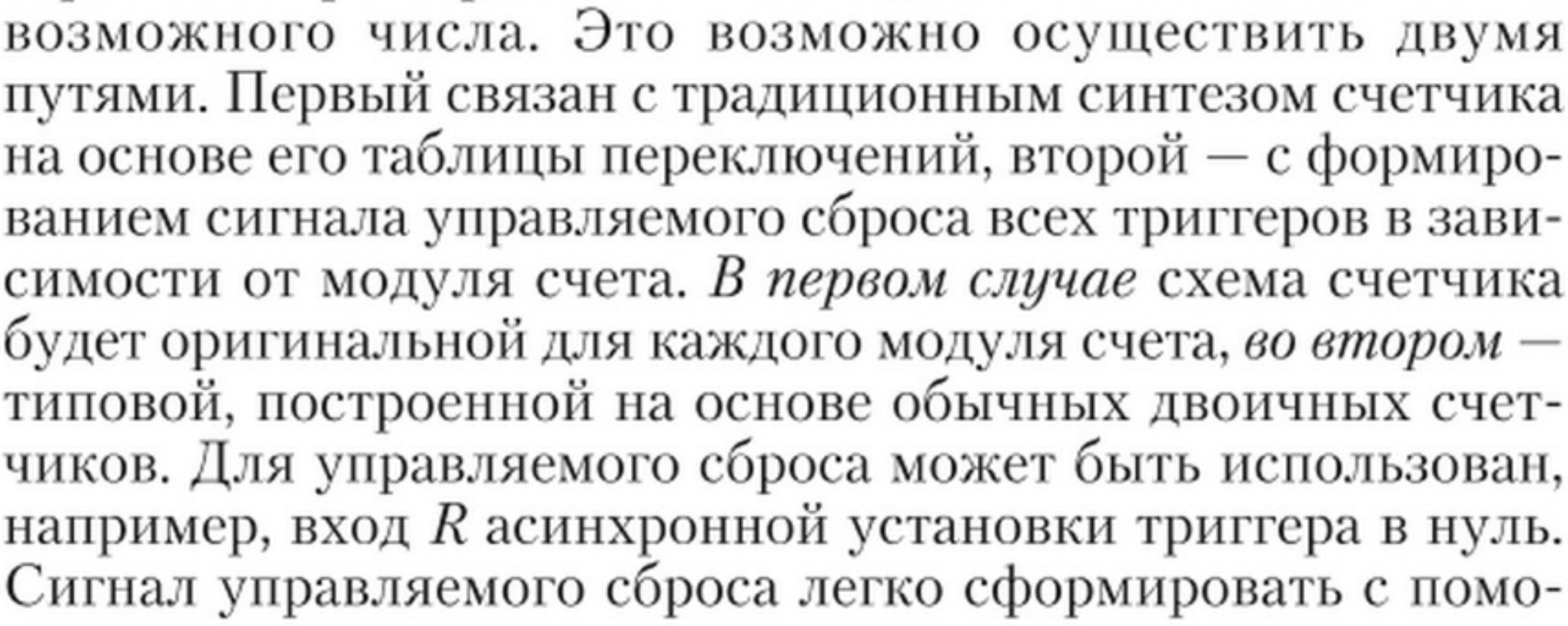


Рис. 3.34. **Счетчик с параллельным переносом**

Содержание

щью схемы И. Для этого на ее входы подаются сигналы с прямых или инверсных выходов триггеров в зависимости от того, каков модуль счета. Пусть, например, необходимо синтезировать счетчик с модулем счета $K_{сч} = 12$. За основу может быть взят четырехразрядный счетчик. С приходом двенадцатого импульса все триггеры должны быть установлены в «0», значит, максимальное число, хранящееся в счетчике, на единицу меньше, т.е. 11. Двоичный эквивалент числа «11» — 1011. Этому числу соответствуют состояния триггеров $Q_3\overline{Q}_2Q_1Q_0$. Соответствующие выходы триггеров подаются на четыре входа схемы И. Поскольку сброс должен происходить в момент подачи счетного импульса, необходима пятивходовая схема И, на пятый вход которой подаются счетные импульсы. Таким образом, для синтеза счетчика требуется типовая схема четырехразрядного двоичного счетчика и один конъюнктор на пять входов.

На основе счетчиков в системах управления технологическими процессами формируют генераторы линейно изменяющихся напряжений, а также генераторы пилообразных напряжений. Действительно, счетчик обеспечивает суммирование числа поступивших на его вход импульсов — в режиме прямого счета, или вычитание — в режиме обратного счета. Если на вход счетчика импульсы будут поступать равномерно с определенной частотой, значит на выходе будет формироваться линейно нарастающая либо спадающая последовательность кодов. В момент перехода счетчика от кода 11...1 до 00...0 (в режиме прямого счета) или от 00...0 до 11...1 (в режиме обратного счета) на выходе будет формироваться мгновенный скачок. Соединив выходы счетчика с цифроаналоговым преобразователем, получим генератор ступенчато изменяющегося напряжения, величина ступеньки которого будет соответствовать цене деления младшего разряда ЦАП. У современных ЦАП цена деления младшего разряда может составлять несколько милливольт, такая форма напряжения с высокой степенью приближения соответствует линейно изменяющемуся напряжению (рис. 3.35).

Крутизна выходного напряжения определяется частотой генератора импульсов. Если использовать генератор с регулируемой переменной частотой, схема будет способна формировать квазилинейное напряжение, скорость нарастания которого может изменяться в самых широких пределах. Наконец, за счет использования реверсивного счетчика

Содержание

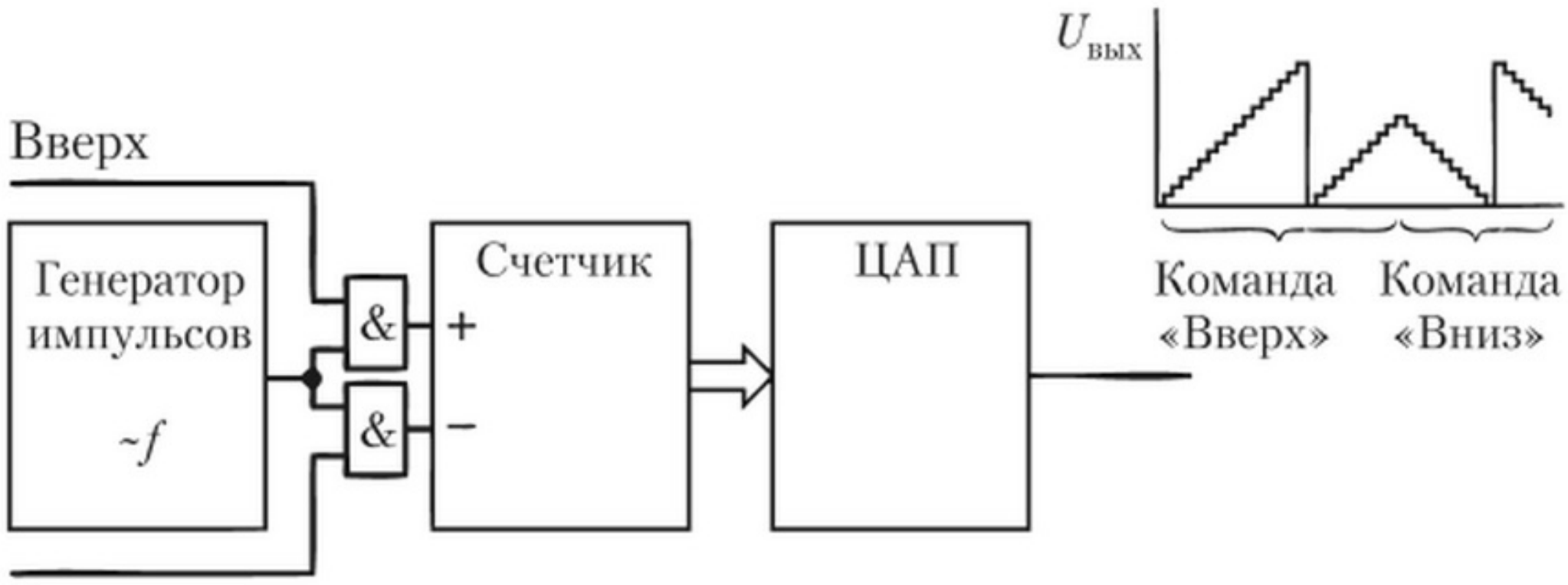


Рис. 3.35. **Схема генератора линейно изменяющегося и пилообразного напряжения на основе реверсивного счетчика и ЦАП**

можно формировать как нарастающее, так и спадающее напряжения. Команда «Вверх» обеспечивает прохождение импульсов на суммирующий вход, и напряжение будет линейно нарастающим, а команда «Вниз» — на вычитающий вход, формируя ниспадающее выходное напряжение. Скачки напряжения формируются в моменты, соответствующие переполнению и сбросу счетчика. Если такие скачки нужно исключить, с помощью логики формируют запрет на прохождение импульсов при достижении критического значения кода.

3.11. Регистры

Регистром называют устройство, осуществляющее прием, хранение и выдачу двоичных чисел в определенном коде. В отличие от запоминающих устройств в регистрах информация хранится не более нескольких тактов. Запоминающими элементами в регистрах служат триггеры, число которых равно числу разрядов хранимых чисел. Схемы регистров служат для ввода и вывода хранимых чисел, преобразования их кодов, сдвига кодов на то или иное число разрядов. Регистры подразделяют на параллельные (регистры памяти), последовательные (регистры сдвига) и параллельно-последовательные (например, ввод — в параллельном коде, а вывод — в последовательном, и наоборот).

В регистрах памяти числа вводят и выводят в параллельном коде, в регистрах сдвига — в последовательном (разряд

за разрядом), поэтому в регистрах памяти число вводится (выводится) за один такт, а в регистрах сдвига — за n тактов, где n — разрядность чисел.

По способу ввода—вывода различают регистры однофазного и парафазного типа. В однофазных ввод (и вывод) возможен только в прямом или обратном коде, в парафазных — как в прямом, так и в обратном кодах. Вид осуществляемого ввода (вывода) определяется сигналами управления.

Исходным состоянием регистра является нулевое. После ввода двоичного числа X_1 триггеры переходят в состояния, соответствующие значениям разрядов числа X_1, которое хранится до момента передачи его другому устройству либо до ввода в регистр другого числа X_2.

В последовательных регистрах при каждом сдвигающем (обычно синхронизирующем) импульсе происходит сдвиг кода хранимого числа влево или вправо на один разряд. При этом число выводится в последовательном коде разряд за разрядом, начиная с младшего (при сдвиге вправо) или старшего (при сдвиге влево). В информационной технике регистры сдвига применяют для преобразования последовательного кода в параллельный (и обратно), для умножения и деления многоразрядных чисел, построения распределителей импульсов, преобразователей кодов и т.д.

В параллельных регистрах схемы разрядов не обмениваются данными между собой. Общими для разрядов обычно являются цепи синхронизации (тактирования), сброса (установки), разрешения выхода (выдачи данных) или приема, т.е. цепи управления.

Для современной схемотехники характерно построение регистров именно на D-триггерах, преимущественно с динамическим управлением. Многие регистры имеют выходы с третьим состоянием, а некоторые — мощные выходные каскады и относятся к числу буферных. Такие регистры способны работать на большую емкостную или низкоомную активную нагрузку. Это обеспечивает их работу непосредственно на магистраль.

3.11.1. Регистровые файлы

Из параллельных регистров составляют блоки регистровой памяти — регистровые файлы. Такие блоки обеспечивают хранение нескольких слов с возможностью независимой и одновременной записи одного слова и чтения другого. Управляющие входы обеспечивают выбор того

или иного регистра и разрешают запись в него слова данных (напомним, что в литературе, посвященной цифровой схемотехнике, под «словом» понимается двоичное число).

Размерность регистровой памяти можно наращивать, составляя из нескольких ИС блок памяти. При наращивании числа хранимых слов выходы отдельных ИС с тремя состояниями соединяются в одной точке. При наращивании разрядности слова входы разрешения и адресации нескольких ИС соединяют параллельно, тогда выходы ИС в совокупности дают единое информационное слово.

3.11.2. Регистры сдвига

Последовательные (сдвигающие) регистры представляют собой цепочку триггеров, связанных цепями переноса.

В однотактных регистрах со сдвигом на один разряд вправо (рис. 3.36, *а*) слово сдвигается при поступлении синхронизирующего сигнала. При каждом импульсе сдвига

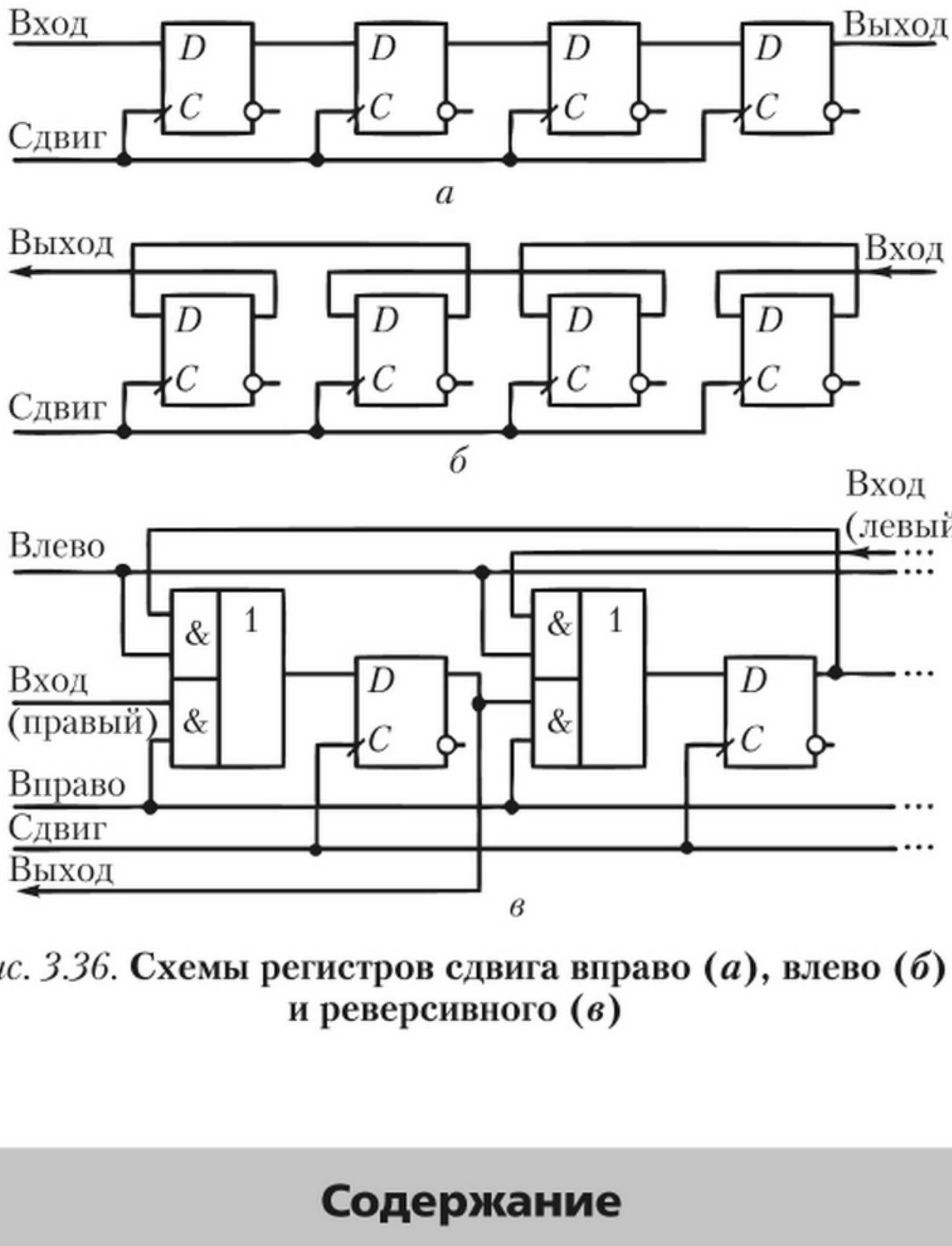


Рис. 3.36. **Схемы регистров сдвига вправо (*а*), влево (*б*) и реверсивного (*в*)**

Содержание

происходит перезапись содержимого предыдущего триггера в последующий и запись очередного разряда в крайний левый триггер. В результате за четыре такта в регистр будет занесено четырехразрядное слово. Причем первый (по времени поступления) разряд будет занесен в крайний правый регистр.

В регистре со сдвигом влево (рис. 3.36, *б*) заполнение его данными происходит справа налево. В реверсивном регистре (рис. 3.36, *в*) имеются связи триггеров с обоими соседними разрядами. С помощью команд «влево» и «вправо» разрешается прохождение сигнала на вход триггера либо через один, либо через другой конъюнктор. Через нижний конъюнктор проходит сигнал с выхода левого триггера, а через верхний — с выхода правого триггера (на рисунке не показан). Естественно, что команды «влево» и «вправо» одновременно подаваться не могут.

3.11.3. Универсальные регистры

В сериях ИС и библиотеках БИС/СБИС программируемой логики имеется много вариантов регистров. Среди них есть многорежимные (многофункциональные) или универсальные, способные выполнять набор микроопераций. Так, например, универсальный регистр может обеспечивать параллельный прием и выдачу данных, сдвиг параллельно записанного слова влево или вправо, поразрядный ввод слова, как это осуществляется в регистрах сдвига, и выдачу этого слова на выход в параллельном коде, сброс всей записанной информации. Многорежимность достигается композицией в одной и той же схеме частей, необходимых для выполнения различных операций. Управляющие сигналы, задающие вид выполняемой операции, активизируют необходимые для этого части схемы. На рис. 3.37 показано условное обозначение типового универсального регистра. Входы D служат для параллельной записи восьмиразрядного слова данных, а выходы Q — для его выдачи. Возможна последовательная поразрядная загрузка слова со сдвигом всех разрядов влево или вправо. Через вход *DSR* осуществляется поразрядная загрузка слова через нулевой разряд со сдвигом всего записываемого слова в сторону старших разрядов. Вход *DSL* обеспечивает загрузку слова через седьмой разряд со сдвигом в сторону младших разрядов. Входы S_0 и S_1 — управляющие. Вход S_0 разрешает сдвиг влево, а вход S_1 — вправо. Вход R обеспечивает асин-

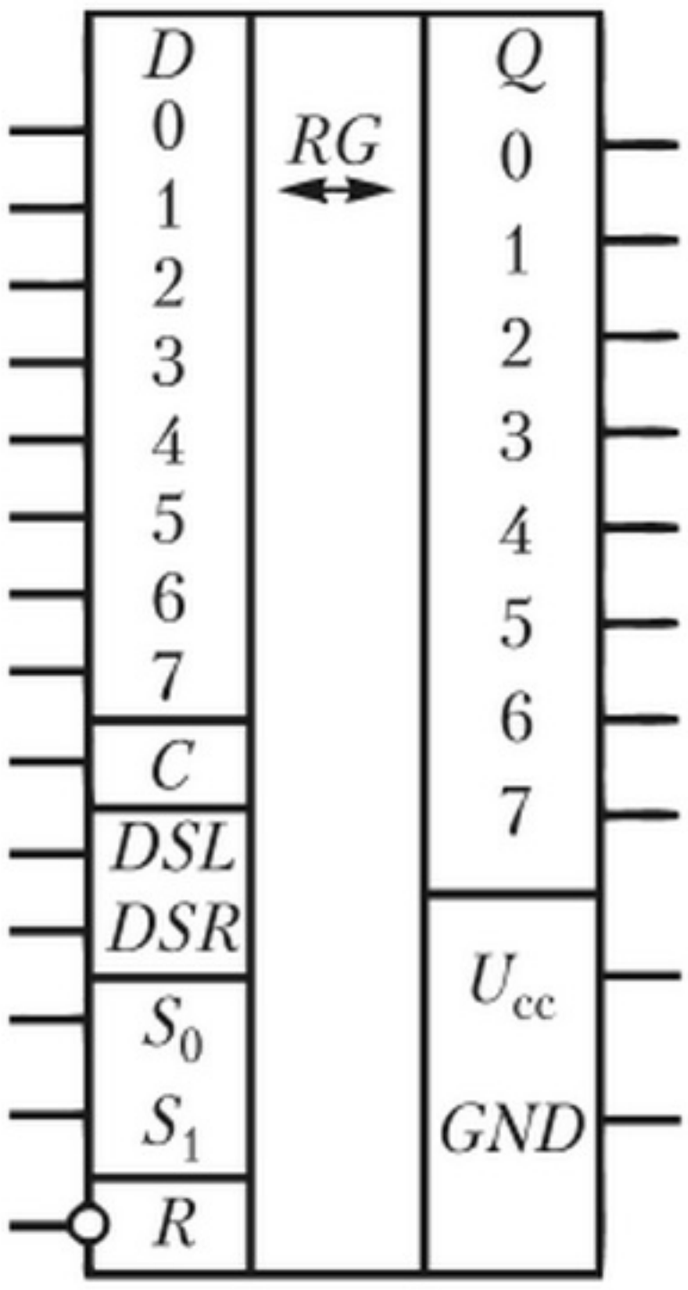


Рис. 3.37. **Универсальный регистр**

хронный сброс всех разрядов регистра, т.е. его «обнуление», вход C — синхронизирующий. Выводы U_{cc} и GND служат для подключения к микросхеме напряжения питания.

Регистры, имеющие разнотипные вход и выход, служат основными блоками преобразователей параллельных кодов в последовательные и обратно. Так, например, через параллельные входы в регистр может быть занесено слово данных. Подавая в регистр импульсы сдвига, на его последовательном выходе можно получить последовательность разрядов загруженного слова. Если регистр имеет последовательный вход и параллельный выход, то в него в последовательном коде разряд за разрядом можно занести слово данных, а через параллельный выход его целиком прочитать. Преобразование параллельного кода в последовательный — очень актуальная задача, так как передача цифровой информации в сетях осуществляется в последовательном коде, а обработка ее в процессорах вычислительных устройств — в параллельном.

Контрольные вопросы и задания

1. Каково представление в двоичной системе чисел 14 и 11?
2. Выполните сложение чисел 14 и 11 в двоичном коде.

Содержание

3. Опишите суть операций конъюнкции, дизъюнкции и инверсии.

4. Запишите СовДНФ для функции $\overline{F'}$, получаемой из функции F', как инверсия ее значений на всех наборах (см. табл. 3.4).

5. Составьте логическую функцию с помощью карт Карно (см. рис. 3.2), если она равна 1 на всех наборах $ABCD = 0101; 0111; 1101; 1111$ (единицы в центральном квадрате).

6. Чему равно значение логической функции $F = a \wedge \bar{b} \wedge \bar{c}$, если $a = 1, b$ 0, ⇐ 0? =

7. Что изменится, если $b = 1$?

8. Упростите логическую функцию $F = a \wedge (\bar{b} \vee b) \wedge c \wedge d$.

9. Какую логическую функцию выполняет ключевая схема?

10. Для чего в ключевой схеме на биполярном транзисторе применяют диод Шоттки?

11. За счет чего ключевая схема на комплементарных транзисторах обеспечивает низкую потребляемую мощность?

12. Какую логическую функцию реализует базовый элемент ТТЛШ в позитивной логике? В негативной логике?

13. Определите, в каком состоянии будут транзисторы на схеме рис. 3.11, и какой будет сигнал на выходе схемы, если на вход x_1 подан лог. 0.

14. Сколько выходов будет иметь четырехразрядный дешифратор? Сколько для его реализации потребуется трехразрядных дешифраторов?

15. Сколько схем «ИЛИ» понадобится для реализации шифратора на 25 входных сигналов (унитарный код)?

16. Какой дешифратор необходим для реализации 16-входового мультиплексора (см. рис. 3.17)?

17. Почему в сумматоре с параллельным переносом чем старше разряд, тем сложнее схема ускоренного переноса?

18. В чем недостаток многоразрядного сумматора, выполненного путем последовательного соединения одноразрядных?

19. Какие операции может выполнять арифметико-логическое устройство?

20. Какой из триггеров самый простой? Почему?

21. За счет чего *RS*-триггер сохраняет свое состояние после кратковременной подачи сигналов на его входы *R* или *S*?

22. В чем состоит принцип работы *D*-триггера? Почему на его основе можно легко получить *T*-триггер?

23. Как с помощью *JK*-триггера получить *RS*-, *D*- и *T*-триггеры?

24. Какое минимальное число импульсов может «подсчитать» счетчик на рис. 3.22? Каким станет счетчик, если к выходу последнего триггера подключить еще один?

25. Какой недостаток счетчиков с последовательным переносом устраняется в счетчиках с параллельным переносом?

Содержание

26. Можно ли на основе *T*-триггеров построить реверсивный счетчик? Почему?

27. Чем отличается принцип записи в регистровый файл по сравнению с регистром сдвига?

28. Каковы возможности по записи данных у универсального регистра?

Содержание

Глава 4
СХЕМОТЕХНИКА ПРОГРАММИРУЕМЫХ ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

В результате изучения главы 4 студент должен:

знать

- основы построения запоминающих устройств;
- структуру микропроцессорных систем и режимы обмена;
- работу основных узлов микропроцессора;
- работу основных интерфейсных устройств;
- виды программируемых логических интегральных схем;
- принципы функционирования микропроцессоров как программируемых устройств;
- поколения микропроцессоров и их отличия;

уметь

- использовать полупроводниковые запоминающие устройства различного назначения;
- выполнять сопоставительный анализ микропроцессоров различных поколений;

владеть

- приемами разработки простейших ассемблерных программ.

4.1. Запоминающие устройства

Работа программно-управляемых цифровых вычислительных устройств и систем заключается в последовательном выполнении команд программы. В процессе работы происходит обращение за данными, при этом результаты работы одних устройств часто являются исходной информацией для работы других. Постоянно происходит обмен информацией между отдельными частями системы. Таким образом, для функционирования цифровых систем обработки информации необходимы устройства, которые обе-

спечивали бы хранение программы работы, хранение исходных данных и результатов обработки, обмен цифровой информацией между отдельными частями системы. Эти функции выполняют полупроводниковые запоминающие устройства (ЗУ). Микросхемы памяти в общем объеме выпуска ИС занимают около 40% и играют важнейшую роль во многих системах различного назначения. Микросхемы и системы памяти постоянно совершенствуются. Повышается быстродействие микросхем ЗУ, растут объемы информации, которые могут храниться в одной микросхеме, совершенствуется архитектура ИС. В настоящее время созданы и используются десятки различных типов ЗУ.

4.1.1. Важнейшие параметры ЗУ

Информационная емкость — максимально возможный объем хранимой информации. Выражается в битах или словах (в частности, в байтах). Бит хранится запоминающим элементом (ЗЭ), а слово — запоминающей ячейкой, т.е. группой ЗЭ, к которым возможно лишь одновременное обращение. Добавление к единице измерения множителя «К» (кило) означает умножение на $2^{10} = 1024$, а множителя «М» (мега) — умножение на $2^{20} = 1\,048\,576$.

Организация ЗУ определяет, сколько слов и какой разрядности хранит запоминающее устройство. ЗУ одинаковой емкости могут иметь разную организацию. Например, ЗУ, хранящее 64 8-разрядных слова, имеет емкость 512 бит. Ту же емкость имеет ЗУ, хранящее 128 4-разрядных слов.

Быстродействие (производительность) ЗУ оценивают временем считывания, записи и длительностью циклов чтения/записи. Время считывания — интервал между моментами появления сигнала чтения и слова на выходе ЗУ. Время записи — интервал после появления сигнала записи, достаточный для установления запоминающей ячейки в состояние, задаваемое входным словом. Минимально допустимый интервал между последовательными чтениями или записями образует соответствующий цикл.

Для любой цифровой вычислительной системы характерна *шинная структура* (подробнее см. параграф 4.4). Чтобы процессор мог однозначно выбрать нужную команду или данные, он обращается к определенной ячейке ЗУ, которая имеет свой номер. Этот номер называется *адресом*. Адрес ячейки передается по шине адреса *A*. Очевидно, что

число различных адресов, которые могут быть переданы по шине, содержащей n проводников, равно

$$N = 2^n.$$

При этом самый младший адрес ячейки состоит из одних нулей A = 00...0, а самый старший — из одних единиц A = 11...1 = 2^{n-1}. Поэтому нумерация ячеек начинается с нуля.

Типичный набор сигналов полупроводникового ЗУ (рис. 4.1) включает следующие сигналы:

- A — адрес, разрядность которого n определяется максимально возможным числом хранимых в ЗУ слов. Для полупроводниковых ЗУ число ячеек, размещаемых в одной ИС, должно быть кратно целой степени числа 2, например: 2^9 = 512; 2^{10} = 1 Кбайт = 1024; 2^{20} = 1 Мбайт = 1 048 576. Разрядность адреса n связана с числом ячеек N соотношением $n = \log_2 N$. Например, ЗУ с информационной емкостью 64 Кбайт с байтной организацией имеет 16-разрядные адреса, выражаемые словами $A = A_{15}A_{14}A_{13}...A_0$;
- *DI* (Data Input) и *DO* (Data Output) — шины входных и выходных данных, разрядность которых m определяется организацией ЗУ (разрядностью его ячеек). В некоторых ЗУ для входных и выходных данных используется одна и та же шина, обозначаемая как *DIO*;
- *R/W* (Read/Write) — задает выполняемую операцию (при единичном значении — чтение, при нулевом — запись);
- *CS* (Chip Select) или *CE* (Chip Enable) — разрешает или запрещает работу данной микросхемы. Для организации памяти вычислительного устройства одной микросхемы ЗУ обычно недостаточно. В этом случае с помощью сигнала *CS* обеспечивается обращение к нужной микросхеме, на остальные же микросхемы при этом подается запрещающий уровень сигнала *CS*;

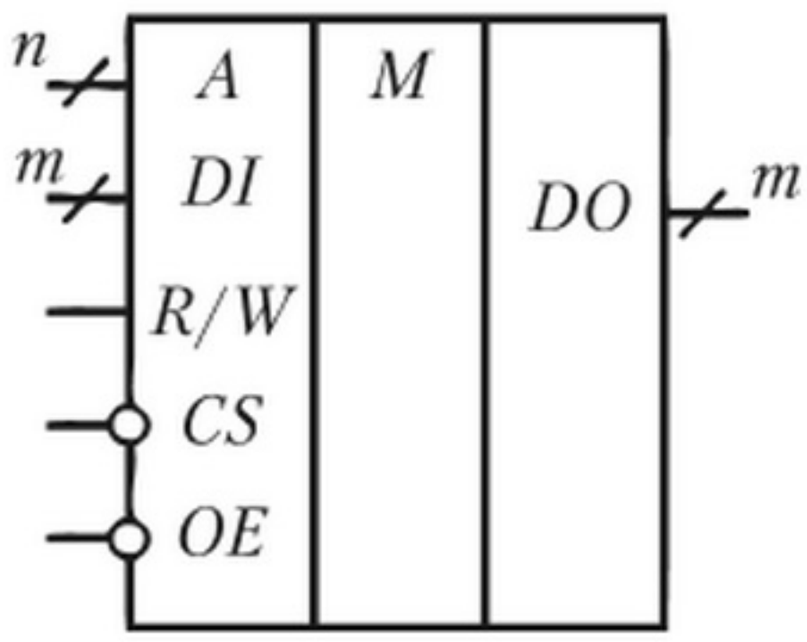


Рис. 4.1. **Типичные сигналы ЗУ**

• *OE* (Output Enable) — осуществляет управление выходными буферными каскадами. Чтобы ИС не нагружала выходные линии, ее выходные каскады кроме обычных состояний лог. 1 и лог. 0 имеют еще третье состояние высокого выходного сопротивления, эквивалентного отключению ИС от шины данных. Вход *OE* обеспечивает перевод выходных буферных каскадов в третье состояние.

Очередность поступления рассмотренных выше сигналов следующая. Прежде всего, подается адрес, чтобы последующие операции не коснулись какой-либо другой ячейки, кроме выбранной. Затем сигналом *CS* (*CE*) разрешается работа микросхемы, после чего подается сигнал чтения/записи *R/W*. Если задана, например, операция чтения, то после подачи перечисленных сигналов ЗУ готовит данные для чтения, что требует определенного времени, спустя которое на выходах *DO* появляются данные. По заднему фронту сигнала *R/W* производится считывание данных. Если же задана операция записи, то данные, подлежащие записи, подаются на входы *DI* и по заднему фронту сигнала *R/*W происходит запись данных в ЗУ.

Для полупроводниковых ЗУ характерна специфическая характеристика — *свойство энергонезависимости* — способность ЗУ сохранять данные при отключении напряжения питания. Если ЗЭ обладают свойством энергонезависимости, то память, построенная на их основе, будет хранить информацию при отключенном питании неограниченно долгое время. Если же ЗЭ этим свойством не обладают, а информацию необходимо сохранять, в память вводят резервные источники питания, автоматически подключаемые при снятии основного питания.

4.1.2. Классификация полупроводниковых ЗУ

Важнейшим признаком ЗУ является способ доступа к данным, в соответствии с которым их классифицируют на три типа:

- адресные ЗУ;
- ЗУ с последовательным доступом;
- ЗУ с ассоциативным доступом.

Первыми были созданы полупроводниковые адресные ЗУ. Чтобы обратиться к ячейке адресного ЗУ для чтения или записи информации, необходимо указать адрес этой ячейки. Эти ЗУ лежат в основе постоянной и оперативной памяти всех ЭВМ и вычислительных устройств, построен-

ных на базе микропроцессоров, в частности, устройств числового программного управления станков (УЧПУ).

ЗУ с последовательным доступом используются там, где поступающие данные могут быть выстроены в очередь. Очередь может обслуживаться либо по принципу «первый пришел — первый ушел» — FIFO (*first in first out*), либо «последний пришел — первый ушел» — LIFO (*last in first out*), т.е. запись данных осуществляется не по какому-либо адресу, а в конец очереди, а чтение — либо из конца, либо из начала очереди. ЗУ с последовательным доступом могут использоваться, например, в видеопамяти, обеспечивающей хранение данных о цвете и яркости каждой точки экрана монитора.

В ЗУ с ассоциативным доступом поиск информации осуществляется не по адресу ячейки и не по месту в очереди, а по некоторому признаку. Отсюда в их названии присутствует слово «ассоциация». Наиболее важной областью использования ассоциативных ЗУ является кэш-память ЭВМ (подробнее см. параграф 4.7).

Адресные ЗУ. Эти ЗУ наиболее разработаны, и другие виды памяти часто строят на основе адресной с соответствующими модификациями.

Адресные ЗУ делятся на два типа: RAM и ROM.

RAM (Random Access Memory) — русские синонимы ОЗУ (оперативные ЗУ) или ЗУПВ (ЗУ с произвольной выборкой). Оперативные ЗУ хранят данные, участвующие в обмене при исполнении текущей программы, которые могут быть изменены в произвольный момент времени, либо коды самой программы. Для этого программа предварительно должна быть загружена в оперативную память. Запоминающие элементы ОЗУ, как правило, не обладают энергонезависимостью.

ROM (Read Only Memory) — русский эквивалент — ПЗУ (постоянные ЗУ). В процессе работы вычислительного устройства содержимое ПЗУ не изменяется. Процессор вычислительного устройства из этой памяти может лишь прочитать данные или очередные команды программы. Отсюда буквальный перевод английского названия этого вида ЗУ — «память только для чтения». На основе этого вида ЗУ изготавливается постоянная память вычислительных устройств. Информация в микросхему ЗУ заносится либо в процессе ее производства, либо пользователем в специальном режиме программирования.

ОЗУ делятся на статические и динамические. В *статических* ЗУ запоминающими элементами являются триггеры, состоящие из нескольких транзисторов. В *динамических* ЗУ информация хранится в виде зарядов конденсаторов, образуемых элементами МОП-транзисторов, и для одного ЗЭ достаточно одного транзистора. Поэтому емкость динамических ЗУ в несколько раз превышает емкость статических. Статические ЗУ называются SRAM (Static RAM), а динамические — DRAM (Dynamic RAM).

Статические ОЗУ можно разделить на асинхронные, тактируемые (синхронные) и конвейерные. В *асинхронных* сигналы управления могут задаваться как импульсами, так и уровнями. Например, сигнал разрешения работы *CS* может оставаться неизменным и разрешающим на протяжении многих циклов обращения к памяти. В *тактируемых* ЗУ сигнал разрешения работы *CS* в каждом цикле обращения к памяти должен переходить из пассивного состояния в активное, т.е. должен формироваться фронт этого сигнала в каждом цикле. Этот тип ЗУ называют также синхронным. В *конвейерных* ЗУ тракт передачи данных реализован по конвейерному принципу. Конвейер работает с тактовой частотой процессора. Это позволяет повысить темп передачи данных в несколько раз.

Так как динамические ЗУ характеризуются наибольшей информационной емкостью, именно они используются как основная память ЭВМ. Разработаны многочисленные схемы повышенного быстродействия.

Статические ЗУ в 4—5 раз дороже динамических и во столько же раз меньше по информационной емкости. Их достоинством является более высокое быстродействие, а типичной областью использования — схемы кэш-памяти.

Постоянная память типа ROM (М) программируется при изготовлении методами интегральной технологии с помощью одной из используемых при этом масок. Поэтому в русской терминологии эти ЗУ называются масочными ПЗУ. Для потребителя это в полном смысле постоянная память, так как изменить ее содержание он не может.

В следующих трех разновидностях ROM в обозначениях присутствует буква *P* (от *programmable*). Это программируемая пользователем память (в русской терминологии ППЗУ — программируемые ПЗУ). Для записи информации в ЗЭ подаются специальные электрические сигналы. В ЗУ типа PROM данные могут быть занесены один раз

путем прожигания плавких перемычек или, наоборот, за счет создания перемычек путем электрического пробоя. В ПЗУ типа EPROM и EEPROM имеется возможность стирания старой информации и записи новой, поэтому их называют репрограммируемыми ПЗУ. В EPROM стирание выполняется путем облучения кристалла ультрафиолетовыми лучами, а в EEPROM — электрическими сигналами. Программирование этих разновидностей ROM обычно производится пользователем с помощью специальных приборов — программаторов — в лабораторных условиях.

Память флэш-типа по запоминающему элементу подобна памяти типа EEPROM, отличие между ними — в скорости стирания информации. Чтобы стереть информацию из ПЗУ типа EEPROM, необходимо обратиться к каждой запоминающей ячейке, что требует сравнительно большого времени. В ПЗУ флэш-типа вся информация может быть стерта подачей одного сигнала, т.е. мгновенно (*flash* — вспышка). В некоторых ПЗУ флэш-типа весь объем запоминающих ячеек делится на блоки и стирание информации осуществляется поблочно.

ЗУ с последовательным доступом. В ЗУ этого класса записываемые данные образуют очередь. Считывание происходит из очереди слово за словом либо в порядке записи, либо в обратном порядке. Моделью такого ЗУ является последовательная цепочка запоминающих элементов, в которой данные передаются между соседними элементами.

Прямой порядок считывания имеет место в буферах FIFO, а также в файловых и циклических ЗУ.

Разница между памятью FIFO и файловым ЗУ состоит в том, что в FIFO запись в пустой буфер сразу же становится доступной для чтения, т.е. поступает в конец цепочки. В файловых ЗУ данные поступают в начало цепочки и появляются на выходе после некоторого числа обращений, равного числу элементов в цепочке. Записываемые данные объединяются в блоки, обрамляемые специальными символами конца и начала (*файлы*). Прием данных из *файлового* ЗУ начинается после обнаружения приемником символа начала блока.

В циклических ЗУ слова доступны одно за другим с постоянным периодом, определяемым емкостью памяти. К такому типу среди полупроводниковых ЗУ относится видеопамять (VRAM).

Считывание в обратном порядке свойственно *стековым* ЗУ, для которых реализуется принцип «последний пришел — первый ушел». Такие ЗУ называют буферами LIFO.

Содержание

ЗУ с ассоциативным доступом. ЗУ этого типа реализует поиск информации по некоторому признаку, а не по ее расположению в памяти (адресу или месту в очереди). В наиболее полной версии все хранимые в памяти слова одновременно проверяются на соответствие признаку, например на совпадение определенных полей слов — *тегов* (англ. — *tag*) с признаком, задаваемым входным словом (*теговым адресом*). Таким признаком может быть полный адрес ячейки оперативной памяти. (Заметим, что емкость оперативной памяти современных компьютеров составляет гигабайты, в то время как емкость ассоциативного ЗУ измеряется сотнями килобайт — единицами мегабайт.) На выход выдаются слова, удовлетворяющие признаку. Принцип выдачи слов, если тегу удовлетворяют несколько слов, а также принцип записи новых данных могут быть разными. Основная область применения ассоциативной памяти современных ЭВМ — кэш-память данных.

4.1.3. Структура адресных ЗУ

В полупроводниковых ЗУ адресного типа можно выделить две основные функциональные части: матрицу запоминающих элементов, обеспечивающую хранение данных, и схему выборки, обеспечивающую запись или считывание информации в выбранной ячейке памяти (рис. 4.2). Матрица ЗЭ — прямоугольная и в ранних образцах ЗУ имела размерность $M = k \cdot m$, где M — информационная

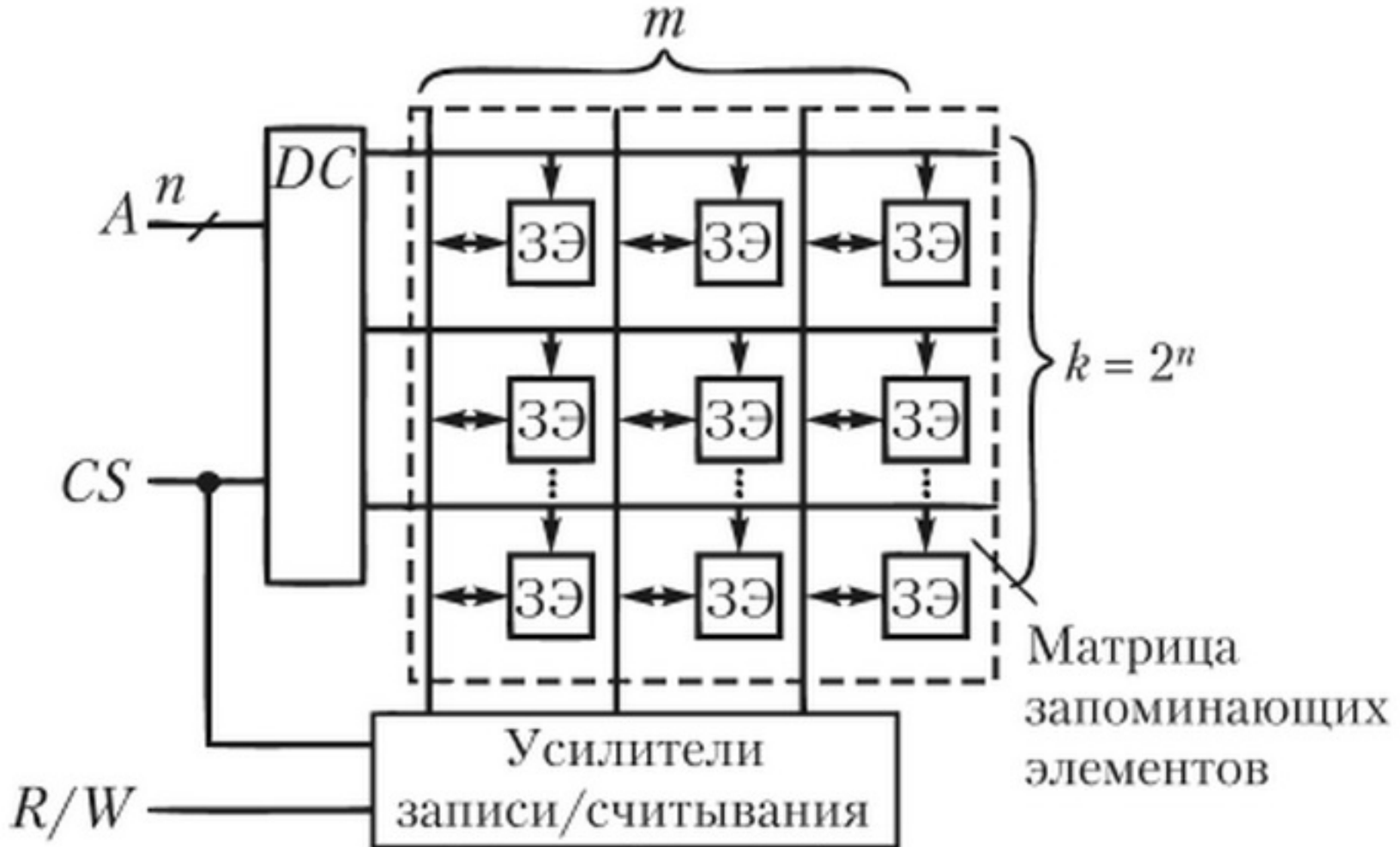


Рис. 4.2. **Структура ЗУ типа 2*D***

емкость памяти в битах; k — число хранимых слов; m — их разрядность. Схема выборки состоит из дешифратора адресного кода DC и усилителей записи/считывания.

Каждая строка матрицы ЗЭ соответствует ячейке памяти и хранит одно слово данных, а номера строк соответствуют адресам ячеек. Адрес выбранной ячейки по шине адреса поступает на все микросхемы памяти, но воспринимается только той микросхемой, на которую пришел разрешающий сигнал *CS* (*Chip Select* — выбор кристалла). С приходом сигнала *CS* дешифратор адресного кода активизирует одну из выходных линий, разрешая одновременный доступ ко всем элементам выбранной строки. Одноименные разряды всех ячеек образуют столбцы матрицы, соединяемые вертикальными линиями — внутренними линиями данных. Если осуществляется операция чтения, то состояния ЗЭ выбранной строки транслируются по этим линиям, усиливаются и выдаются на внешнюю шину данных. Если же выполняется операция записи, то усилители устанавливают на линиях уровни напряжения в соответствии с записываемыми данными, и ЗЭ переходят в соответствующие состояния «нуля» или «единицы». Подобная структура ЗУ получила название структуры 2*D*.

Структура 2*D* может быть использована только в ЗУ малой информационной емкости. С ростом емкости возрастает сложность дешифратора, поскольку число его выходных линий равно числу хранимых слов. Для ЗУ большой информационной емкости структура была усовершенствована, она получила название структуры 2*DM*. На рис. 4.3 показана такая структура для ЗУ типа ROM. В ней возбужденный выход дешифратора DC_x по-прежнему выбирает целую строку матрицы ЗЭ. Однако, в отличие от предыдущей структуры, длина строки не равна разрядности хранимых слов, а многократно ее превышает. Число строк матрицы, а значит, и число выходов дешифратора теперь меньше количества хранимых слов. Для выбора одной из строк служат не все разряды адресного кода, а их часть A_{n-1}, ..., A_k. Остальные разряды адреса (от A_{k-1} до A_0) используются для того, чтобы выбрать необходимое слово из множества слов, содержащихся в этой строке. Это выполняется с помощью мультиплексоров, на адресные входы которых подаются коды A_{k-1}, ..., A_0. Длина строки равна $m2^k$, где m — разрядность хранимых слов. Каждый отрезок строки длиной 2^k хранит все одноименные (нулевые, первые, ..., m-е) разряды всех 2^k слов этой строки. Из каждого

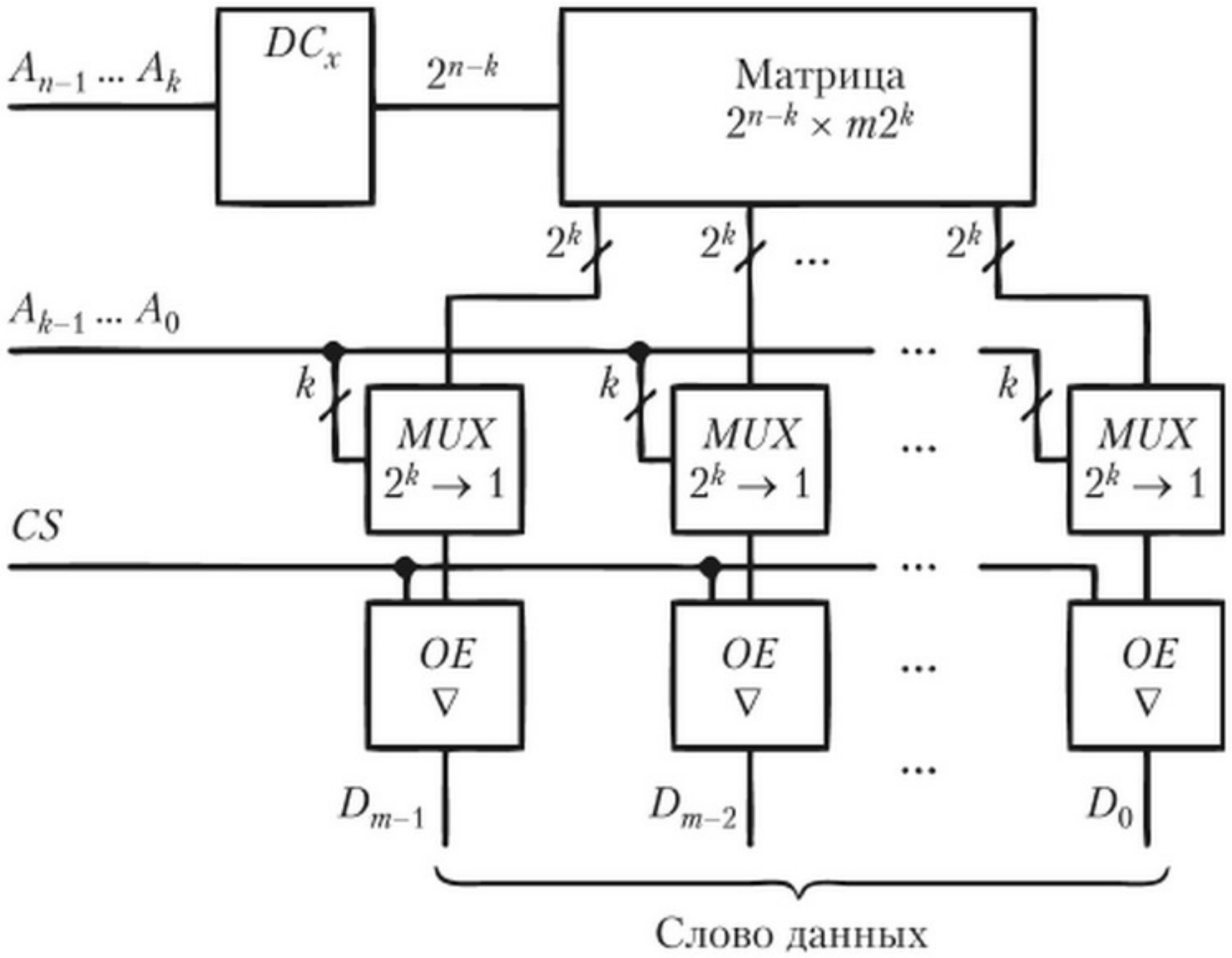


Рис. 4.3. **Структура ЗУ типа 2*DM***

такого отрезка мультиплексор выбирает один бит. Количество мультиплексоров соответствует количеству разрядов в слове. Таким образом, на выходах мультиплексоров формируется выходное слово. По разрешению сигнала *CS*, поступающего на входы *OE* управляемых буферов с тремя состояниями, выходное слово передается на внешнюю шину.

Структура 2*DM* используется не только для ЗУ типа ROM, но и для ЗУ типа RAM с операциями чтения и записи. В этих ЗУ вместо мультиплексоров используются управляемые буферы данных, выполняющие не только функции мультиплексирования и выдачи данных в шину *DO*, но и функции приема данных по шине *DI* и формирования сигналов записи для соответствующих ЗЭ. Направление передачи данных определяется сигналом *R/W*, а управление буферами данных осуществляется дешифратором.

4.2. Запоминающие устройства для хранения постоянной информации

Запоминающие устройства типа ROM хранят информацию, которая либо вообще не изменяется (в ЗУ типа ROM (M) и PROM), либо изменяется редко и не во время

работы вычислительного устройства (в ЗУ типа EPROM и EEPROM).

В масочные ЗУ типа ROM (М) информацию записывают при изготовлении на промышленных предприятиях с помощью шаблона (маски) на завершающем этапе технологического процесса.

ЗУ типа PROM программируют после изготовления их предприятием электронной промышленности в лабораториях потребителей без использования сложных технологических процессов. Для этого используют несложные устройства — программаторы.

ЗУ типа ROM имеют многоразрядную организацию (чаще всего 8- или 4-разрядную, для некоторых ИС 16-разрядную) и обычно выполняются по структуре 2*DM*.

4.2.1. Масочные ЗУ

В качестве запоминающих элементов в масочных ЗУ могут выступать диоды, биполярные транзисторы, МОП-транзисторы и т.д.

В матрице диодного ROM (М) (рис. 4.4, *а*) горизонтальные линии (строки) являются линиями выборки слов, а вертикальные — линиями считывания. Считываемое слово определяется расположением диодов в узлах координатной сетки. Для считывания нужного слова в соответствующую строку подается импульс напряжения. При наличии диода высокий потенциал передается на соответствующую вертикальную линию, и в данном разряде слова появляется сигнал логической единицы. При отсутствии диода потенциал близок к нулевому, так как вертикальная линия через резистор связана с корпусом. В изображенной матрице при подаче импульса в линию выборки Л1 считывается слово 11010001 (это слово хранится в ячейке номер 1). При подаче импульса в линию Л2 считывается слово 10101011 (оно хранится в ячейке номер 2). Линии выборки являются выходами дешифратора адреса, каждая адресная комбинация формирует единицу только на своем выходе дешифратора, что приводит к считыванию слова только из адресуемой ячейки.

Обычно при производстве масочных ЗУ выпускается целая номенклатура однотипных микросхем, отличающихся различной «зашитой» в них информацией. Для удешевления производства стремятся сделать так, чтобы процесс

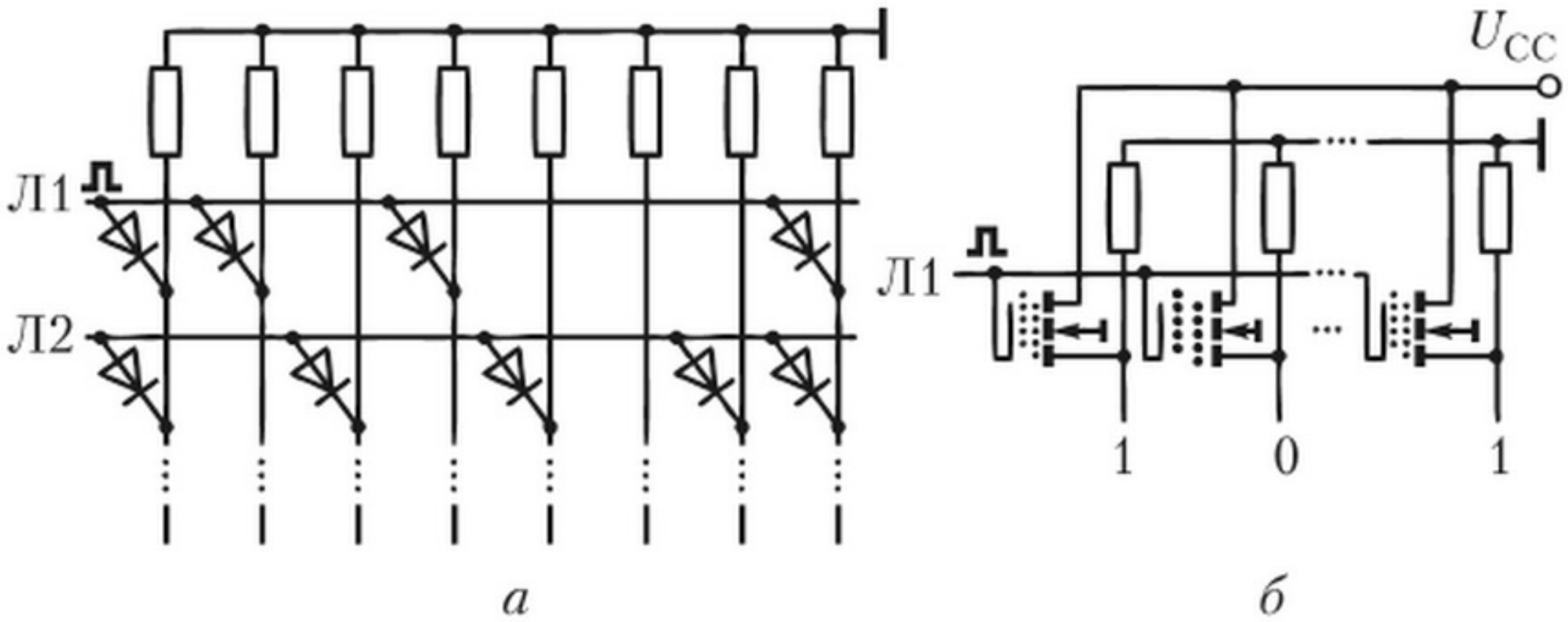


Рис. 4.4. **Матрицы запоминающих элементов масочного ЗУ:**
а — диодная; *б* — МОП-транзисторная

изготовления различных микросхем отличался минимумом технологических операций. Например, если в качестве ЗЭ используются МОП-транзисторы (на рис. 4.4, *б* показана одна из строк матрицы), то на этапе изготовления затворов в транзисторах, соответствующих хранению логического нуля, увеличивают толщину подзатворного окисла. Это ведет к увеличению порогового напряжения этих транзисторов, и при подаче обычного рабочего уровня они остаются закрытыми. Таким образом, при производстве всей номенклатуры микросхем все технологические операции, кроме одной — напыление подзатворного окисла, будут одинаковыми. Для этого требуются только различные шаблоны — маски, через которые осуществляется процесс напыления.

ЗУ с масочным программированием отличаются компактностью ЗЭ и, следовательно, высоким уровнем интеграции. Область применения масочных ЗУ — хранение стандартной информации, имеющей широкий круг потребителей. В частности, масочные ЗУ используются для хранения кодов букв русского и латинского алфавитов, таблиц типовых функций (синуса, квадратичной функции и др.), стандартного программного обеспечения и т.п.

4.2.2. ЗУ типа PROM

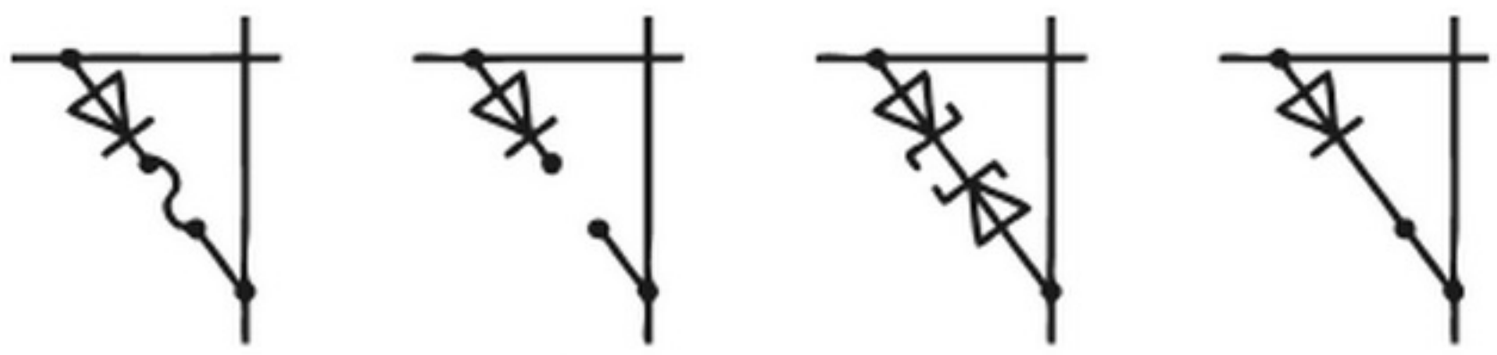
В ЗУ типа PROM микросхемы программируют устранением или созданием специальных перемычек. Поэтому этот вид ЗУ позволяет занести информацию только один раз. В исходной заготовке имеются (или отсутствуют) все

перемычки. После программирования остаются (или возникают) только необходимые.

Часть перемычек устраняют при программировании ЗУ с плавкими перемычками. При этом в исходном состоянии ЗУ имеет все перемычки, а при программировании часть их ликвидируют путем расплавления импульсами тока достаточно большой амплитуды и длительности. В исходном состоянии ЗЭ хранит логическую единицу, для записи логического нуля перемычку необходимо расплавить.

Создание части перемычек соответствует схемам ЗУ, которые в исходном состоянии имеют непроводящие перемычки в виде пары встречно включенных диодов или тонких диэлектрических слоев, пробиваемых при программировании. В исходном состоянии цепи всех запоминающих элементов хранят логический нуль. При программировании на запоминающий элемент подается напряжение, пробивающее встречный диод или диэлектрическую перемычку. В результате создается проводящая перемычка и записывается логическая единица.

Запоминающие элементы с плавкими перемычками и парами диодов показаны на рис. 4.5, *а*, *в* в исходном состоянии и после программирования на рис. 4.5, *б*, *г*.



а *б* *в* *г*

Рис. 4.5. **Запоминающие элементы с плавкими перемычками и диодными парами:**

а, *в* — до программирования; *б*, *г* — после программирования

4.2.3. ЗУ типов EPROM и EEPROM

В репрограммируемых ЗУ типов EPROM и EEPROM возможно стирание старой информации и замена ее новой в результате специального процесса. В ЗУ типа EPROM старая информация стирается путем засвечивания кристалла через прозрачное окно мощным ультрафиолетовым излучением. В ЗУ типа EEPROM стирание осуществляется подачей специальных электрических импульсов.

Запоминающими элементами современных репрограммируемых ЗУ являются транзисторы типов ЛИЗМОП (добавление ЛИЗ к обозначению МОП происходит от слов Лавинная Инжекция Заряда) и МНОП. МНОП-транзистор отличается от обычного МОП-транзистора двухслойным подзатворным диэлектриком. На поверхности кристалла расположен тонкий слой диоксида кремния SiO_2, далее более толстый слой нитрида кремния Si_3N_4 (отсюда буква «Н» в аббревиатуре МНОП) и затем уже затвор (рис. 4.6, *а*).

При создании электрического поля достаточно высокой напряженности носители заряда могут проходить через тонкую пленку оксида толщиной не более 5 нм и скапливаться на границе раздела слоев. После снятия электрического поля заряд остается захваченным приграничным слоем нитрида кремния и уже не рассасывается. Этот заряд и является носителем информации, хранимой МНОП-транзистором. В зависимости от направленности электрического поля можно создавать заряд любого знака. Для МНОП-транзистора с *n*-каналом отрицательный заряд на границе раздела слоев повышает пороговое напряжение (компенсирует воздействие положительного напряжения на затворе, отпирающего транзистор). В результате рабочие напряжения на затворе транзистора не в состоянии его открыть.

При программировании ЗУ используют относительно высокие напряжения, около 20 В. После снятия напряжения прохождение носителей заряда через диэлектрик прекращается и заданное транзистору пороговое напряжение остается неизменным. ЗУ на основе МНОП-транзисторов может хранить записанную информацию десятками лет.

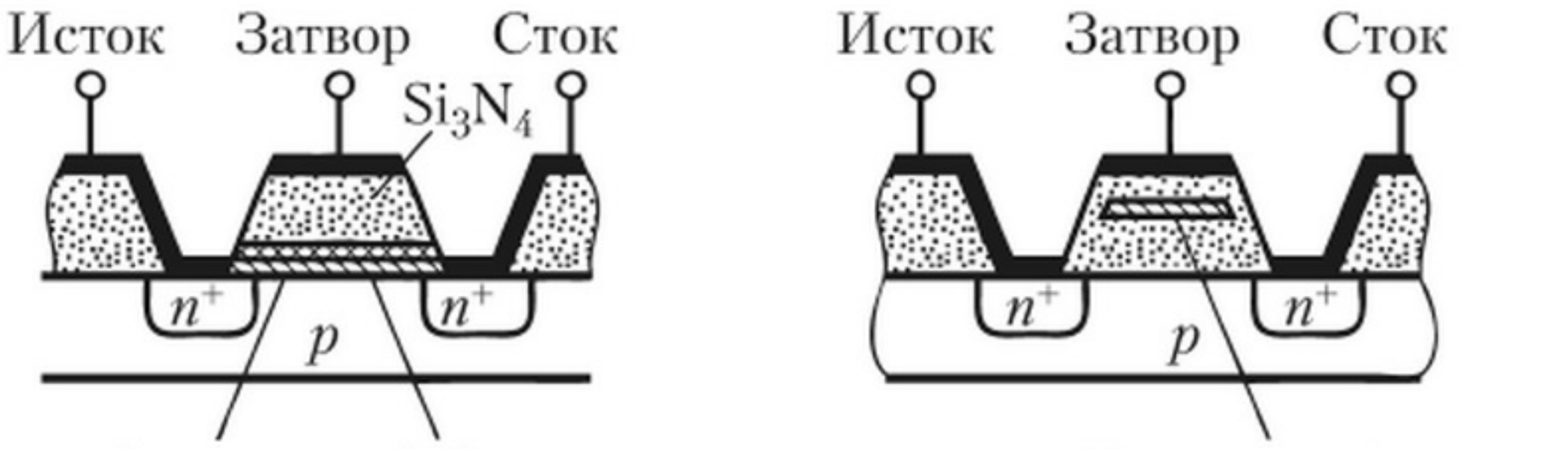


Рис. 4.6. **Структура МНОП-транзистора (*а*) и ЛИЗМОП-транзистора (*б*)**

Чтобы стереть записанную информацию, необходимо удалить носители заряда из приграничной области. Для этого подается напряжение, создающее электрическое поле противоположной направленности, и носители заряда через слой SiO_2 возвращаются в полупроводник.

Транзисторы типа ЛИЗМОП (рис. 4.6, *б*) всегда имеют так называемый плавающий затвор, который может быть единственным или вторым, дополнительным к обычному управляющему затвору. Здесь так же, как и в МНОП-транзисторе, в область затвора можно вводить заряд. Только область введения заряда представляет собой не границу раздела слоев диэлектрика, а окруженную со всех сторон диэлектриком проводящую область, называемую плавающим затвором. Введенный в нее, как в ловушку, заряд также может сохраняться в течение очень долгого времени. При подаче на управляющий затвор, исток и сток импульса положительного напряжения относительно большой амплитуды 20÷25 В в обратно смещенных *p-n*-переходах возникает лавинный электрический пробой. Часть электронов, имеющих энергию, достаточную для преодоления диэлектрической области, проникает — *инжектируется* — в плавающий затвор. Снятие высокого программирующего напряжения восстанавливает обычное состояние областей транзистора и запирает электроны в плавающем затворе.

«Заряженный» транзистор подобно МНОП-транзистору остается всегда запертым. При отсутствии на плавающем затворе заряда транзистор работает в обычном ключевом режиме. Таким образом, в исходном состоянии все ЗЭ, выполненные на основе ЛИЗМОП-транзисторов, хранят лог. 1, а при программировании в них записываются лог. 0.

Информация может стираться двумя способами — ультрафиолетовым облучением или электрическими сигналами.

В первом случае корпус ИС имеет специальное прозрачное окошко для облучения кристалла. Эти лучи вызывают в областях транзистора фототоки и тепловые токи, что делает области прибора проводящими и позволяет заряду покинуть плавающий затвор. Операция стирания информации этим способом занимает десятки минут, информация стирается сразу во всем кристалле.

Воздействие лучей приводит к постепенному изменению свойств полупроводниковых материалов, поэтому число циклов перепрограммирования ограничено до 10÷100.

Электрическое стирание информации осуществляется подачей на управляющие затворы низкого, а на стоки — высокого напряжения. При этом можно стирать информацию не со всего кристалла, а выборочно. Длительность процесса стирания значительно меньше, поэтому циклов перепрограммирования может быть гораздо больше — $10^4 \div 10^6$. В то же время схемы с электрическим стиранием занимают больше места на кристалле, в связи с чем уровень их интеграции меньше, а стоимость выше.

Структура матрицы ЗЭ на основе ЛИЗМОП-транзисторов аналогична структуре, изображенной на рис. 4.4, *б*. Выборка строки матрицы осуществляется подачей импульсов в линии выборки, при этом «заряженные» транзисторы останутся запертыми, что соответствует лог. 0, а «незаряженные» — откроются, что соответствует лог. 1.

4.2.4. Флэш-память

Разработка флэш-памяти явилась наивысшим достижением развития схемотехники полупроводниковых программируемых ЗУ. По типу ЗЭ флэш-память подобна памяти типа EEPROM с электрическим стиранием, но имеет целый ряд особенностей. В схемах флэш-памяти не предусмотрено стирание отдельных слов, стирание информации осуществляется либо для всей памяти одновременно, либо для достаточно больших блоков.

Одновременное стирание всей информации ЗУ реализуется наиболее просто, но имеет один недостаток. При замене даже одного слова необходимо перезаписывать всю информацию. Поэтому наряду со схемами с одновременным стиранием всей информации имеются схемы с блочной структурой с размером блока от 256 байт до 128 Кбайт.

Двумя основными направлениями эффективного использования флэш-памяти являются хранение редко изменяемых данных и замена памяти на магнитных дисках. Для первого направления используется *флэш-память* с *адресным доступом*, а для второго — *файловая память*.

Флэш-память с адресным доступом. Микросхемы этого типа могут осуществлять одновременное стирание как всей информации, так и отдельных блоков. Запись и стирание информации осуществляет не программатор, а сам процессор вычислительного устройства в обычном рабочем

режиме. Для этого флэш-память имеет дополнительное управление словами-командами, записываемыми процессором в специальный регистр микросхемы. При подаче специального напряжения программирования схема обеспечивает запись и стирание информации, а при его отсутствии работает как обычная микросхема ПЗУ. Перед программированием процессор считывает из микросхемы код — идентификатор, содержащий код фирмы-изготовителя и микросхемы. Эти сведения позволяют согласовать алгоритмы стирания и записи, что производится автоматически.

В режиме программирования работа микросхемы осуществляется под управлением внутреннего автомата, который управляет схемами стирания и программирования памяти в соответствии с кодом команды. По команде стирания стираются все байты памяти или выбранного блока, после чего все они должны быть проверены. Если не все байты стерты, выполняются повторное стирание и проверка. Программирование памяти ведется байт за байтом. При этом проверяется записанная информация. Процессор считывает из ЗУ записанный байт и сравнивает его с исходным.

В схемах с блочным стиранием размер блоков разный. Один из блоков предназначен для хранения программного обеспечения базовой системы ввода/вывода микропроцессорной системы BIOS, необходимой для правильной эксплуатации системы, и аппаратно защищен от случайного стирания. В ЗУ есть также блоки параметров и главные блоки, не защищенные от случайного стирания. Главные блоки хранят основные управляющие программы, а блоки параметров — относительно часто меняемые параметры системы.

Файловая флэш-память. В течение многих лет хранение больших объемов данных возлагалось в компьютерах на хорошо отработанные и сравнительно недорогие внешние ЗУ на магнитных дисках. При этом система памяти компьютера организована как сочетание жесткого магнитного диска (винчестера), хранящего все программное обеспечение, с динамическим полупроводниковым ОЗУ, в которое те или иные программы переписываются по мере их использования. Винчестер, как электромеханическое устройство, чувствителен к ударам, вибрациям, имеет ограниченное быстродействие, связанное с необходимостью

перемещения блока магнитных головок, и значительное потребление мощности для вращения дисков.

Файловая флэш-память применяется для замены жестких дисков, а также в съемных флэш-накопителях («флэшках»), флэш-картах сотовых телефонов и цифровых фотоаппаратов. По сравнению с винчестером файловая флэш-память в сотни раз сокращает потребляемую мощность, повышает надежность, уменьшает размеры и вес ЗУ, на несколько порядков повышает быстродействие при чтении данных. Действительно, питание для полупроводниковой памяти можно подавать лишь тогда, когда к ней идет обращение, а для доступа к блоку данных достаточно вычислить адрес. Твердотельные диски — SSD флэш-накопители (Solid-State Drive) — создаются на базе чипов флэш-памяти, емкость которых к 2010 г. достигала уже 128 Гбит в одном чипе. SSD флэш-накопитель, собранный из множества чипов, может иметь емкость в сотни гигабайт.

4.3. Запоминающие устройства для хранения оперативной информации

4.3.1. Статические ЗУ

Область применения относительно дорогостоящих статических ЗУ в системах обработки информации определяется их высоким быстродействием. В частности, они широко применяются в кэш-памяти, которая при сравнительно малой емкости должна иметь максимальное быстродействие.

Статические ОЗУ, как правило, имеют структуру 2*DM*. В качестве ЗЭ в них используются триггеры с цепями установки и сброса (рис. 4.7).

Такой триггер в КМОП-технологии содержит 6 транзисторов. На четырех выполняется сам ЗЭ, а два служат для выборки ЗЭ и чтения/записи. Комплементарные пары транзисторов *VT*1, *VT*2 и *VT*3, *VT*4 работают противофазно: если открыт один транзистор, другой — закрыт. Они составляют плечи триггера, которые также работают противофазно. Пусть, например, в левом плече транзистор *VT*1 открыт. Тогда высокий потенциал от $+U_{CC}$ через *VT*1 подается на затворы транзисторов *VT*3 и *VT*4. При этом транзистор *VT*3 запирается, а *VT*4 — открывается. Низкий потенциал корпуса через открытый *VT*4 подается

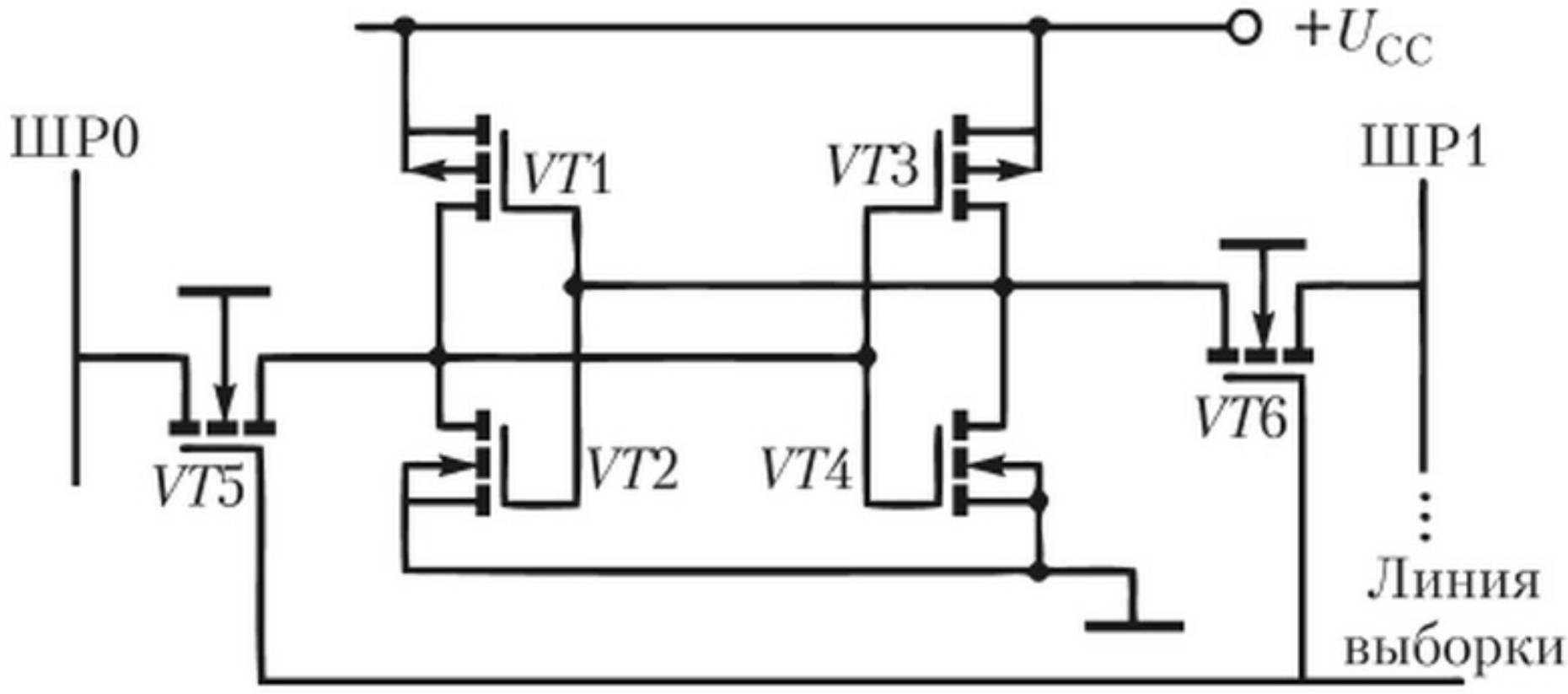


Рис. 4.7. **Схема триггерного запоминающего элемента статических ЗУ**

на затворы транзисторов *VT*1 и *VT*2 и поддерживает открытое состояние *VT*1 и закрытое *VT*2. Если же транзистор *VT*3 будет открыт, то будут закрыты транзисторы *VT*1 и *VT*4, а транзистор *VT*2 — открыт, поддерживая открытое состояние транзистора *VT*3. Таким образом, ЗЭ может находиться только в двух устойчивых состояниях. Если открыт транзистор *VT*1, то ЗЭ хранит лог. 1, если же открыт транзистор *VT*3 — то лог. 0. Транзисторы *VT*5 и *VT*6 своими затворами подключены к линии выборки и открываются при подаче в нее высокого потенциала. При открытии транзисторов *VT*5 и *VT*6 потенциалы с плечей ЗЭ поступают на разрядные шины ШР1 и ШР0.

Для того чтобы перевести ЗЭ в противоположное состояние, на шину ШР1 или ШР0 подается потенциал лог. «0», а в линию выборки — высокий потенциал, открывающий транзисторы *VT*5 и *VT*6. Пусть, например, ЗЭ хранит «1», т.е. *VT*1 и *VT*4 — открыты, а *VT*2 и *VT*3 — закрыты. Подача лог. 0 от шины ШР0 через открытый *VT*5 на затворы транзисторов *VT*3 и *VT*4 приводит к запиранию транзистора *VT*4 и отпиранию *VT*3. В результате высокий уровень напряжения +U_{CC}, поступивший через *VT*3 на затворы *VT*1 и *VT*2, закрывает *VT*1 и отпирает *VT*2. Таким образом, триггер переходит в противоположное состояние, соответствующее хранению лог. 0.

Статические ОЗУ энергозависимы — при снятии питания информация в триггерных ЗЭ теряется. Можно придать им энергонезависимость с помощью резервного источника питания. Это наиболее пригодно для ЗУ на КМОП-элементах, так как они в режиме хранения потребляют чрезвычайно малую мощность.

Как известно, ИС, выполненные по КМОП-технологи, не способны работать на емкостную нагрузку (см. параграф 3.4), поэтому одним из направлений развития статических ЗУ является совмещение КМОП и биполярной технологий в одной ИС. В таких микросхемах матрица ЗЭ выполняется по КМОП-технологии, а выходные каскады — по биполярной технологии, обеспечивающей более высокое быстродействие микросхемы и эффективную работу на емкостную нагрузку.

4.3.2. Динамические ЗУ

В динамических ЗУ данные хранятся в виде зарядов емкостей МОП-структур, и основой ЗЭ является просто конденсатор небольшой емкости. Такой ЗЭ значительно проще триггерного, содержащего 6 транзисторов, что позволяет разместить на кристалле намного больше ЗЭ. В то же время конденсатор со временем неизбежно теряет свой заряд, и хранение данных требует их периодической регенерации (несколько миллисекунд).

Электрическая схема однотранзисторного ЗЭ показана на рис. 4.8, *а*. Запоминающий конденсатор $C_з$ образуется между стоком транзистора, не имеющего внешнего вывода, и подложкой, т.е. корпусом. Лог. 1 соответствует заряженной емкости, а лог. 0 — разряженной. Транзистор отклю-

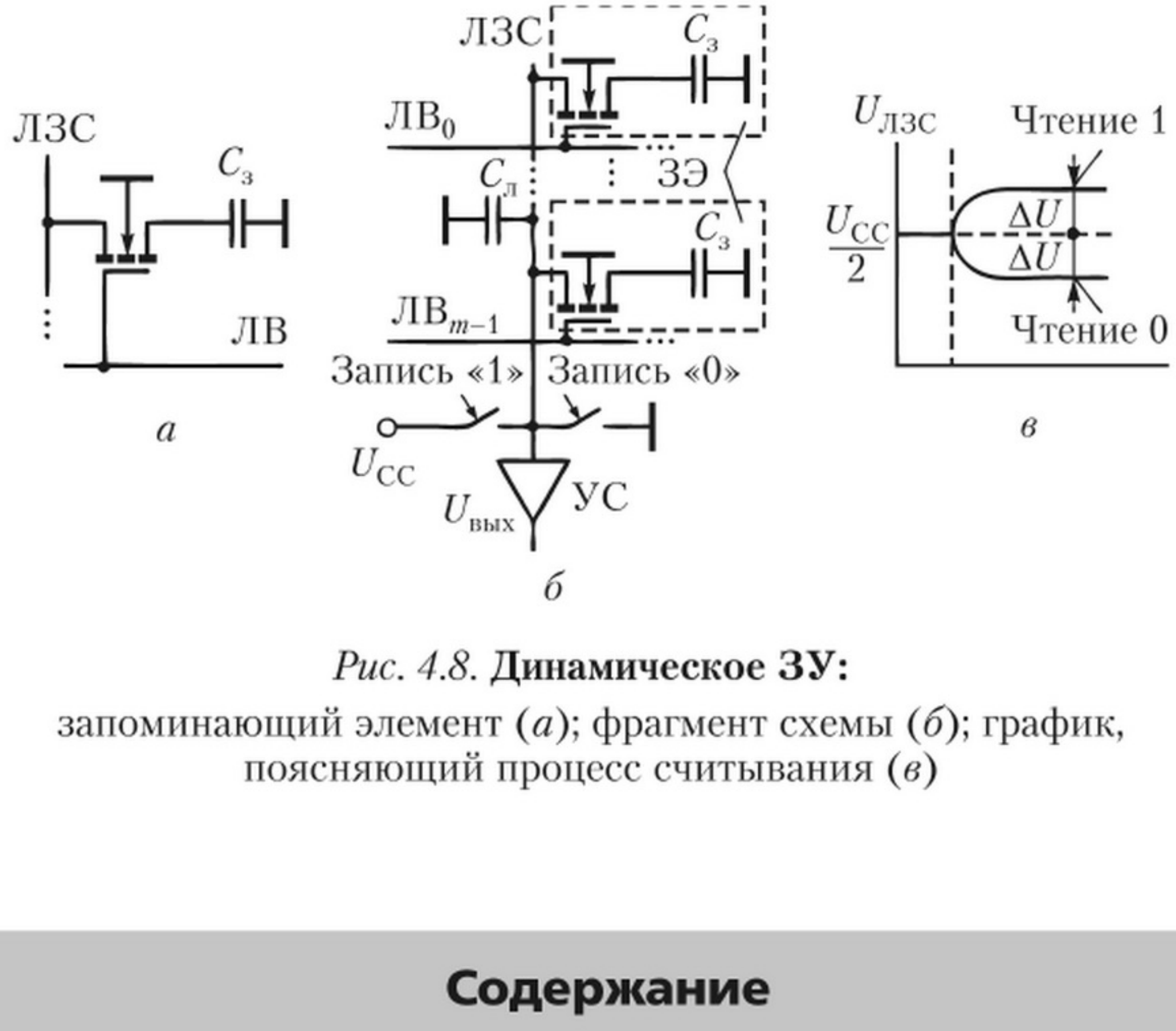


Рис. 4.8. **Динамическое ЗУ:**
запоминающий элемент (*а*); фрагмент схемы (*б*); график, поясняющий процесс считывания (*в*)

чает запоминающий конденсатор от линии записи/считывания (ЛЗС) или подключает его к ней.

ЗЭ образуют матрицу, один столбец которой показан на рис. 4.8, *б*. Ключевые транзисторы своими затворами подключены к линиям выборки (ЛВ), которые образуют строки матрицы ЗЭ. Истоки транзисторов, которые образуют столбцы матрицы, подключены к ЛЗС. Таким образом, ЗЭ включаются на пересечении строк и столбцов.

В режиме хранения ключевой транзистор заперт. При записи на затвор транзистора выбранного ЗЭ подается отпирающее напряжение, транзистор открывается, и через один из записывающих ключей емкость $C_з$ либо заряжается до напряжения U_{CC} (лог. 1), либо разряжается на корпус (лог. 0). При чтении транзистор открывается и емкость подключается к ЛЗС. При этом в зависимости от того, заряжена или разряжена емкость, потенциал ЛЗС изменяется по-разному. ЛЗС сама имеет некоторую емкость $C_л$, которая перед подключением к ней ЗЭ заряжается до потенциала, равного половине уровня питающего напряжения (рис. 4.8, *в*). Если емкость ЗЭ была заряжена (лог. 1), то часть заряда перетекает в ЛЗС и ее потенциал повышается. Если же ЗЭ был разряжен (лог. 0), то часть заряда с ЛЗС перетекает в емкость ЗЭ и потенциал ЛЗС понижается. Усилитель считывания УС воспринимает потенциал ЛЗС и формирует выходной сигнал.

Как видно из описанных процессов, заряд ЗЭ при чтении изменяется, поэтому после чтения его необходимо восстановить. Для этого в микросхеме имеются специальные усилители-регенераторы, которые автоматически по остаточному заряду $C_з$ определяют, каков был заряд ЗЭ до чтения. Если ЗЭ хранил лог. 1, то на ЛЗС устанавливается высокий уровень, и $C_з$ дозаряжается до единичного потенциала, если же ЗЭ хранил лог. 0, на ЛЗС устанавливается нулевой уровень и $C_з$ полностью разряжается.

Усилители-регенераторы обеспечивают восстановление данных не только после операции чтения, но и в режиме периодической регенерации данных всех ЗЭ. В каждом столбце матрицы ЗЭ есть свой усилитель-регенератор, поэтому регенерация осуществляется построчно и одновременно для всех ЗЭ одной строки. Режим строчной регенерации осуществляется с помощью циклов чтения по всем строкам матрицы ЗЭ. При этом процесс не сопровождается выдачей данных на выходные буферы, а целиком проходит

внутри ЗУ. Используются только адреса строк, а адреса столбцов не требуются.

Модуль памяти на основе динамических ЗУ составлен из одноразрядных микросхем, число которых равно разрядности хранимых слов. Относительно входных сигналов все микросхемы включены параллельно. В рабочем режиме модулем управляет процессор, а в режиме регенерации — специальный контроллер. В некоторых ЗУ схемы регенерации данных реализованы на самом кристалле памяти и контроллер регенерации не нужен. Такие ЗУ называют *квазистатическими*.

Современные микропроцессоры характеризуются высоким быстродействием. Это требует увеличения скорости работы ОЗУ, обменивающихся информацией с процессорами. Особенно остро эта задача стоит перед разработчиками динамических ОЗУ, которые благодаря максимальной информационной емкости и низкой стоимости занимают ведущее место в составе памяти компьютеров. В последнее время предложен ряд вариантов динамических ОЗУ повышенного быстродействия. Методы, использованные в этих ОЗУ, основаны на предположении о *кучности адресов* при обращениях к ОЗУ. Действительно, команды программы обычно располагаются в ОЗУ компактно, команда за командой. Точно так же располагаются и массивы данных.

Архитектура, технология и схемотехника полупроводниковых ЗУ постоянно развиваются. Поколения динамических ЗУ сменяются приблизительно каждые пять лет. В 1990 г. доминировали ЗУ емкостью 1 Мбит, в начале этого века в основном использовались микросхемы ЗУ с емкостью 64 и 128 Мбит. К 2010 г. ведущие фирмы-производители микросхем памяти освоили выпуск чипов емкостью в 4 Гбита. Число транзисторов в них стало составлять уже более 4 млрд. Это позволило создавать модули оперативной памяти емкостью в 4 Гбайт при установке чипов на одной стороне модуля памяти и 8 Гбайт — при двухсторонней.

4.4. Микропроцессоры

Термин «микропроцессор» появился в начале 70-х гг. XX в. и стал настолько популярным, что сегодня трудно встретить человека, который с ним не знаком. Микропро-

цессоры являются основой любого компьютера, на них основана работа разнообразных систем обработки информации, систем управления техническими средствами, технологическими процессами. Можно сказать, что именно микропроцессоры обеспечили проникновение вычислительной техники в самые разнообразные сферы жизни.

Микропроцессор (МП) — программно-управляемое устройство, осуществляющее процесс обработки цифровой информации и управление им, построенное на одной или нескольких БИС/СБИС.

Основное отличие МП от всех ранее рассмотренных цифровых устройств заключается в способе обработки информации. В обычных цифровых устройствах — последовательностных или комбинационных — обработка входных сигналов осуществляется аппаратно, и результат определяется схемой соединения различных элементарных узлов — конъюнкторов, дизъюнкторов, триггеров и т.д. В МП процесс обработки информации осуществляется *программно*, т.е. путем последовательного выполнения элементарных действий — *команд программы*, и результат обработки определяется этой программой. Решаемая задача, таким образом, задается программой, а сами аппаратные средства (микропроцессор и набор дополнительных ИС) остаются неизменными. Это определяет универсальность устройств и систем на основе МП.

МП появились, когда уровень развития микроэлектроники позволил в одной ИС разместить все узлы, необходимые для работы устройств программной реализации алгоритмов. Такие устройства назывались процессорами. Процессоры ЭВМ, выполненные на транзисторах, а затем и на элементах малой и средней степени интеграции, представляли собой целые шкафы, начиненные платами. БИС, выполняющие все функции процессора, пусть и над словами малой разрядности, получили название «микропроцессоры», а электронные системы, главным вычислительно-управляющим узлом которых является МП, стали называться *микропроцессорными системами*.

В составе микропроцессорной системы всегда можно выделить *микроЭВМ* и набор *устройств ввода/вывода* информации (поскольку они находятся за пределами микроЭВМ их называют *внешними устройствами* (ВУ)). Кроме микропроцессора в микроЭВМ входят также память

и устройства, непосредственно взаимодействующие с внешними устройствами. Набор микросхем, пригодных для совместного применения в составе микроЭВМ, называют микропроцессорным комплектом БИС/СБИС (МПК). Микросхемы, входящие в МПК, могут быть выполнены по различным технологиям, но они должны быть совместимы по архитектуре, электрическим параметрам, конструктивным признакам.

Приведенное выше классическое определение микропроцессора было сформулировано, когда микропроцессорная техника только начинала развиваться. При этом сразу наметились два направления. Первое — разработка однокристальных МП, второе — многокристальных. В однокристальных МП все элементы процессора размещались в одной БИС, при этом разрядность обрабатываемых слов составляла 1 байт, а система команд была раз и навсегда заданной. Эти МП выполнялись по МОП-технологии, которая в то время не могла обеспечить высокого быстродействия. Более быстродействующими были ИС на основе ТТЛШ-технологии, но уровень интеграции в то время не позволял в одной ИС совместить все узлы процессора. Тогда схема процессора была разделена на части — секции небольшой (2 бита) разрядности, осуществляющие более элементарные (чем команды) действия — микрокоманды. Каждая секция выполнялась в виде отдельной БИС. Путем последовательного их соединения обеспечивалось наращивание разрядности обрабатываемых слов. Для выработки команд, складывающихся из последовательности микрокоманд, разрабатывались другие БИС — блоки микропрограммного управления. Таким образом, чтобы собрать схему процессора, требовалось несколько БИС. Такие МП назывались многокристальными.

В настоящее время уровень развития КМОП-технологии стал настолько высок, что однокристальные МП практически вытеснили многокристальные из большинства областей применения, оставив за ними узкую область специализированной вычислительной техники.

4.4.1. Структура и принципы работы микропроцессорной системы

Структура любой микропроцессорной системы является *магистрально-модульной*. Это означает, что в ней можно выделить набор модулей — устройств, подключенных

к общим магистралям, называемых шинами. Под *шиной* понимают набор линий связи, по которым передается информация определенного типа, осуществляется обмен информацией между различными модулями системы.

Обобщенная структура микропроцессорной системы представлена на рис. 4.9.

Любую микропроцессорную систему можно представить как микроЭВМ и набор ВУ. Под ВУ понимают устройства двух типов:

- устройства ввода/вывода информации, обеспечивающие вычислительный процесс и связь с оператором (монитор, клавиатура, внешние запоминающие устройства и т.д.);
- устройства, обеспечивающие управление техническими средствами технологического оборудования, станками и т.п.

Кроме МП, который также называют *центральным процессорным элементом*, в состав микроЭВМ входят ПЗУ, ОЗУ и *блок интерфейса*. ПЗУ обеспечивает хранение неизменяемых программ работы системы. Если это универсальная система типа персонального компьютера, то в ПЗУ хранится программа базовой системы ввода/вывода, обеспечивающая функционирование и начальную загрузку системы — инициализацию. Если это специализированная система, типа устройства числового программного управ-

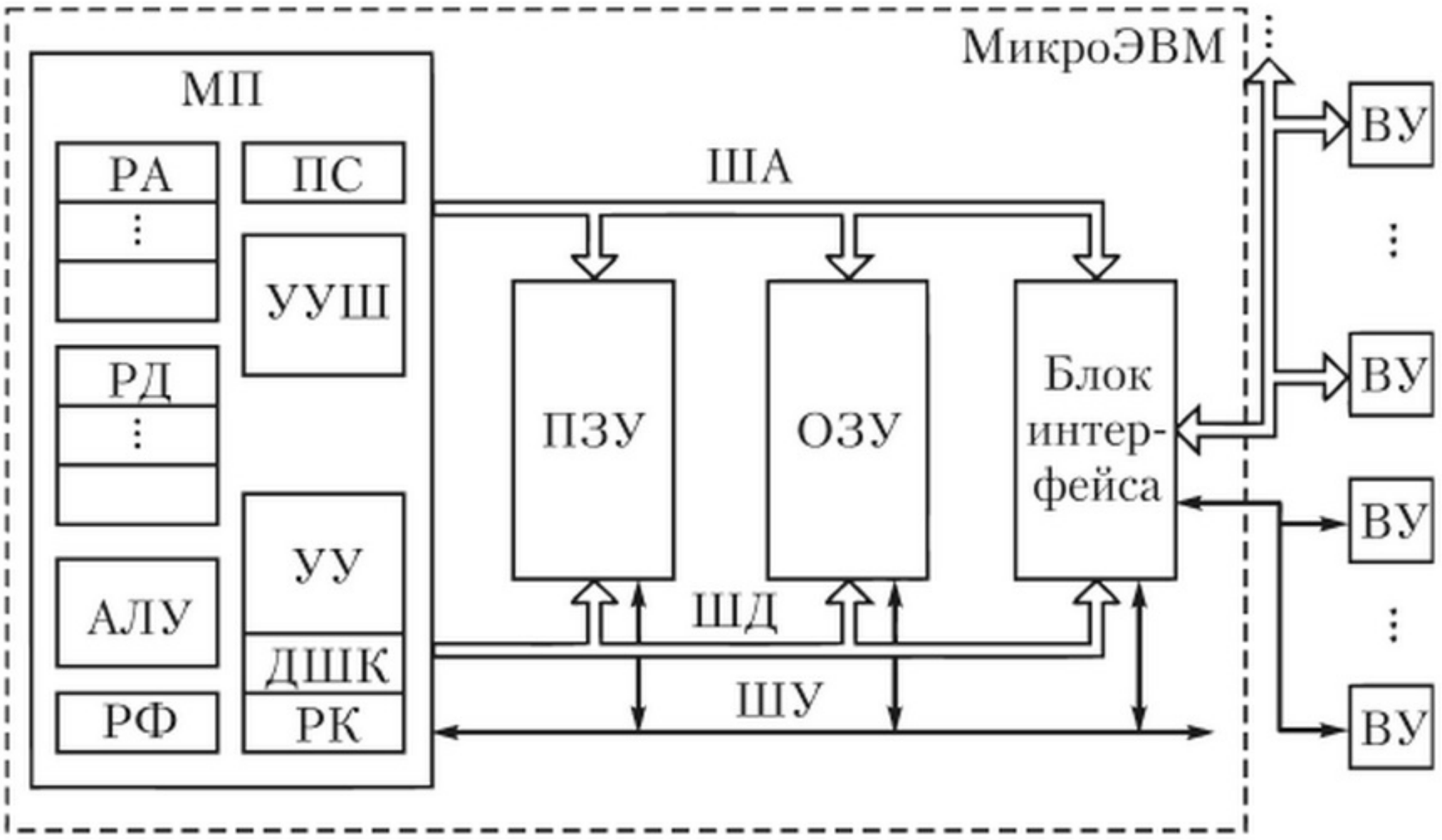


Рис. 4.9. **Структура микропроцессорной системы**

ления, то в ПЗУ заносится все программное обеспечение системы. ОЗУ предназначено для хранения информации, которая может изменяться в процессе работы системы. Это могут быть данные, промежуточные результаты вычислений и программы, исполняемые в текущий момент времени. В простых системах это только входная информация и промежуточные результаты.

Весь обмен информацией МП с ВУ осуществляется через блок интерфейса. ВУ передают данные из внешней среды в МП или ОЗУ или получают их из микроЭВМ. Для подключения ВУ к микропроцессорной системе его сигналы, скорость передачи информации, формат слов необходимо привести к стандартному виду, с которым работает МП. Все эти преобразования данных выполняются в интерфейсном блоке. Фактически блок интерфейса это набор различных узлов — *адаптеров* и *контроллеров*. Сложные ВУ, типа монитора или накопителей на магнитных дисках подключаются через контроллеры ВУ, которые обеспечивают не только преобразование данных, но и управление самими ВУ. Они на структурной схеме не показаны.

Взаимодействие узлов микроЭВМ между собой осуществляется с помощью трех шин: *шины адреса* (ША), *шины данных* (ШД) и *шины управления* (ШУ). Чтобы МП мог однозначно выбрать нужную ячейку памяти или регистр ВУ, они имеют адреса. Адрес ячейки (регистра) передается от МП в память или интерфейсный блок по ША. ША однонаправленная, так как направление передачи информации по ней только одно — из МП. В отличие от нее ШД является двунаправленной, так как передача данных по ней осуществляется как из МП в память и интерфейс, так и наоборот. ША и ШД состоят из параллельных линий, передача информации по которым осуществляется одновременно для всех линий (поэтому на рисунке эти шины обозначены широкими стрелками). Число линий ШД определяется разрядностью МП, а ША — объемом памяти, т.е. разрядностью двоичного кода, необходимого для адресации всех ячеек. ШУ состоит из отдельных линий, по которым передаются те или иные управляющие сигналы. Естественно, что они передаются не одновременно, поэтому на рис. 4.9 ШУ обозначена узкими стрелками. В основном это сигналы, передаваемые из МП в остальные узлы, но некоторые имеют обратную направленность — в МП. Примером первых могут служить сигналы чтения и записи, указыва-

ющие, какую именно следует выполнять операцию с ячейкой, адрес которой выставлен на ША. Ко вторым относят осведомительные сигналы запроса обслуживания, поступающие от ВУ, а также сигнал сброса МП в начальное (нулевое) состояние.

Внешние устройства в зависимости от способа передачи информации разделяются на две большие группы: устройства, обменивающиеся параллельными словами данных (на рис. 4.9 они подключены к параллельной шине), и устройства, обменивающиеся информацией в последовательном коде, т.е. последовательно, бит за битом (подключены к однопроводной шине, обозначенной узкой стрелкой).

Основными узлами МП являются *устройство управления* (УУ), *регистр команд* (РК), *дешифратор команд* (ДШК), *арифметико-логическое устройство* (АЛУ), *регистр флажков* (РФ), набор *внутренних регистров*, разделяемых на *адресные регистры* (РА) и *регистры данных* (РД), *программный счетчик* (ПС), *устройство управления шинами* (УУШ).

Координация работы всех узлов в соответствии с выполняемой командой осуществляется тремя узлами: УУ, РК и ДШК. РК обеспечивает хранение команды в течение всего цикла ее исполнения, а ДШК выполняет расшифровку кода этой команды. УУ вырабатывает серию импульсов, обеспечивающих последовательное и слаженное срабатывание узлов МП в соответствии с выполняемой командой. Для выработки управляющих импульсов на вход УУ поступают импульсы синхронизации от внешнего генератора. Такой генератор может быть также встроен в УУ. Кроме управления внутренними узлами, УУ обеспечивает прием и выдачу внешних управляющих сигналов.

АЛУ обеспечивает выполнение всех операций, с помощью которых осуществляется переработка данных в МП. Оно может выполнять несложные арифметические, логические и сдвиговые операции. Количество *операндов*, т.е. двоичных чисел, над которыми выполняются действия в АЛУ, может колебаться от одного до двух. Например, при инвертировании (логическое НЕ) АЛУ достаточно одного операнда, а для операции сложения двух чисел необходимо два операнда. Перечень операций, выполняемых АЛУ, зависит от типа МП. Для большинства МП в АЛУ выполняются следующие операции: сложение, вычитание, логические

И, ИЛИ, НЕ, исключающее ИЛИ (сумма по модулю 2), сдвиг вправо, сдвиг влево, сложение с единицей (инкремент), вычитание единицы (декремент). Сложные арифметические операции, такие как умножение и деление, АЛУ не выполняет. В зависимости от результата операции АЛУ формирует признаки результата, называемые *флажками*. Эти признаки используются не в текущей, а в последующих командах, поэтому для их хранения в МП используется РФ.

Регистры — составная и очень важная часть МП. Каждый регистр МП можно использовать для временного хранения одного слова данных. Некоторые регистры имеют специальное назначение, другие — многоцелевое. Внутренние РА и РД являются внутренней памятью МП. РА используются для временного хранения двоичных чисел, с помощью которых МП вычисляет адреса ячеек памяти, к которым он обращается в процессе работы. РД используются как для непосредственного хранения операндов, так и для вычисления адресов ячеек ОЗУ, хранящих операнды. Через РД также осуществляется обмен информацией между МП и ВУ. Программный счетчик служит для хранения адреса ячейки памяти, в которой хранится очередная исполняемая команда программы.

Выполняя программу, МП обрабатывает команду за командой, которые обычно располагаются в ячейках памяти последовательно одна за другой. Команда задает выполняемую операцию и содержит сведения, где находятся операнды. Выполнение команды можно разбить на две фазы: фазу выборки команды и фазу ее исполнения. Первая фаза начинается с того, что МП выставляет на ША содержимое ПС, хранящего адрес ячейки памяти с очередной командой. Содержимое ячейки выставляется на ШД, МП считывает информацию с ШД и помещает команду в РК.

Вторая фаза заключается в собственно выполнении команды. При этом сначала МП должен подготовить операнды. Операнды могут храниться как в самом МП, так и в ОЗУ. В первом случае они хранятся в регистрах данных, и МП может переходить к непосредственному исполнению математической или логической операции в соответствии с кодом команды. Во втором случае МП должен сначала вычислить адрес ячейки ОЗУ, хранящей операнд, потом выставить этот адрес на ША и считать содержимое

указанной ячейки ОЗУ, и только затем выполнить операцию. Выполнение операции осуществляется в АЛУ, после чего результат должен быть помещен на место первого операнда. Если это один из внутренних регистров МП, результат сразу же переписывается в этот регистр, если это ячейка ОЗУ, требуется еще один цикл обращения к памяти. Таким образом время исполнения команды зависит от количества циклов обращения к памяти, и самыми короткими являются те команды, в которых операнды хранятся непосредственно в МП.

Во время выполнения команды при каждом обращении МП к памяти программ содержимое ПС автоматически увеличивается на единицу. Команды могут занимать не только одну ячейку памяти, а две и даже три, при этом, чтобы считать всю команду, МП должен несколько раз обратиться к памяти программ. В результате в конце выполнения команды в ПС уже хранится адрес следующей, и МП готов к выполнению очередной команды. Отсюда и название этого регистра — «программный счетчик».

Регистр ПС хранит адрес следующей выполняемой команды только в случае естественного порядка следования команд программы — команда за командой. В случае разветвления алгоритма в зависимости от выполнения или невыполнения заданного условия необходимо идти по одной из двух ветвей программы. Такие разветвления выполняются с помощью команд условного перехода. Для этого в команде условного перехода задается проверяемое условие и указывается адрес команды, подлежащей исполнению в случае выполнения условия. При невыполнении условия сохраняется естественный порядок следования команд, т.е. выполняется следующая по порядку команда. Так как адресация осуществляется через программный счетчик, то при выполнении заданного условия в ПС загружается адрес, указанный в команде, если же условие не выполняется, то адрес следующей команды оказывается уже сформированным в ПС. Проверка тех или иных условий в МП обычно заключается в анализе признаков результата, которые были сформированы при исполнении предыдущей команды и сохранены в регистре флажков.

В процессе работы МП постоянно обращается к ША и ШД. Передача информации внутри МП осуществляется по внутренним шинам, которые непосредственно не связаны с внешними шинами. Для передачи адресов и данных

из МП во внешние шины и приема данных с ШД в МП необходимо буферное устройство, которым служит УУШ. В простейшем случае — это набор буферных регистров, управляемых УУ. Буферный регистр адреса принимает данные с внутренней шины и хранит его в течение цикла обращения к памяти или ВУ, при этом адрес через выходные каскады регистра выставляется на ША. Буферный регистр данных — двунаправленный и может как передавать данные с внутренней шины во внешнюю, так и принимать их с внешней ШД и передавать во внутреннюю. Эти регистры имеют третье состояние и переводятся в него, когда МП с ША и ШД не работает. В более сложных МП в состав УУШ, помимо буферных, входит набор внутренних регистров, некоторые адресные регистры и комбинационные схемы. Такое УУШ работает самостоятельно, обеспечивая взаимодействие МП с внешними шинами.

4.4.2. Режимы обмена в микропроцессорной системе

В процессе работы в микропроцессорной системе постоянно происходит обмен информацией между МП и внешними устройствами. При этом возможны три режима обмена:

- программно-управляемый обмен;
- обмен по инициативе ВУ с прерыванием работы МП;
- обмен между ВУ и памятью в режиме прямого доступа к памяти.

Наиболее простой — *программно-управляемый обмен*. В этом режиме взаимодействие МП с ВУ осуществляется по инициативе МП в процессе выполнения основной программы. Имеются ВУ двух типов: всегда готовые к обмену и не всегда готовые к обмену. В первом случае обмен осуществляется без задержки. Во втором — МП вынужден ожидать готовности устройства к обмену. При этом программа не выполняется, МП ждет появления сигнала готовности ВУ к обмену, и только после его появления выполняет операцию обмена. Так как МП в это время не выполняет полезной работы, а его быстродействие гораздо выше быстродействия ВУ, такой режим обмена сопряжен с непроизводительными потерями времени.

Обмен по инициативе ВУ *с прерыванием работы* МП более производительный, так как время ожидания МП исключается. При своей готовности к обмену ВУ сооб-

щает об этом МП специальным *сигналом запроса прерывания*. Процессор завершает выполнение текущей команды программы, выдает *сигнал подтверждения прерывания* и переходит к выполнению подпрограммы обмена с ВУ. Эта подпрограмма носит название *подпрограммы обслуживания прерывания*. Чтобы начать выполнение подпрограммы достаточно адрес ее первой команды загрузить в ПС. Но в результате выполнения программы содержимое всех внутренних регистров МП уже не будет соответствовать тому, что было в момент прерывания. Поэтому возникает необходимость сохранения состояния МП на момент прерывания в оперативной памяти, чтобы можно было вернуться к прерванной основной программе. Для этого в микропроцессорных системах используется так называемая *стековая память*. Стек — это область памяти, запись в которую и считывание осуществляется по принципу «последний пришел — первый ушел». Процесс функционирования стека напоминает работу с пачкой документов, когда каждый новый документ кладется сверху пачки, т.е. самый первый по времени поступления документ оказывается в самом низу пачки и его рассмотрение происходит в последнюю очередь, в то время как последний документ рассматривается в первую очередь. Точно так же в стековую память осуществляется загрузка данных. При переходе от текущей программы к программе обслуживания прерывания содержимое некоторых адресных регистров МП заносится в стековую память автоматически. Запись содержимого остальных регистров должна быть предусмотрена в начале программы обслуживания прерывания с помощью специальных команд работы со стеком. При этом в конце программы должны стоять команды, обеспечивающие восстановление состояния регистров МП. Порядок, в котором осуществляется извлечение содержимого регистров из стековой памяти и запись его обратно в регистры, соответствует принципу «последний пришел — первый ушел».

Для работы со стековой памятью в МП есть специальный регистр — *указатель стека* (УС). Этот регистр всегда содержит адрес «верхней» (по аналогией со стопкой документов) загруженной ячейки стека, называемой вершиной стека. На рис. 4.10 показаны процессы записи и извлечения данных из стека.

При включении слова данных в стек происходит автоматическое уменьшение содержимого УС на единицу, а при

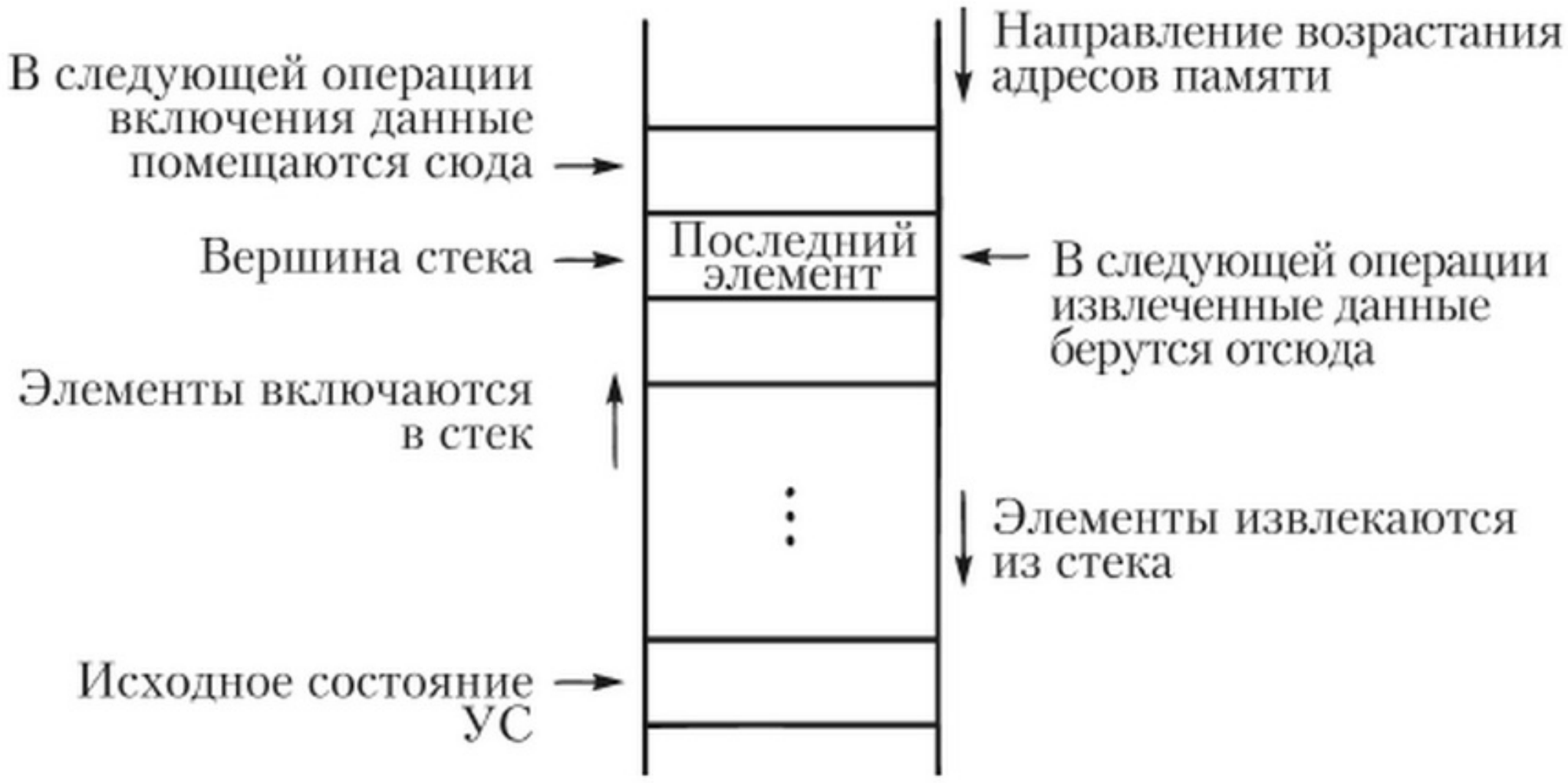


Рис. 4.10. **Процессы записи данных в стек и извлечения из стека**

извлечении — увеличение на единицу. Таким образом, при работе со стековой памятью МП не нужно вычислять адрес ячейки памяти, так как он всегда имеется в УС.

Причиной прерывания в микропроцессорных системах может быть не только готовность медленно действующего ВУ к обмену. Прерывание может быть вызвано действиями оператора, когда возникает необходимость вмешаться в ход управляемого процесса. В режиме прерывания осуществляется также работа МП с различными технологическими устройствами и системами в случае возникновения аварийных ситуаций.

Режим обмена внешнего устройства с памятью *в режиме прямого доступа к памяти* (ПДП) используется при передаче больших объемов информации. Например, в режиме ПДП осуществляется передача информации между ОЗУ и внешней памятью. Если бы такой обмен осуществлялся под управлением МП, то каждое слово данных сначала необходимо было прочитать из памяти и поместить его в МП, и только затем из МП передать приемнику — регистру ВУ. Такой обмен потребовал бы слишком больших непроизводительных затрат времени. В режиме ПДП происходит передача данных между памятью и ВУ без участия процессора. При этом МП отключается от шин адреса и данных и весь процесс обмена осуществляется под управлением специального контроллера ПДП. Происходит временный *захват системных шин* со стороны ВУ, памяти и контроллера ПДП. Режим ПДП начинается с того, что ВУ

сигнализирует процессору о своем «желании» начать обмен с памятью выдачей *сигнала запроса захвата шин*. Если процессор разрешает такой обмен, он выдает *сигнал разрешения захвата*, отключается от системных шин и передает управление контроллеру ПДП. В режиме ПДП выдача данных на ШД источником и считывание их с ШД приемником происходит в одном цикле записи/чтения. За счет устранения одного лишнего цикла записи/чтения значительно увеличивается темп передачи данных.

4.5. Однокристальный микропроцессор Intel 8086 (К1810 ВМ86)

В настоящее время однокристальные МП с фиксированной системой команд и фиксированной разрядностью самые распространенные в средствах вычислительной техники. В своем развитии они претерпели целый ряд схемотехнических и архитектурных усовершенствований, однако базовая структура и основополагающие принципы их работы остались прежними. Изучать микропроцессоры лучше всего на конкретном образце. В качестве такого образца был выбран МП Intel 8086 и его российский аналог К1810 ВМ86. Этот МП был разработан в конце 1970-х гг. Именно с его появлением началась интенсивная компьютеризация всех областей деятельности, включая науку, производство, медицину, образование и просто быт.

МП Intel 8086 был создан для использования в качестве процессора в персональных компьютерах. Этот МП — 16-разрядный, поэтому его вычислительные возможности стали совместимы с возможностями процессоров больших ЭВМ. В настоящее время в компьютерах используются гораздо более мощные МП, а сферой использования этого МП остаются средства автоматизации производственных процессов. Однако идеи, заложенные в нем, продолжали развиваться в последующих типах МП — 80186, 80286, 80386, 80486 (обратите внимание на две последние цифры каждого МП). МП *Pentium* также можно считать дальнейшим развитием МП 8086, и первоначально он должен был иметь шифр 80586. В некоторых литературных источниках все семейство МП *Pentium* также проходит под номерами 80...86.

4.5.1. Структура МП Intel 8086

Как известно, перед тем как МП начинает выполнять команду, он должен ее считать из памяти программ. В первых МП процессы чтения очередной команды и ее выполнения производились последовательно друг за другом. В МП Intel 8086 впервые был применен принцип совмещения во времени выполнения очередной команды с выборкой следующей. Этот принцип получил название *конвейерного* и был положен в основу работы всех последующих типов МП. Для этого МП (рис. 4.11) разделен на два сравнительно независимых устройства: *операционное устройство* (ОУ) и *шинный интерфейс* (ШИ).

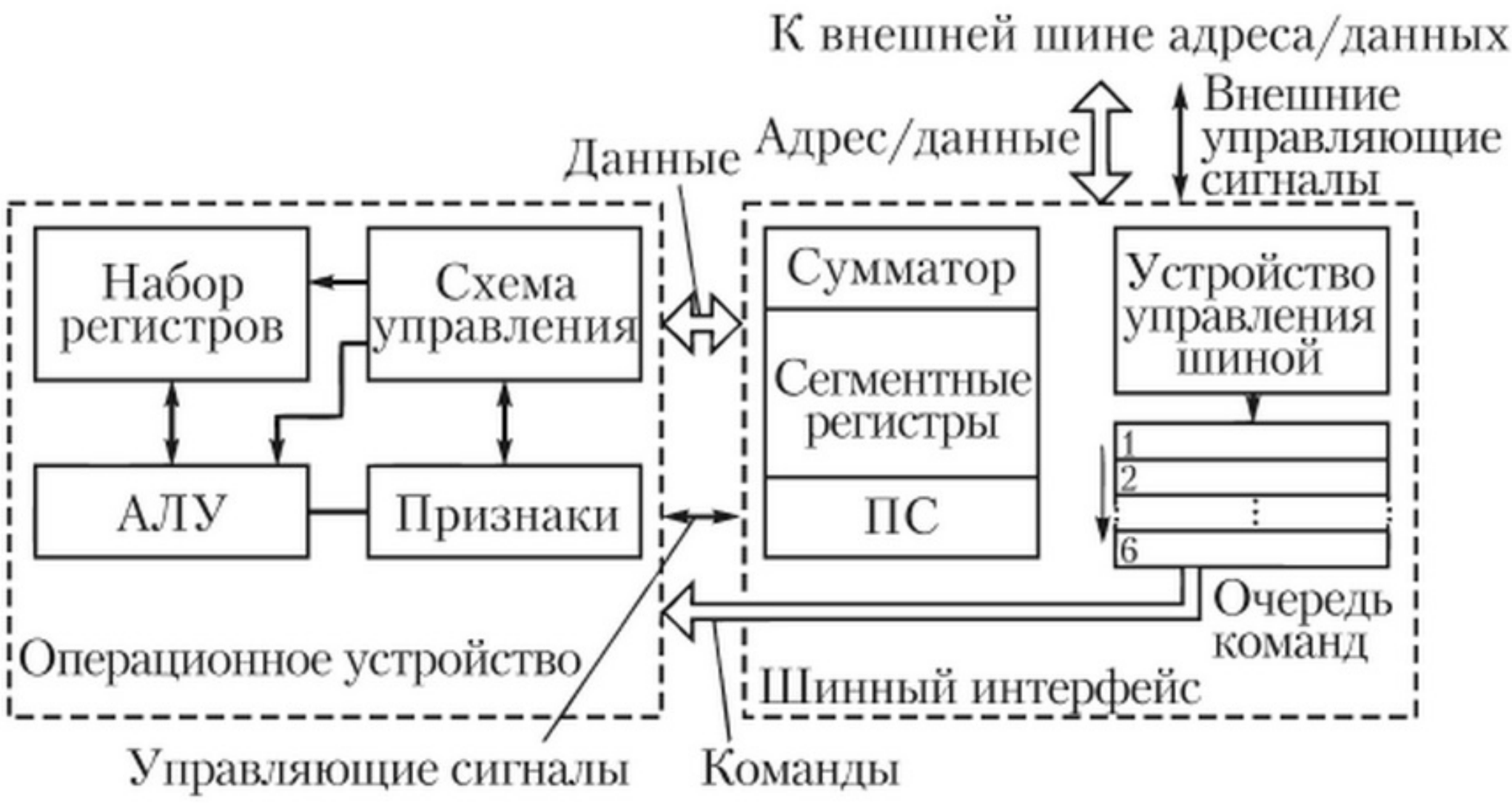


Рис. 4.11. **Структура микропроцессора Intel 8086**

ОУ осуществляет выполнение операций (математических, логических и т.д.), заданных командой. Шинный интерфейс выбирает команды из памяти программ, считывает операнды из памяти или ВУ, записывает результаты. Оба устройства работают параллельно, и пока ОУ выполняет операцию, ШИ подготавливает к выполнению следующую команду.

ОУ содержит набор 16-разрядных регистров, доступных при программировании микропроцессора, т.е. регистров для занесения или извлечения информации по командам программы. В состав ОУ входят регистры данных и адресные регистры (рис. 4.12). Регистры данных могут использоваться для хранения операндов и результатов операций. Каждый из них можно разделить и использовать как два независимых 8-разрядных регистра. В этом случае старшая



Рис. 4.12. **Внутренние регистры микропроцессора**

половина регистра обозначается буквой *H* (*high*), а младшая — *L* (*low*).

Каждый из регистров данных кроме арифметических функций имеет также и специальное назначение в некоторых командах. Так, через регистр *AX* (*accumulator* — аккумулятор) осуществляется передача и прием данных между микропроцессором и внешними устройствами в командах ввода/вывода. Регистр *BX* (*base* — база) может использоваться для хранения базового двоичного числа, используемого при вычислениях адресов ячеек ОЗУ. Регистр *CX* (*counter* — счетчик) в некоторых командах используется как счетчик. Регистр *DX* (*data* — данные) в некоторых командах ввода/вывода содержит адрес регистра внешнего устройства — порта ввода/вывода.

Адресные регистры могут использоваться только для хранения 16-разрядных двоичных чисел, используемых при вычислении адресов ячеек ОЗУ. Регистры *SI* и *DI* называют *индексными* (*index*), а *BP* и *SP* — *указательными* (*pointer* — указатель). Регистры *SI*, *DI* и *BP* используются во многих командах для адресации ячеек ОЗУ и являются универсальными адресными регистрами. Регистр *SP* — специальный регистр, предназначенный для работы со стеком (*stack pointer* — указатель стека).

В состав ОУ также входят АЛУ, буферные регистры, регистр состояния процессора и схема управления. АЛУ, как и во многих МП, обеспечивает выполнение арифметических, логических и сдвиговых операций, формируя при этом признаки результата. Буферные регистры использу-

ются для временного хранения операндов и результата операции во время выполнения команды. Эти регистры недоступны для пользователя. Регистр состояния процессора осуществляет запись признаков результата — флажков, а также хранение флажков условий, от которых зависит выполнение некоторых специальных функций.

Каждому признаку соответствует свой флажок. Весь набор различных условий в командах условного перехода формируется на основе анализа содержимого разрядов слова состояния процессора — *флажков*. Таких флажков всего шесть. Это флажок знака, соответствующего старшему биту результата (1 — отрицательный, 0 — положительный); флажок нуля — если результат равен нулю; флажок паритета — если в младших битах четное число единиц; флажок переноса — если возникает перенос (заем) из старшего бита; флажок вспомогательного переноса — если возникает перенос (заем) из третьего бита при выполнении операций в двоично-десятичной арифметике; флажок переполнения — если результат не помещается в установленном диапазоне.

Схема управления осуществляет дешифровку команды и формирует необходимые управляющие сигналы. ОУ изолировано от внешней шины, поэтому обмен данными и адресами между ОУ и ШИ осуществляется по внутренней шине.

ШИ выполняет все операции обмена данными с памятью и ВУ по требованию ОУ. ШИ содержит устройство управления шиной, набор адресных регистров, включая ПС и специальные *сегментные регистры*, сумматор и набор регистров, *называемых очередью команд*.

Устройство управления шинами обеспечивает взаимодействие МП с внешними шинами. Когда ОУ занято выполнением команд и не требует выборки операндов из памяти или ВУ, ШИ самостоятельно выбирает из памяти программ очередные команды. Эти команды хранятся во внутренней регистровой памяти ШИ — очереди команд (см. рис. 4.11). Длина очереди — 6 байт. По мере выполнения команд очередь движется и в ней появляется свободное место. При этом ШИ обращается к памяти и считывает очередную команду. Такой порядок работы действует только при отсутствии команд условного или безусловного перехода. Если необходимо перейти к новому участку программы, очередь сбрасывается, в ПС заносится соответствующий

адрес, ШИ выбирает эту команду и передает ее в ОУ. Пока ОУ выполняет команду, ШИ заполняет очередь следующими командами.

4.5.2. Мультиплексирование информационных линий

С повышением разрядности обрабатываемых слов и расширением объема памяти микропроцессору требуется все больше линий для передачи данных и адресов. Если для каждой линии выделять вывод микросхемы, потребуется чрезвычайно большое число выводов, которое часто просто невозможно сделать в корпусе ИС. Для того чтобы обходиться меньшим количеством выводов, в МП стали применять *мультиплексирование* линий ввода/вывода информации. Одним из первых МП, использующих мультиплексирование линий, стал МП Intel 8086. Так, для передачи адресов и данных в нем используется одна и та же 16-разрядная мультиплексная шина адреса/данных. В первый момент времени по шине передается адрес, а в следующий — данные. Адрес заносится во внешний буферный регистр и хранится там в течение всего цикла работы МП с памятью или ВУ. При этом выходы буферного регистра связаны с ША микроЭВМ и обеспечивают работу этой шины.

В МП Intel 8086 применено мультиплексирование и для некоторых других выводов ИС. Часть выводов имеет двойное назначение. В одних режимах они используются для передачи одной информации, а в других те же выводы несут другую информацию.

4.5.3. Сегментация памяти

Производительность микроЭВМ и микропроцессорной системы в целом во многом определяется емкостью ОЗУ. При этом МП должен иметь возможность адресации всего объема ОЗУ, т.е. вычисления адресов ячеек памяти и передачи их по ША в ОЗУ. Так, например, для адресации ОЗУ емкостью 1 Мбайт (2^{20} байт) требуется вычисление 20-разрядных двоичных чисел — адресов ячеек ОЗУ и передача их в ОЗУ по шине адреса.

В то же время архитектура МП и разрядность его регистров определяется длиной обрабатываемых слов и обычно она составляет число, кратное восьми — 1 байту. Так, МП Intel 8086, так же как и многие другие, обеспечивает обра-

ботку 16-разрядных слов и все его регистры, в том числе и адресные, 16-разрядные. Для расширения адресного пространства МП применяют механизм *сегментации памяти*. МП рассматривает весь объем памяти как набор сегментов памяти (рис. 4.13).

Сегмент — это часть памяти такого объема, адресация которого возможна с помощью одного регистра МП. Таким образом, разрядность регистров МП определяет размер сегментов памяти. Так, например, для 16-разрядных МП это 64 Кбайт, т.е. 2^{16} байт. Выделение сегмента осуществляется определением его начального адреса, или базы сегмента. В МП Intel 8086 для выделения сегментов имеется четыре специальных *сегментных регистра* — *CS*, *DS*, *SS*, *ES*. Регистр *CS* содержит начальный адрес сегмента программы, *DS* — сегмента данных, *SS* — сегмента стека, а *ES* — дополнительного сегмента. Эти регистры, так же как и все остальные регистры МП, 16-разрядные, и в них нельзя записать все разряды начального адреса. Поэтому в них загружаются 16 старших разрядов начального адреса сегмента. Оставшиеся младшие разряды адреса считаются равными нулю. МП Intel 8086 имеет адресное пространство, равное 1 Мбайту, и равными нулю считаются четыре младших разряда адреса сегмента. Всего МП Intel 8086 одновременно работает с четырьмя сегментами, которые могут быть как смежными, так и перекрывающимися. В последнем случае одни и те же ячейки памяти могут быть адресованы как в одном, так и в другом сегментах (см. рис. 4.13). Местоположение ячейки памяти внутри сегмента определяется

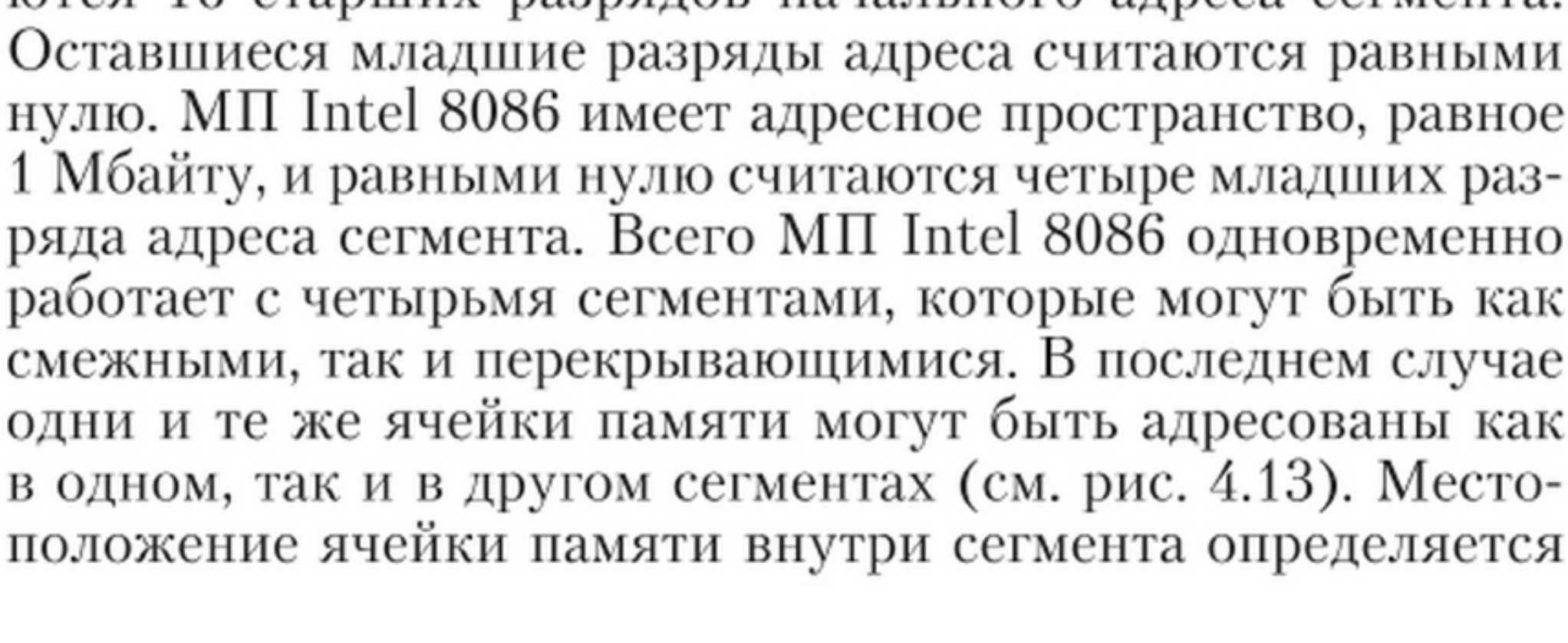


Рис. 4.13. **Сегментация памяти в МП Intel 8086**

Содержание

16-разрядным *внутрисегментным адресом*. Полный физический адрес ячейки памяти образуется сложением адреса сегмента с внутрисегментным адресом, которое выполняет сумматор шинного интерфейса. Если это адрес команды, то складываются содержимые сегментного регистра программы и программного счетчика, если это адрес операнда, то внутрисегментный адрес поступает из операционного устройства и складывается с содержимым сегментного регистра данных. Младшие 16 разрядов вычисленного адреса поступают на внешнюю шину адреса/данных. Для передачи четырех старших разрядов используются еще четыре линии, которые также работают в мультиплексном режиме.

4.5.4. Управляющие сигналы МП Intel 8086

В микропроцессоре имеется целый набор управляющих сигналов, обеспечивающих работу микроЭВМ и ее взаимодействие с ВУ. Всего таких сигналов 17; в зависимости от выполняемой функции их можно разбить на четыре группы.

1. Выходные сигналы, обеспечивающие управление устройствами при их работе с внешней мультиплексной шиной адреса/данных. Это такие сигналы, как «Запись» и «Чтение» (в микроЭВМ центральным узлом считается МП, поэтому под «Записью» понимается запись информации из МП в ОЗУ или ВУ, а под «Чтением» — чтение данных из ОЗУ или ВУ в МП); сигналы, идентифицирующие тип устройства, с которым работает МП, — память или ВУ; стробирующие сигналы, указывающие устройствам, что именно передается в данный момент по мультиплексной шине, — адрес или данные.

2. Сигналы, обеспечивающие работу МП в режиме прерывания. К ним относят сигнал запроса прерывания (входной для МП) и сигнал подтверждения прерывания (выходной).

3. Сигналы, обеспечивающие работу микроЭВМ в режиме ПДП. К этим сигналам относят сигнал запроса захвата шин и сигнал подтверждения захвата.

4. Сигналы, управляющие работой МП, — сигналы синхронизации, сброса, а также сигналы, обеспечивающие многопроцессорный режим работы. С сигнала сброса начи-

нается работа МП. Приняв этот сигнал, МП обнуляет все свои регистры, после чего он формирует нулевой адрес первой команды программы, считывает ее и начинает работу.

МП Intel 8086 может работать в режиме *минимальной конфигурации*, т.е. когда МП один и все управляющие сигналы он генерирует сам, и в режиме *максимальной конфигурации*. В этом режиме в одну систему объединяется несколько МП, которые взаимодействуют по одной системной шине. Управление системной шиной в этом случае осуществляет не МП, а специальный контроллер шины. В многопроцессорном режиме часть сигналов управления системной шиной и четыре мультиплексных линии, по которым передаются четыре старших разряда адреса, используются для индикации состояния МП. Объединение нескольких МП для увеличения производительности микропроцессорной системы используется очень часто.

4.5.5. Структура команды МП Intel 8086

Команды, выполняемые МП, записаны в памяти — постоянной или оперативной. Поскольку информационной единицей для адресной памяти является один байт, то длина команды обычно кратна одному байту. В МП Intel 8086 длина команды может изменяться от 1 до 6 байтов. Команда содержит сведения о выполняемой операции в виде ее кода, а также сведения об операндах (рис. 4.14).

Код операции	Поле операнда 1	Поле операнда 2

Рис. 4.14. **Обобщенный формат команды**

Система команд МП насчитывает более ста команд, поэтому поле кода операции в большинстве команд составляет один байт и помещается в первом байте команды. В нем помимо сведений об операции содержатся некоторые сведения об операндах и, в частности, об их разрядности. Чтобы сократить длину команды, число операндов обычно ограничивают одним-двумя. Если операндов два, то один из них обязательно должен храниться в одном из внутренних регистров МП, поэтому для указания места его расположения достаточно нескольких разрядов, размещаемых во втором байте команды. Другой операнд может присутствовать в самой команде в виде двоичного числа, но гораздо чаще в команде указаны данные, по которым МП определяет

непосредственное местонахождение операнда. Операнд может храниться во внутреннем регистре процессора или в ячейке (нескольких ячейках) ОЗУ. В последнем случае процессор должен вычислить адрес этой ячейки. Результат операции обычно размещается на месте первого операнда. При этом сам первый операнд теряется, но обычно это не играет роли. Если его необходимо сохранить для дальнейших вычислений, можно предварительно, до выполнения команды, запомнить его в какой-либо другой ячейке ОЗУ.

4.6. Режимы адресации и система команд микропроцессора Intel 8086

4.6.1. Режимы адресации

Способ определения местонахождения операнда называется режимом адресации. Различают семь основных режимов адресации данных, которые можно разделить на две группы. К первой относят режимы, в которых место, в котором находится операнд, указывается непосредственно в команде. Это — непосредственный, регистровый и прямой режимы адресации. В *непосредственном* режиме адресации операнд располагается в самой команде в виде двоичного числа (рис. 4.15, *а*). В *прямом* режиме в команде после кода операции располагается внутрисегментный адрес ячейки ОЗУ, в которой хранится операнд (рис. 4.15, *б*). Наконец, в *регистровом* режиме в команде указан один из регистров данных или адреса, в котором хранится операнд (рис. 4.15, *в*).

Ко второй группе режимов адресации относят режимы, в которых содержатся данные для вычисления внутрисегментного адреса ячейки ОЗУ, хранящей операнд. Это — четыре *косвенных* режима: регистровый косвенный, регистровый относительный, базовый индексный и относительный базовый индексный. При формировании адреса в этих режимах используется содержимое универсальных адресных регистров, дополнительно к которым выделяется регистр данных *BX*.

Самым простым является *регистровый косвенный* режим (рис. 4.15, *г*), в котором адрес операнда хранится в одном из индексных регистров, или в регистре *BX*, а в команде указывается код этого регистра. Более сложными явля-

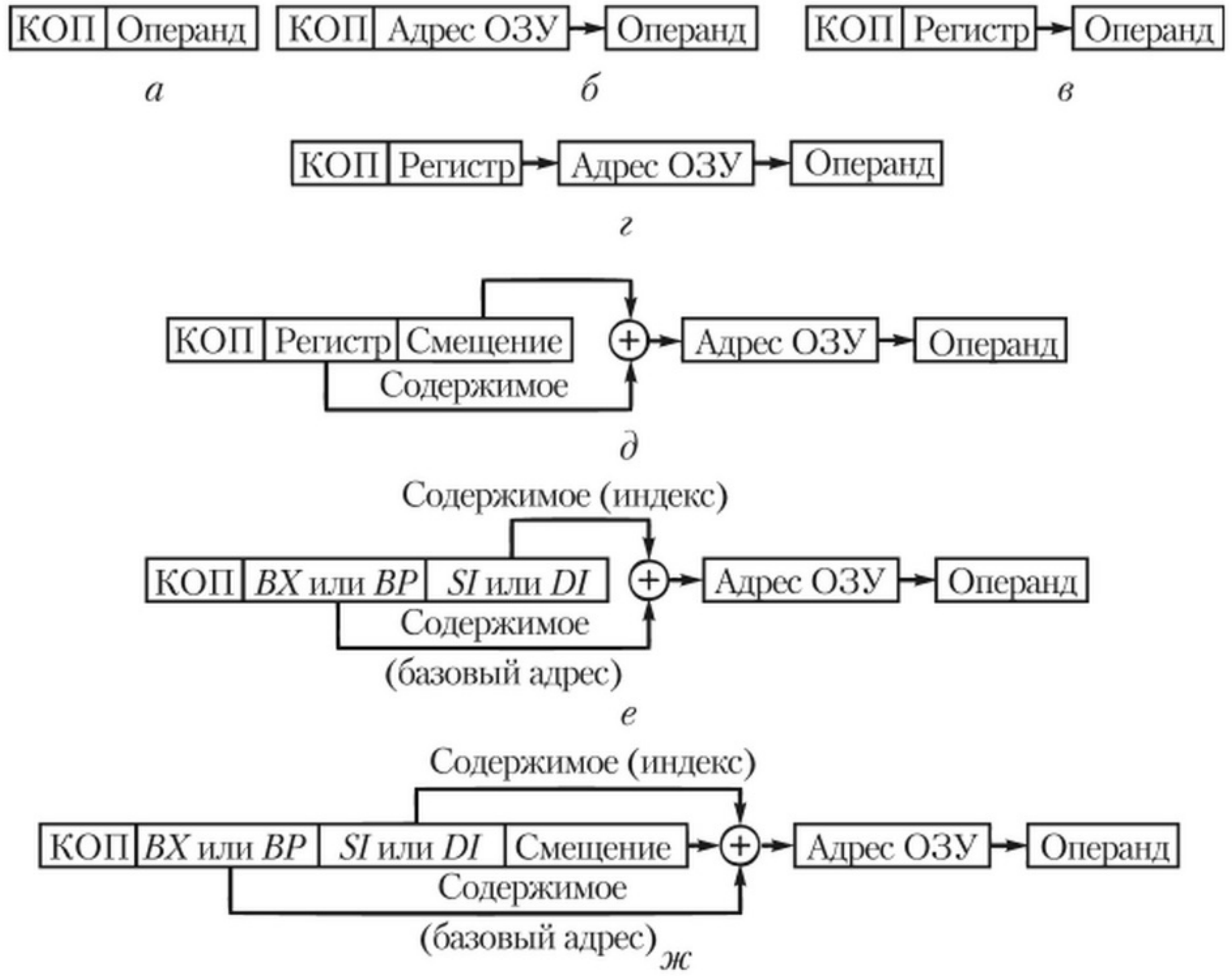


Рис. 4.15. **Непосредственный (*а*), прямой (*б*), регистровый (*в*), регистровый косвенный (*г*), регистровый относительный (*д*), базовый индексный (*е*) и относительный базовый индексный (*ж*) режимы адресации**

ются *регистровый относительный* (рис. 4.15, *д*) и *базовый индексный* (рис. 4.15, *е*) режимы адресации. В первом адрес образуется путем сложения содержимого одного из указанных регистров и *смещения* — двоичного числа, хранящегося в команде в поле операнда. Во втором режиме адрес получается путем сложения содержимого одного из базовых регистров (*BP*, *BX*) с содержимым одного из индексных регистров (*SI*, *DI*).

Самым сложным косвенным режимом адресации является *относительный базовый индексный* режим (рис. 4.15, *ж*). Здесь адрес операнда образуется путем сложения трех чисел: содержимого одного из базовых регистров, одного из индексных регистров и смещения.

Прямой и все косвенные режимы адресации данных определяют только внутрисегментный адрес. Для вычисления физического адреса ячейки ОЗУ кроме внутрисегментного необходим начальный адрес сегмента. Код сегментного регистра, хранящего этот адрес, содержится во втором байте команды.

Косвенные режимы адресации обеспечивают гибкость создаваемых программ. Как известно, очень часто программы содержат циклические участки, когда одна и та же группа команд исполняется многократно, но с разными данными. Если в программе использовать команды с прямым режимом адресации, то процессор будет постоянно обращаться к одним и тем же данным. При использовании косвенных режимов адресации в процессе исполнения команд можно модифицировать содержимое адресного регистра, например, увеличивая в каждом цикле на единицу его содержимое. В этом случае в очередном цикле вычисленный адрес ячейки ОЗУ будет на единицу больше, чем в предыдущем. Таким образом, одни и те же команды программы будут обращаться уже к разным ячейкам ОЗУ. Это дает возможность небольшим циклическим участком программы обработать большой массив данных.

Работу с массивами можно организовать следующим образом. Поместим в базовый регистр (*BX* или *BP*) начальный адрес массива, а в индексный регистр (*DI* или *SI*) — номер элемента массива. Номер элемента можно в каждом цикле увеличивать на единицу (например, *SI*: = *SI* + 1). В результате в каждом новом цикле адрес ячейки ОЗУ, вычисленный в режиме базовой индексной адресации, будет соответствовать новому элементу массива.

С помощью режима относительной базовой индексной адресации можно оперировать не только с одномерными, но и с двухмерными массивами. Для этого в качестве смещения необходимо указать начальный адрес массива. В базовом регистре формируют адрес первого элемента строки относительно начального адреса массива, а в индексном регистре — адрес элемента массива в своей строке относительно начального адреса строки.

Кроме режимов адресации, используемых для определения местонахождения операндов, в МП Intel 8086 имеется несколько режимов для определения адреса следующей выполняемой команды в случаях переходов. Переходы возможны как внутри сегмента исполняемой программы, так и в другие сегменты. Для переходов внутри сегмента исполняемой программы используются *внутрисегментный прямой* и *внутрисегментный косвенный* режимы адресации. В прямом режиме адрес следующей команды формируется путем сложения содержимого программного счетчика со смещением (положительным или отрицательным), содержащимся в самой команде. В косвенном режиме адрес следующей команды загружается на место текущего непосредственно в программный счетчик. Адрес может быть взят либо из регистра, либо из ячейки ОЗУ. Чтобы указать эту ячейку ОЗУ, используются все возможные режимы адресации данных.

Для перехода из одного сегмента памяти в другой недостаточно изменить только содержимое программного счетчика. Для этого нужно занести новый начальный адрес сегмента в регистр *CS*. Для этого используют прямой и косвенный межсегментные режимы адресации. В первом из них в команде содержится два адреса — новые адрес сегмента и внутрисегментный. Во втором — оба адреса хранятся в соседних ячейках ОЗУ.

4.6.2. Система команд

Понятие об Ассемблере. Команды, выполняемые микропроцессором, представляются комбинациями единиц и нулей. Такое представление команд называют *машинными командами*. Чтобы представить машинные команды более понятными для человека, используют их символьные (мнемонические) обозначения. В этом случае каждой машинной команде соответствует командный оператор на специальном языке — *Ассемблере* и машинная программа

представлена в виде ассемблерной программы. Программа на языке Ассемблер является простейшей, так как в точности описывает последовательность действий микропроцессора. Для перевода ассемблерной программы на машинный язык единиц и нулей можно, конечно, воспользоваться таблицами перевода, но практически для этого имеются специальные ассемблирующие программы. Процесс преобразования программы с языка Ассемблер в машинные коды называется *трансляцией*. Программа, написанная на языке Ассемблер, кроме операторов содержит также *директивы*, т.е. специальные указания ассемблирующей программе. Они не транслируются в машинные команды, но информация, заложенная в них, используется для преобразования командных операторов в машинные команды.

Общий формат ассемблерной команды имеет следующий вид:

Метка: Мнемоника Операнд, Операнд; Комментарий.

Метка — это идентификатор, присваиваемый команде. С помощью меток в программе можно указывать команды, которым следует передать управление в случае условных или безусловных переходов. *Мнемоника* — краткое обозначение команды, обычно состоящее из трех-четырех латинских букв, представляющих сокращение от английского названия команды: ***ADD*** — *addition* (сложение); ***MOV*** — *move* (переслать) и т.д. *Операнд*, *Операнд* — поля первого и второго операндов. Первый операнд является приемником результата операции. Запятая служит признаком разделения полей операндов. Один из операндов — содержимое регистра, второй задается всеми возможными режимами адресации. *Комментарий* — необязательная составляющая оператора. Он игнорируется при трансляции и служит только для пояснений программы, ее отдельных фрагментов или операторов.

Для записи операндов используются следующие правила.

1. Если в регистре содержится сам операнд, его обозначение дается без скобок, если в регистре содержится составляющая адреса ячейки ОЗУ при косвенной адресации, то дается обозначение регистра в квадратных скобках. Например, запись ***ADD AX, [BX]*** означает, что в операции сложения первый операнд находится в регистре ***AX***, а второй — в ячейке ОЗУ, адрес которой содержится в реги-

стре ***BX***. По записи оператора ассемблирующая программа автоматически определяет режим адресации.

2. Операнд может быть задан идентификатором — каким-либо сочетанием букв, не совпадающим с наименованием регистров, например ***ADD AX, BETA***. В этом случае идентификатору (в нашем случае метке ***BETA***) соответствует ячейка ОЗУ, которая должна быть зарезервирована за ним в начале программы специальным оператором. Ассемблирующая программа при трансляции вычислит смещение этой ячейки относительно начала сегмента, и соответствующий операнд будет задан в машинной команде в режиме прямой адресации в виде адреса.

3. Знак «+» в записи операнда используется для задачи смещения. Например, запись ***ADD AX,*** **[*SI*] +8** определяет второй операнд в режиме регистровой относительной адресации. Адрес ячейки ОЗУ, содержащей второй операнд, определится как сумма содержимого регистра ***SI*** и смещения «**8**». Знак «**+**» можно использовать также для задачи дополнительного смещения. Например, операнд, заданный как ***BETA*+2**, будет указывать на ячейку ОЗУ, адрес которой больше адреса ячейки ***BETA*** на «**2**». При этом ассемблирующая программа вычислит соответствующий адрес и поместит его в машинную команду в режиме прямой адресации.

4. Операнд может быть задан непосредственно числом. При этом необходимо указать, в какой системе счисления оно записано. Этот признак проставляется сразу после числа. Ассемблирующая программа самостоятельно переведет это число в двоичную форму и занесет в машинную команду. Двоичные числа сопровождаются признаком *B* (например, 10100110*B*); десятичные сопровождаются признаком *D* или совсем без признака; шестнадцатеричные — признаком *H*. В шестнадцатеричных числах для обозначения цифр 10, 11, 12, 13, 14 и 15 используются буквы *A*, *B*, *C*, *D*, *E* и *F*. Так, например, запись 53*AFH* обозначает шестнадцатеричное число 53*AF*, которое после трансляции запишется в машинной команде в виде двоичного числа

$$\underbrace{0101}_{5}\,\underbrace{0011}_{3}\,\underbrace{1010}_{10(A)}\,\underbrace{1111}_{15(F)}.$$

Команды передач данных. Во всех компьютерах необходимы команды, предназначенные просто для пересылок данных, адресов и непосредственных операндов в регистры

или ячейки памяти. В МП Intel 8086 имеется пять базовых команд для выполнения таких действий:

MOV — переслать;

LEA — загрузить (*load*) эффективный (внутрисегментный) адрес;

LDS и ***LES*** — загрузить адрес в сегментные регистры ***DS*** и ***ES*** соответственно;

XCHG — обменять (*exchange*).

MOV — наиболее универсальная команда, предназначенная для пересылки информации между регистрами МП, регистрами и ячейками памяти, загрузки непосредственного операнда в регистр или ячейку памяти. Для краткого символического описания сути выполняемых командами МП действий введем следующие обозначения:

(...) — содержимое регистра или ячейки памяти;

← — заменить старое содержимое новым.

Приведем несколько примеров команды ***MOV*** с использованием разных режимов адресации:

- ***MOV AX, SI*** **(*AX*) ← (*SI*)**

в регистр ***AX*** заносится содержимое регистра ***SI***;

- ***MOV ALPHA, CL*** **(*ALPHA*) ← (*CL*)**

в ячейку ***ALPHA*** заносится содержимое регистра ***CL***, при этом сама ячейка должна быть объявлена как однобайтная;

- ***MOV BETA*+5, 9835*H*** **(*BETA*+5) ← 9835*H***

в двухбайтную ячейку, отстоящую от ячейки ***BETA*** на 5 байтов, заносится шестнадцатеричное число **9835**, т.е. 1001100000110101;

- ***MOV DX, [BP] [SI]*+30** **(*DX*) ← ([*BP*] [*SI*]+30)**

в регистр ***DX*** заносится содержимое двухбайтной ячейки, внутрисегментный адрес которой вычисляется сложением содержимого регистров ***BP***, ***SI*** и смещения **30**.

Команды ***LEA***, ***LDS*** и ***LES*** обеспечивают загрузку адресов из памяти. Команда ***LEA*** загружает внутрисегментный адрес в любой регистр МП, этот адрес в ассемблере называется эффективным. Адрес ячейки памяти может быть указан всеми косвенными способами адресации, причем если в командах ***MOV***, ***LDS***, ***LES*** при этом будет выбрано содержимое ячейки, то в этой команде будет загружен именно адрес. Команды ***LDS*** и ***LES*** аналогичны, но первая загружает из памяти регистр ***DS***, а вторая — ***ES***. Обе команды кроме сегментных регистров загружают также дополнительно еще один внутренний регистр МП (несегментный),

указанный в качестве операнда в команде. Типичный вид команд:

LEA SI, COL [BX] **(*SI*) ← *COL [BX]***
LDS DI, TABLE [BX] **(*DI*) ← (*TABLE [BX]*)**
(*DS*) ← (*TABLE [BX]*+2).

Команда ***XCHG*** осуществляет обмен содержимым двух операндов, т.е. первый операнд помещается на место второго, а второй — на место первого:

XCHG BX, [BP] [SI]+68 **(*BX*) ↔ ([*BP*] [*SI*]+68).**

Арифметические команды. К арифметическим операциям, выполняемым МП Intel 8086, относятся сложение, вычитание, умножение и деление. Однако набор арифметических команд МП гораздо больше. Так как разрядность регистров МП составляет 16, диапазон чисел, представленных таким числом двоичных разрядов, ограничен и составляет всего ±32 768. Для расширения диапазона представляемых чисел двухбайтные слова могут сцепляться в четырехбайтные, шестибайтные и т.д. Поэтому наряду с командами, обеспечивающими сложение и вычитание обычных 16-разрядных чисел, в МП есть команды для выполнения арифметических операций с учетом переноса или заема, образовавшегося при обработке предыдущего двухбайтного слова цепочки.

Команды сложения и вычитания без учета и с учетом переноса (заема) имеют следующий вид:

ADD op1, op2 — сложение без учета переноса;
SUB op1, op2 — вычитание без учета заема;
ADC op1, op2 — сложение с учетом переноса;
SBB op1, op2 — вычитание с учетом заема.

Под «***op1***» и «***op2***» понимают первый и второй операнды, один из которых — содержимое регистра, а другой может быть задан всеми возможными режимами адресации данных.

Кроме команд, реализующих обычные операции сложения и вычитания двух операндов в МП, как и во многих других, имеются команды сложения (инкремент) и вычитания (декремент) операнда с единицей: ***INC op1*** — инкремент; ***DEC op1*** — декремент.

В этих командах только один операнд (другой определен операцией, это — единица), а результат операции помещается на прежнее место операнда. В качестве операнда может выступать содержимое регистра МП или ячейки ОЗУ:

INC AX	**(*AX*) ← (*AX*)+1**
***DEC* [*BX*] [*DП*]+45**	**([*BX*] [*DП*]+45) ← ([*BX*] [*DП*]+45) − 1.**

Очень часто возникает задача сравнения двух операндов. При этом необходимо определить, какой из операндов больше или меньше, но сами операнды нужно сохранить. Для этого в МП есть команда сравнения двух операндов ***CMP***. Сравнение операндов осуществляется путем вычитания второго операнда из первого. При этом формируются все признаки результата операции, но сам результат нигде не сохраняется. Эта команда схожа с командой ***SUB***, но отличается от нее тем, что результат операции не запоминается.

В двоичной арифметике, как и в любой другой, разрядность произведения больше разрядности сомножителей. Если в качестве сомножителей выступают однобайтные операнды, то произведение будет двухбайтным, если операнды двухбайтные, то произведение — четырехбайтное. В качестве одного из операндов всегда выступает содержимое регистра ***AX***, или его младшей половины ***AL***, и в команде он не обозначается. Другой операнд может быть содержимым регистра или ячейки ОЗУ. Если в команде указан однобайтный операнд, то он умножается на содержимое регистра ***AL***, а двухбайтное произведение размещается в регистре ***AX***. Если же операнд двухбайтный, то он умножается на содержимое всего регистра ***AX***, для размещения произведения требуется еще один регистр. Старшая половина произведения помещается в регистр ***DX***, а младшая — в ***AX***.

Операция деления является обратной умножению. Здесь также указывается только один операнд — делитель, а делимое всегда размещается либо в регистрах ***DX***, ***AX***, либо только в регистре ***AX***. Частное всегда помещается в регистр ***AX*** **(*AL*)**.

Другая особенность двоичной арифметики — в работе с двоичными числами со знаком и без знака. Если число со знаком, его функцию выполняет старший разряд, при этом «0» соответствует положительному числу, а «1» — отрицательному. Поэтому операции умножения и деления с учетом знака чисел и без учета знака также отличаются друг от друга, и команды МП, выполняющие операции с учетом и без учета знака операндов, разные. Приведем команды умножения и деления беззнаковых чисел с кратким символическим пояснением:

MUL GAMMA **(*DX*) (*AX*) ← (*AX*)×(*GAMMA*)**
(при условии, что ***GAMMA*** — двухбайтная ячейка);
DIV CH **(*AL*) ← (*AX*):(*CH*)**
DIV [SI] **(*AX*) ← (*DX, AX*):([*SI*])**
(при условии, что ячейка, адрес которой содержится в ***SI***, — двухбайтная).

Беззнаковые команды умножения и деления используются при обработке чисел с разрядностью, кратной двум байтам, составленных из цепочек. Мнемоника команд, обеспечивающих обработку чисел с учетом знака, отличается наличием буквы «*I*» в обозначении: ***IMUL op1*** и ***IDIV op1***.

Наконец, еще одной особенностью арифметики МП является возможность представления двоично-десятичных чисел. В таких числах четыре двоичных разряда объединяются в тетраду и соответствуют одной десятичной цифре. Двухбайтное двоичное число может соответствовать четырехразрядному десятичному числу. Такая форма представления чисел и называется двоично-десятичной. Так, например, десятичное число 9580_{10} запишется в двоично-десятичном виде как 1001010110000000. Операции над двоично-десятичными числами выполняются в два этапа. Сначала осуществляется операция над операндами, как обычными двоичными числами, а затем осуществляется коррекция результата с помощью специальных команд коррекции.

Логические команды и команды сдвига. В МП могут выполняться все базовые логические операции, а также некоторые другие. Их мнемоника практически совпадает с названием операций:

AND op1, op2 — (логическое И);
OR op1, op2 — (логическое ИЛИ);
NOT op1 — (логическое НЕ);
XOR op1, op2 — (исключающее ИЛИ или сумма по модулю два).

Каждый разряд двоичного числа в логических командах рассматривается как логическая переменная — «1» или «0» и выполнение логических операций осуществляется поразрядно. Команда ***AND*** используется для того, чтобы сбросить в «0» отдельные разряды операнда, не затронув остальные. Для этого операция логического И выполняется между операндом и «маской» — двоичным числом, все разряды которого равны единице, а те, которые у операнда должны быть установлены в «0», — равны нулю. Напри-

мер, для установки 3-го разряда у некоторого операнда A в нуль достаточно выполнить операцию логического «И» с маской, в которой все разряды, кроме третьего, равны «1», т.е. 11110111:

$$a_7a_6a_5a_4a_3a_2a_1a_0 \wedge 11110111 = a_7a_6a_5a_4 0\, a_2a_1a_0.$$

Для установки в единицу отдельных разрядов операнда, не затрагивая других, используется команда ***OR*** (логическое ИЛИ). Для этого в качестве маски выбирается число, в котором все разряды равны нулю, кроме тех, которые в операнде должны быть установлены в единицу:

$$a_7a_6a_5a_4a_3a_2a_1a_0 \vee 00100001 = a_7a_6 1 a_4a_3a_2a_1 1.$$

В данном примере с помощью операции логического ИЛИ пятый и нулевой разряды операнда A устанавливаются в «1», не затрагивая остальных.

Команда ***XOR*** используется для получения инверсии всех или отдельных разрядов операнда A или обнуления всех его разрядов. В первом случае необходима маска, в которой разряды, подлежащие инвертированию, равны «1», во втором достаточно выполнить команду, используя в качестве операндов одно и то же слово:

$$a_7a_6a_5a_4a_3a_2a_1a_0 \oplus 11111111 = \bar{a}_7\bar{a}_6\bar{a}_5\bar{a}_4\bar{a}_3\bar{a}_2\bar{a}_1\bar{a}_0;$$

$$a_7a_6a_5a_4a_3a_2a_1a_0 \oplus a_7a_6a_5a_4a_3a_2a_1a_0 = 00000000.$$

Логической командой, позволяющей осуществлять проверку отдельных разрядов операнда без изменения его, является команда ***TEST***. Эта команда выполняет операцию логического И операнда с маской. При этом устанавливаются все признаки результата, а сам результат не сохраняется и операнд таким образом не изменяется.

Для выполнения операций сдвига битов операнда на определенное число разрядов влево или вправо в МП Intel 8086 имеется восемь команд. Сдвиги подразделяются на простые и циклические, арифметические и логические, сдвиги с учетом бита переноса ***CF*** или без учета (рис. 4.16).

В командах логических сдвигов влево или вправо (рис. 4.16, *а*) с противоположной стороны операнда «вдвигаются» нули, а выдвигаемые биты теряются, кроме последнего, который сохраняется в бите переноса ***CF***. Арифме-

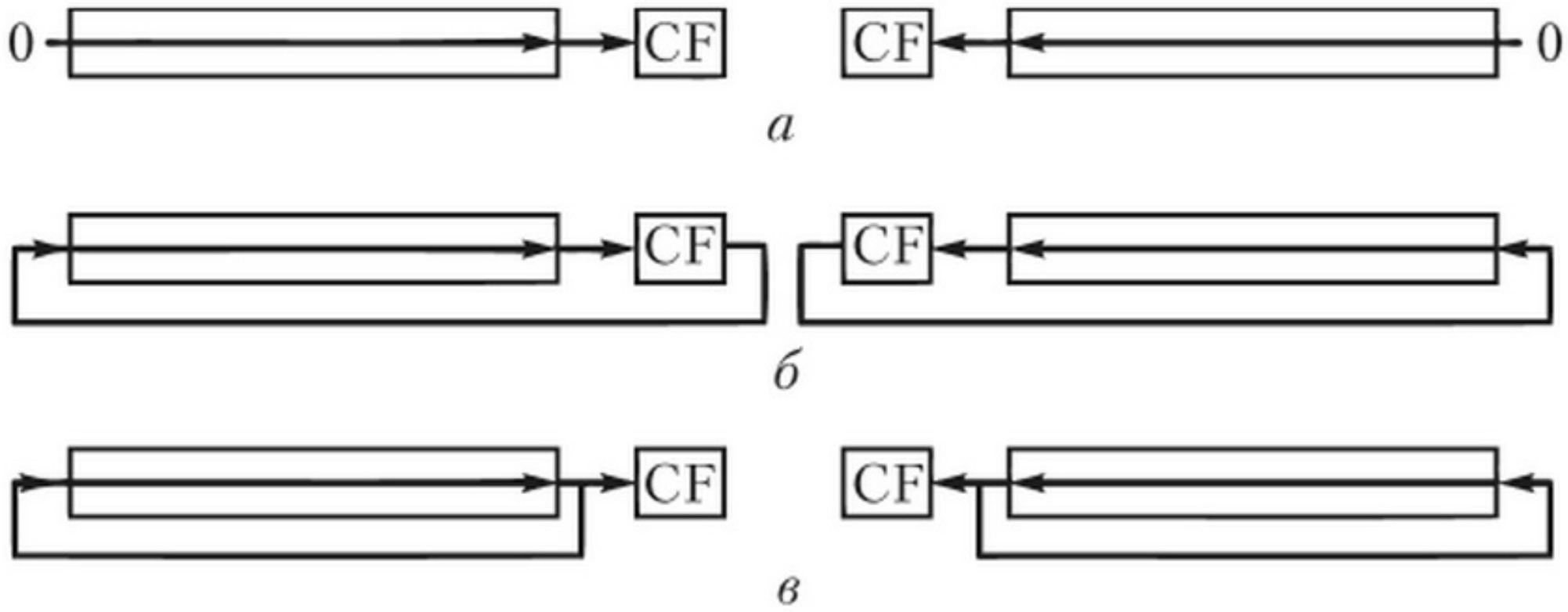


Рис. 4.16. **Логический сдвиг (*а*); циклический сдвиг через перенос (*б*); циклический сдвиг (*в*)**

тический сдвиг влево аналогичен логическому, а сдвиг вправо (на рисунке не показан) отличается от логического тем, что вместо нулей на место выдвигаемых битов дублируется самый старший знаковый бит. Команды циклического сдвига отличаются тем, что операнд считается «кольцом», в котором выдвигаемые с одной стороны биты вдвигаются с другой (см. рис. 4.16, *б*, *в*). Циклический сдвиг может осуществляться с учетом бита переноса или без учета. Если сдвиг осуществляется с учетом бита переноса (см. рис. 4.16, *б*), то этот бит включается в кольцо.

Команда сдвига имеет следующий обобщенный вид:

Мнемоника Операнд, Число сдвигов.

Мнемоника команд сдвига состоит из трех символов. Первый обозначает тип сдвига: простой ***S*** (shift), или циклический, называемый ротацией ***R*** (*rotation*). Второй символ конкретизирует тип сдвига: арифметический ***H***, логический ***L***, простой циклический ***O***, циклический с учетом переноса ***C***. Третий символ обозначает направление сдвига: влево ***L*** и вправо ***R***. Например, команда ***RCL*** соответствует циклическому сдвигу с учетом переноса влево. В качестве операнда может выступать содержимое регистра или ячейки ОЗУ. Число сдвигов может быть указано числовой константой либо косвенно, путем указания регистра ***CL***. В последнем случае в этот регистр предварительно должна быть занесена константа, соответствующая числу сдвигов. Приведем примеры команд сдвигов:

***SLR* [*SI*]+50, 1**

логический сдвиг вправо содержимого ячейки ОЗУ с адресом [***SI***] +**50** на один разряд;

ROL SIGMA, CL

циклический сдвиг влево содержимого ячейки ОЗУ ***SIGMA***. Количество сдвигаемых разрядов содержится в регистре ***CL***.

Команды условных и безусловных переходов. Нарушение естественного порядка следования команд, когда адрес следующей выполняемой команды автоматически вычисляется программным счетчиком, осуществляется путем загрузки в программный счетчик и, при необходимости, в сегментный регистр ***CS*** новых адресов. Эти действия выполняются при помощи команд условных и безусловных переходов. Мнемоника этих команд начинается с символа ***J*** (*jump*).

Команды безусловного перехода имеют одну и ту же мнемонику ***JMP*** и дополнительную уточняющую информацию, определяющую дальность перехода. В командах безусловного перехода адрес новой команды может формироваться двумя способами. В первом способе в самой команде указывается смещение относительно текущей команды, которое может быть как положительным, так и отрицательным. Адрес, загружаемый в программный счетчик, образуется путем сложения с его текущим содержимым этого смещения, или его вычитания. Смещение может быть задано как однобайтным числом, так и двухбайтным. В первом случае переход называется коротким (***SHORT***), отстоящим от текущего адреса не более, чем на 127, во втором случае — близким (***NEAR***), т.е. находящимся в одном сегменте с текущим адресом, напомним, что объем одного сегмента — $2^{16} = 64$ Кбайт.

Во втором способе адрес новой исполняемой команды непосредственно загружается в программный счетчик. При этом он может быть взят из команды в виде двухбайтного числа, или из ячейки памяти, адрес которой указан в команде всеми возможными режимами адресации. Второй способ обеспечивает переход не только внутри текущего сегмента программы, но и в любой другой. Для этого новый адрес загружается не только в программный счетчик, но и в сегментный регистр ***CS***. Он может быть задан либо вторым двухбайтным словом, содержащимся в команде, либо содержимым двух ячеек ОЗУ, находящихся следом за ячейками с внутрисегментным адресом для программного счетчика. Такой переход называют далеким (***FAR***).

Содержание

При транслировании команд с языка Ассемблер в машинные коды дальность перехода и тип определяются либо дополнительной информацией — ***SHORT***, ***NEAR*** или ***FAR***, либо размером адреса (одно-, двух-, четырехбайтный). Очень часто в качестве адреса в команде дается ссылка на метку, которой помечена команда, к которой осуществляется переход:

JMP SHORT NEXT.

По этой команде адрес следующей исполняемой команды определится как сумма текущего с однобайтным смещением. Расчет смещения выполняет ассемблирующая программа по отстоянию текущей команды от команды, помеченной меткой ***NEXT***:

JMP FAR [DI]+100.

Эта команда определяет межсегментный переход. Адреса, загружаемые в программный счетчик и регистр ***CS***, содержатся в четырех соседних ячейках ОЗУ. Адрес первой из них определяется как содержимое регистра ***DI*** плюс смещение 100.

Команды условных переходов имеют только один способ определения адреса перехода — как смещение, не более чем ±127 относительно текущего адреса. В этих командах осуществляется проверка различных условий выполнения перехода. Порядок действия МП при выполнении команды условного перехода следующий: если условие, заданное в команде, выполняется, то следующий адрес вычисляется путем сложения (вычитания) текущего содержимого программного счетчика со смещением; если же условие не выполняется, то автоматически сформированный адрес и есть адрес следующей команды. Таким образом, сохраняется естественный порядок следования команд и следующая команда стоит непосредственно за командой условного перехода.

Мнемоника команд условного перехода начинается с символа ***J***. Затем идут символы, определяющие проверяемое условие. Проверка условий осуществляется путем анализа различных флажков — битов регистра слова состояния процессора, хранящих признаки результата предыдущей операции. Таким образом, непосредственно перед командой условного перехода обязательно должна стоять команда, с помощью которой формируются признаки результата. У одной части команд осуществляется анализ

только одного признака результата — одного бита регистра флажков, у другой двух или трех битов.

Приведем мнемоники некоторых команд с их наименованиями и необходимыми пояснениями.

JZ — «Перейти, если ноль» — переход осуществляется, если результат предыдущей операции равен нулю.

JNZ — «Перейти, если не ноль» — команда противоположна команде ***JZ***.

JS — «Перейти, если знаковый бит равен 1» — переход выполняется, если результат вычитания двух операндов отрицателен.

JNS — «Перейти, если знаковый бит равен 0» — переход выполняется, если результат вычитания положителен.

В следующих четырех командах осуществляется проверка условий, соответствующих математическим операторам отношений $<$, $>$, $\geqslant$, $\leqslant$. Эти команды предназначены для использования после команды сравнения ***CMP***, в которой второй операнд вычитается из первого, а результат операции не сохраняется.

JL — «Перейти, если меньше».

JNL — «Перейти, если больше».

JLE — «Перейти, если меньше или равно».

JNLE — «Перейти, если больше или равно».

Команды условного перехода обычно имеют следующий обобщенный вид:

Мнемоника Метка, например ***JLE NEXT***.

В данном примере ***NEXT*** — это метка команды, которую следует выполнить в случае выполнения условия. Она должна находиться не далее чем на ± 127 относительно команды условного перехода. При трансляции ассемблирующая программа вычислит соответствующее смещение и поместит его в машинную команду. Если команда, которой следует передать управление, находится дальше или в другом сегменте, то используют конструкцию из двух команд: условного и безусловного перехода. Сначала проверяется условие с помощью команды условного перехода, которая дает ссылку на команду безусловного перехода, находящуюся вблизи ее. Затем по команде безусловного перехода осуществляется переход к нужному участку программы в любую точку памяти.

Команды ввода-вывода. Все внешние устройства рассматриваются микропроцессором как внешние регистры или порты ввода-вывода. Для передачи данных между МП и ВУ имеются две команды: ***IN*** — ввод и ***OUT*** — вывод. Весь

процесс обмена ведется через регистр ***AX*** или его отдельные половины. Он является как бы «почтовым ящиком». Через него осуществляются пересылки данных из МП в ВУ и в него же данные принимаются. Обобщенный вид команд можно представить в следующем виде:

Мнемоника Приемник, Источник.

В команде ***IN*** в качестве приемника может выступать регистр ***AX*** (***AH***, ***AL***). В качестве источника должен выступать адрес регистра ВУ, который может быть задан либо однобайтной константой (от 0 до 255), либо косвенно, путем указания регистра ***DX***. В этом случае в этот регистр должен быть занесен двухбайтный адрес регистра ВУ. В команде ***OUT*** регистр ***AX*** теперь выступает уже в качестве источника, а адрес регистра ВУ должен быть помещен на место приемника. Он может быть задан так же, как и в команде ***IN***.

В команде ***IN AX, 40*** приемником является ***AX***, а источником — регистр с адресом **40**.

В команде ***OUT DX, AL*** приемником является регистр с адресом, содержащимся в ***DX***, а источником — регистр ***AL***.

Кроме рассмотренных команд в МП Intel 8086 имеются команды для работы со стеком — записи в стек и извлечения из стека; команды для организации циклических программ; команды для установки и сброса отдельных признаков результата — флажков; команды, обеспечивающие управление микропроцессором.

4.7. Тенденции развития однокристальных микропроцессоров и систем на их основе

Со времени появления первых микропроцессоров прошло более 40 лет. За это время произошел гигантский скачок в технологии и производительности микропроцессоров.

Уже в самом начале истории развития однокристальных микропроцессоров наметились основные тенденции их развития и повышения производительности микропроцессорных систем.

1. Повышение степени интеграции. Степень интеграции непосредственно определяет, сколько транзисторов может поместиться на кристалле. Она характеризуется технологической нормой — минимально возможными размерами напыляемого элемента, т.е. области полупроводника с заданным типом проводимости (не путать с транзисто-

ром, который состоит из нескольких таких элементов). Самый первый МП Intel 4004, появившийся в 1971 г., был выполнен по 10 мкм технологии и содержал только 2300 транзисторов. МП Pentium выполнялся уже по 0,8 мкм технологии и содержал уже 3,1 млн транзисторов. В 2012 г. появилсь первые процессоры, выполненные по технологии 22 нм, которые содержали более миллиарда транзисторов. Специалисты считают, что если развитие техники будет продолжаться такими темпами, то к 2022 г. процессоры будут содержать в одном кристалле десятки и сотни сложнейших устройств с числом транзисторов до ста миллиардов.

2. Повышение тактовой частоты микропроцессора. Это самый простой и наиболее понятный способ повышения производительности микропроцессора. Если в первых МП тактовая частота составляла сотни килогерц, то в современных МП это уже сотни мегагерц — несколько гигагерц.

3. Увеличение разрядности микропроцессоров. У первых МП разрядность обрабатываемых слов составляла 8 бит. Затем вместе с развитием интегральной технологии повысилась степень интеграции БИС, что позволило создавать 16-разрядные МП. Начиная с появления МП Intel 80386, разрядность обрабатываемых слов составила 32 бита. Для большинства сфер применения МП, в том числе в персональных компьютерах, этой разрядности и на сегодняшний день вполне достаточно. Разработанные в последнее время 64-разрядные однокристальные микропроцессоры обеспечивают еще более высокую производительность.

Кроме разрядности обрабатываемых слов, для микропроцессорной системы очень важна разрядность адресов, с которыми может работать МП. Это определяет объем адресуемой памяти системы, а значит, и возможность работы с бóльшим объемом программ и обрабатываемых данных, более совершенным программным обеспечением, более производительными и удобными для пользователя операционными системами.

4. Распараллеливание процесса обработки данных. В самых первых микропроцессорах, использованных в качестве процессоров персональных компьютеров — Intel 8080, — процесс выполнения команды растягивался во времени на большое число тактов. Каждая команда сначала считывалась из программной памяти, причем сам микропроцессор в это время простаивал, и только после этого

команда исполнялась. Уже в следующем МП Intel 8086 процессы считывания команды и ее исполнения возлагались на разные устройства МП — ШИ и ОУ. Это позволило распараллелить процессы выполнения команды и ее выборки. В дальнейшем архитектура микропроцессоров еще более совершенствовалась, и принцип повышения производительности за счет распараллеливания процесса выполнения команд отобразился в конвейерной структуре МП. Процесс исполнения команды при этом разбивается на несколько ступеней, а каждой ступени соответствует свой модуль в структуре МП. С очередным тактовым импульсом команды продвигаются на следующую ступень. Выполненная команда покидает конвейер, а новая поступает в него. Таким образом, процессор одновременно осуществляет обработку нескольких команд, хотя каждая команда, взятая в отдельности, проходит несколько ступеней обработки.

Процессоры, имеющие несколько ступеней, получили название *суперскалярные*, а имеющие несколько конвейеров — *суперконвейерные*.

5. Использование сопроцессоров для выполнения математических и других операций. МП может выполнять множество различных команд, которые являются достаточно элементарными действиями. Более сложные действия, например операции умножения и деления, выполняются программно. Но, как известно, любое сложное вычисление может быть выполнено чисто аппаратным путем за гораздо меньшее время. Поэтому можно добиться существенного повышения производительности микропроцессорной системы, если параллельно с основным процессором будет работать математический сопроцессор. Такие сопроцессоры имеют специальную систему команд, ориентированную на выполнение математических операций — умножение, деление, вычисление трансцендентных функций, операции с вещественными числами (числами с плавающей точкой) и т.д. Сопроцессоры не осуществляют выборку команд — эту функцию выполняет основной процессор. Но при появлении специальных команд, предназначенных для сопроцессора, он активизируется и выполняет требуемую операцию. Команды и данные он получает по системной ШД, а результат операции может быть передан в основной процессор либо в память. В процессе работы основной процессор и сопроцессор обмениваются управля-

ющими сигналами, в соответствии с которыми сопроцессор принимает требуемые команды, а также оповещает основной процессор о своем состоянии.

Первоначально сопроцессоры выполнялись как отдельные БИС, но в дальнейшем основной процессор и сопроцессор стали совмещать в одном кристалле БИС.

Это направление развития процессорной техники привело к появлению многоядерных процессоров, у которых в одном корпусе и на одном кристалле размещается несколько процессорных ядер, каждое из которых является самостоятельным полноценным процессором, выполняющим полный набор процессорных задач.

6. Использование быстродействующей кэш-памяти. Основной причиной снижения производительности микропроцессорной системы являются затраты времени, связанные с обращением к памяти. Динамические ЗУ, на основе которых осуществляется построение оперативной памяти в большинстве микропроцессорных систем, имеют существенно меньшее быстродействие, чем процессоры. Поэтому для сокращения временны́х затрат, связанных с обращениями к оперативной памяти, используют более быстродействующую кэш-память. Кэш-память реализуется на быстродействующих триггерных элементах (статические ЗУ). Кэш-память содержит копии данных, хранящихся в ОЗУ, которые были считаны процессором ранее и параллельно занесены в кэш. Поэтому если к данным уже было обращение, то их копия, отмеченная соответствующим адресом, будет храниться в кэш-памяти.

При чтении данных МП выдает адрес, который поступает в ОЗУ и кэш-память (рис. 4.17). Если эти данные имеются в кэше, формируется сигнал *Hit* — попадание, и данные из кэша выдаются на шину данных, выдача данных из ОЗУ при этом блокируется сигналом *Hit*. Если же копии данных в кэше нет, то осуществляется обычный цикл чтения из ОЗУ, данные поступают на шину данных и параллельно заносятся в кэш-память.

Эффективность использования кэш-памяти обусловлена тем, что программы, исполняемые процессором, как правило, являются циклическими, поэтому происходит многократное обращение к одним и тем же данным или командам программы. Естественно, что степень повышения производительности микропроцессорной системы при этом зависит от характера программы, ее цикличности.

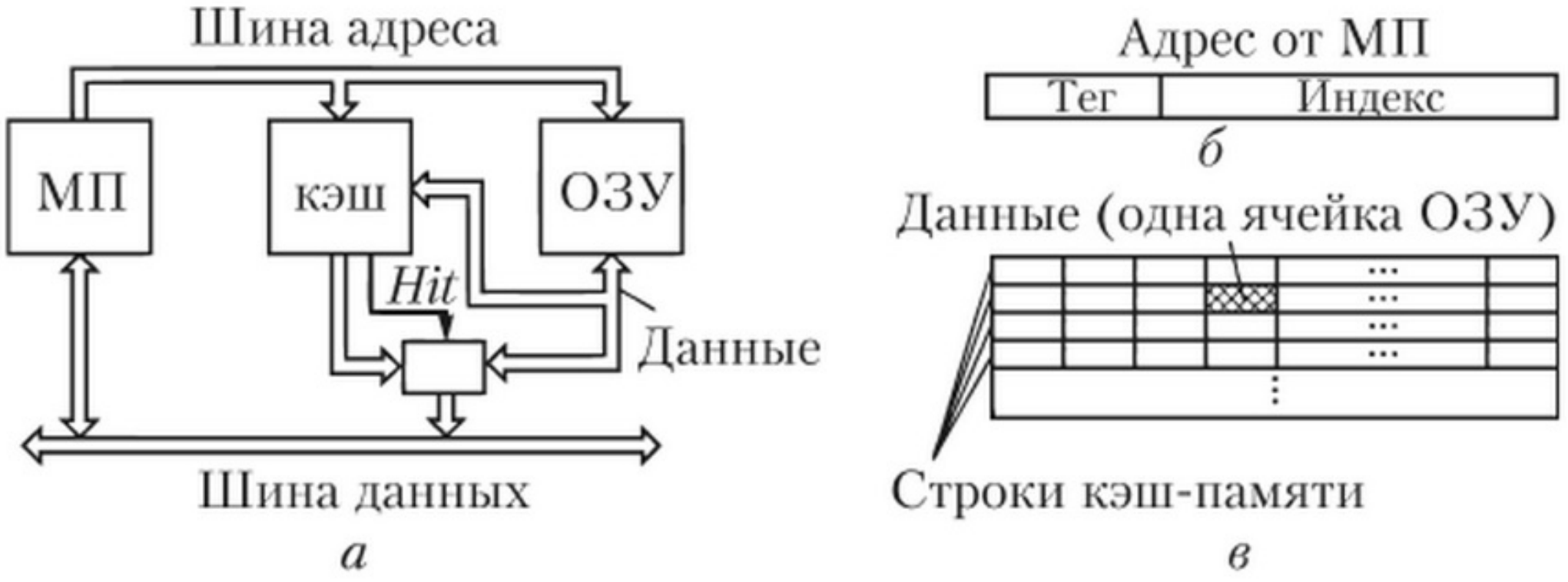


Рис. 4.17. **Структура микропроцессорной системы с кэш-памятью (*а*), структура адреса (*б*) и организация кэш-памяти (*в*)**

Кэш-память имеет небольшой объем, и любое слово, заносимое в кэш, должно сопровождаться дополнительными данными, определяющими, копией какой ячейки памяти оно является. Структура кэш-памяти показана на рис. 4.17, *в*. Запоминающие ячейки кэша разбиты на строки, причем одна строка соответствует нескольким страницам ОЗУ. Адрес ячейки памяти, поступающий от процессора, разбивается на два поля. Старшие разряды адреса образуют *тег* — цифровую метку, которая идентифицирует одну строку кэша. Эта строка имеет поле адреса, куда и заносится тег. Младшие разряды адреса соответствуют *индексу*, который определяет местоположение конкретной ячейки строки кэша (на рисунке эта ячейка заштрихована). Информация из ОЗУ в кэш заносится не отдельными байтами, а целыми страницами, т.е. построчно. Здесь необходимо отметить, что блочные пересылки данных в современных системах выполняются достаточно быстро под управлением контроллера кэш-памяти.

При поступлении в кэш адреса требуемой ячейки ОЗУ осуществляется сравнение старших разрядов (тега) с теговыми адресами всех строк. Если имеется совпадение, то вырабатывается сигнал *Hit* и считывается целая строка, из которой по значениям младших разрядов (индексу) осуществляется выборка нужной ячейки. В противном случае из ОЗУ в строку кэша заносится несколько страниц, старшие разряды адресов которых совпадают с тегом. Эта строка записывается на место строки, к которой «давно» не было обращений.

Рассмотренная структура характерна для внешней кэш-памяти, подключаемой к внешним шинам адреса и данных. В современных микропроцессорах имеется также внутренняя кэш-память, расположенная внутри кристалла процессора. Внутренняя кэш-память разделена на кэш данных и кэш команд. В процессе работы такого процессора сначала осуществляется обращение к внутренней кэш-памяти, если требуемой информации нет (сигнал *Hit* не установился), выполняется обращение к внешней кэш-памяти, и только если в этом случае требуемая информация в кэше отсутствует, выполняется обращение к ОЗУ. Производительность современных микропроцессоров во многом определяется объемами кэш-памяти.

4.7.1. Основные этапы развития однокристальных микропроцессоров

В истории развития микропроцессорной техники выделяют одиннадцать поколений микропроцессоров.

Процессоры первого и второго поколения. К этим процессорам относятся МП фирмы Intel 8080, 8086/8088 и 80286. Первым МП, на основе которого стали создавать персональные компьютеры, был МП 8080. Он был 8-разрядным, имел тактовую частоту 2 МГц и содержал 6 тыс. транзисторов. Производительность такого компьютера была невысокой, однако ее оказалось достаточно для успешного применения в качестве контроллера различных приборов, устройств и систем автоматики. В настоящее время этот МП не используется, но вместо него широко применяется 8-разрядный МП 8085, отличающийся значительно меньшей потребляемой мощностью.

Затем фирмой был разработан МП 80186, не получивший распространения, а в 1982 г. — МП 80286, ставший основой компьютера PC AT, появившегося на рынке в 1984 г. Если МП 8086 содержал 29 тыс. транзисторов, то МП 80286 уже 134 тыс. Этот МП имел ту же разрядность, но более высокие тактовую частоту и объем адресуемой памяти (до 16 Мбайт). Кроме того, МП позволял работать с так называемой виртуальной памятью и поддерживал мультизадачный режим. Если физическая память оказывалась полностью загруженной, то данные, не поместившиеся в память, располагались на винчестере, т.е. процессор работал уже не с реальными, а с виртуальными адресами,

которые формировались с помощью специальных таблиц, хранящихся во внутренней регистровой памяти МП.

В мультизадачном режиме процессор выполнял различные программы в отведенные отрезки времени, при этом пользователю казалось, что они выполняются одновременно. В целом производительность МП 80286 более чем в 6 раз превышала производительность МП 8086.

Процессоры третьего поколения. В 1985 г. был представлен 32-разрядный МП Intel 80386, который с учетом более совершенных модификаций базового МП выпускался до 1992 г. Этот МП содержал 275 тыс. транзисторов. Процессор работал с более высокой тактовой частотой (последние модификации — до 33 МГц). Удвоение разрядности адресов обеспечивало адресацию физической памяти до 4 Гбайт, а виртуальной — до 64 Гбайт. Повышение производительности МП потребовало и более быстродействующей памяти, и у МП 80386 впервые была обеспечена работа с внешней кэш-памятью, выполненной на основе быстродействующих статических ЗУ. Для этого на материнской плате компьютера кроме обычных динамических ЗУ размещались микросхемы кэш-памяти и кэш-контроллер.

Выпуском МП 80386 занималась не только фирма *Intel*, но и многие ее конкуренты — фирмы *Chips & Technologies*, *Texas Instruments*, *Cyrix*. А фирма *AMD* разработала и выпустила в 1991 г. МП, работающий с тактовой частотой 40 МГц.

Процессоры четвертого поколения. МП Intel 80486, появившийся в 1989 г., содержал уже 1,2 млн транзисторов и в течение долгого времени являлся стандартом для высококлассных компьютеров.

Различия между процессором 80386 и 80486 значительны. Преимущество в производительности нового МП перед предшественником определялось не столько более высокой тактовой частотой, сколько усовершенствованиями в архитектуре. МП 80486 имел расширенный набор из шести команд. В микросхему МП была встроена внутренняя кэш-память для данных и команд, управляемая внутренним контроллером. МП 80486 имел встроенный в микросхему математический сопроцессор, обеспечивавший эффективные вычисления с вещественными числами. Наконец, в МП была реализована конвейеризация вычислений, при которой каждая последующая команда начинала

выполняться сразу же после прохождения первой ступени конвейера предыдущей командой.

Фирма *Intel* последовательно совершенствовала этот МП, выпустив несколько модификаций, в которых работа самого МП осуществлялась с вдвое, а затем и вчетверо большей частотой, чем частота работы в материнской плате (эту возможность обеспечивает наличие внутренней кэш-памяти). Некоторое время он даже конкурировал с МП Pentium.

Процессоры пятого поколения. Первый МП этого поколения был выпущен в 1993 г. Он разрабатывался под кодовым номером *P*5 и должен был называться 80586, но впоследствии его стали называть Pentium. МП имеет 3,1 млн транзисторов. Повышение производительности МП пятого поколения связано со следующими архитектурными усовершенствованиями:

• получила дальнейшее развитие конвейеризация вычислений; количество конвейеров увеличилось до двух, а число ступеней одного конвейера — до пяти;

• МП обеспечивает хранение адресов последних 256 условных и безусловных переходов, что позволяет быстро сформировать адрес следующей выполняемой команды (так называемая технология предсказывания переходов);

• кэш-память у МП Pentium раздельная для команд и данных, по 8 Кбайт для каждой (у МП 80486 кэш-память — общая);

• МП имеет более совершенный математический сопроцессор, обеспечивающий 3—4-кратный выигрыш по скорости выполнения операций;

• МП имеет 64-битную шину данных (у 80486—32-битная). Был разработан ряд модификаций МП Pentium, у которых внутренняя тактовая частота в 1,5; 2; 2,5 и 3 раза больше, чем тактовая частота работы материнской платы. Постоянно совершенствовалась технология: если первый МП был изготовлен по 0,8 мкм технологии, последующие модификации — по 0,35 мкм технологии, то последние модели — по 0,25 мкм технологии.

Наивысшим достижением фирмы *Intel* в разработке МП пятого поколения явился МП Pentium *MMX*, выпущенный в 1997 г. Этот МП ориентирован на решение задач мультимедиа, требующих интенсивных операций с целыми числами. Подобные задачи решают игровые, коммуникационные, обучающие программы, которые используют графику,

аудио, трехмерное изображение, мультипликацию и т.п. По сравнению с первыми МП пятого поколения он отличается следующими изменениями архитектуры:

- вдвое большим размером внутренней кэш-памяти команд и данных (по 16 Кбайт);
- увеличенной на один шаг длиной конвейера (6 ступеней);
- новым блоком формирования адресов переходов;
- вдвое большим количеством буферов записи данных.

Производителем МП пятого поколения была не только фирма *Intel*. Фирмы *AMD*, *Cyrix*, *SGS*, *Thomson* и др. разработали свои собственные МП, в том числе и по технологии *MMX*.

Процессоры шестого поколения. В 1995 г. фирмой *Intel* был разработан МП 80686, получивший название Pentium *Pro*. В этом МП, содержащем 5,5 млн транзисторов, наряду со старыми были применены новые приемы повышения производительности.

Новый МП имеет не пять, а четырнадцать ступеней конвейера, число же самих конвейеров возрастает с двух до трех, применены новые приемы заблаговременного определения адресов переходов. МП обладает двухуровневой кэш-памятью. Кэш-память второго уровня, ранее размещавшаяся на материнской плате, теперь встроена в МП и может работать с тактовой частотой МП, в несколько раз большей тактовой частоты материнской платы. Кэш-память второго уровня значительно увеличивает производительность МП при реализации многозадачного режима работы. Но самым существенным нововведением стало использование технологии внеочередного исполнения команд (см. главу 5).

Дальнейшее совершенствование МП шестого поколения привело к появлению МП Pentium II (1997 г.), который сочетает в себе преимущества технологии Pentium *Pro* и *MMX*. Был выпущен целый ряд МП серии Pentium II, отличавшихся друг от друга все более высокой тактовой частотой (от 233 до 450 МГц).

Кроме того, велись работы по повышению тактовой частоты материнской платы. Если у первых образцов всего семейства МП Pentium она составляла 60—66 МГц, то у последних модификаций МП серии Pentium II — 100 МГц.

В 1999 г. на смену МП Pentium II пришел МП Pentium III. Его основное отличие от предыдущего заключалось в суще-

ственном расширении набора специальных команд для обработки целочисленных данных, необходимых для обработки изображений. Совершенствование технологии привело к переходу от 0,25 мкм технологии к 0,18 мкм и появлению ряда модификаций с более высокими тактовыми частотами, последней из которых была 733 МГц.

Больших успехов в разработке МП шестого поколения добилась фирма *AMD*. Некоторые ее процессоры семейства *K*-6 Athlon не только соответствовали МП семейства Pentium III по производительности, но и превосходили их.

Фирмами *Intel* и *AMD* велись разработки более дешевых МП шестого поколения. Упрощенным аналогом МП Pentium стал процессор Celeron, а МП Athlon — Duron. МП этих семейств предназначены для использования в обычных домашних компьютерах, поэтому они все поддерживают технологию *MMX*, но не имеют кэш-памяти второго уровня, которая дает существенный выигрыш в производительности лишь в многозадачном режиме и при использовании профессиональных операционных систем.

Процессоры седьмого поколения. Первый МП этого поколения был выпущен не фирмой *Intel*, а *AMD* в 1999 г. Это — первый процессор семейства *K*-7, выполненный по 0,22 мкм технологии. Этот МП работает с тактовой частотой 500 МГц и содержит 22 млн транзисторов. Главные отличия МП этого семейства заключаются в следующем:

- новая системная шина работает с тактовой частотой до 400 МГц (у МП шестого поколения — только 133 МГц);
- расширен объем кэш-памяти первого и второго уровней. Объем кэша команд и данных первого уровня составляет по 64 Кбайт, а второго уровня варьируется от 512 Кбайт до 8 Мбайт;
- процессор содержит по три конвейерных блока для выполнения операций с целыми числами и операций с вещественными числами. Число конвейеров, таким образом, возросло вдвое;
- расширен набор инструкций для работы с целыми числами и кэш-памятью.

Все это позволило существенно повысить производительность МП этого семейства по сравнению с МП семейства Pentium III.

МП седьмого поколения корпорации *Intel* — Pentium 4 стал серьезной модернизацией архитектуры процессора

Pentium *PRO*. Направления его модернизации были аналогичны тем, что и у МП *K*-7. Это переход на новую системную шину, расширение объемов кэш-памяти и числа конвейеров. Общее число ступеней конвейера у Pentium 4 стало составлять 20 (у *K*-7 только 10). Кроме того, применено усовершенствование, позволившее ускорить процесс выполнения команд. Если у МП *K*-7 команда, содержащаяся в кэш-памяти, декодируется каждый раз при ее выполнении, то МП Pentium 4 хранит уже декодированные команды, т.е. внутренние команды для узлов МП.

Рост производительности процессоров Pentium 4 осуществлялся за счет улучшения технологии и перехода на более низкую технологическую норму, увеличения тактовой частоты работы ядра процессора, объемов кэш-памяти и поднятия частоты системной шины.

Первый процессор Pentium 4 был реализован на ядре Willamette. Он был представлен 20 ноября 2000 г., изготовлен по технологии 0,18 мкм и содержал 42 млн транзисторов. Тактовая частота ядра составляла 1,4 ГГц, объем кэш-памяти — 265 Кбайт, частота системной шины — 400 МГц. В 2002 г. был выпущен новый процессор Pentium 4, в котором использовалось новое ядро Northwood. В нем был увеличен объем кэш-памяти до 0,5 Мбайт. Новый процессор производился по 130 нм технологии и содержал 55 млн транзисторов. Тактовая частота у первых процессоров Pentium 4 Northwood составляла 1,6 ГГц, в последующих моделях была доведена до 3,4 ГГц. Их производительность также повышалась за счет подъема частоты системной шины — 400, 533 и 800 МГц.

Компания *Intel* 2 февраля 2004 г. анонсировала первые процессоры Pentium 4, реализованные на новом ядре — Prescott. Процессоры были реализованы по 90 нм технологии и содержали 125 млн транзисторов. Если в предыдущих обновлениях сама структура процессора изменялась незначительно, то в новом ядре Prescott в процессорную архитектуру были внесены весьма значительные изменения. Тем не менее фирма *Intel* не стала заявлять о разработке нового поколения процессоров. Основной целью при разработке ядра Prescott был дальнейший рост частоты при уменьшенном энергопотреблении, и для этого конвейер был с 20 ступеней увеличен до 31. Увеличенный конвейер предъявлял более высокие требования к правильности предсказаний переходов, поскольку при ошибках возникала необходи-

мость в аннулировании всех команд, находящихся в конвейере. Поэтому в ядре Prescott был усовершенствован блок предсказания переходов, позволивший значительно сократить число таких ошибок. Кроме того, был добавлен блок умножения целых чисел, вдвое увеличен объем кэш-памяти. Процессоры Pentium 4 на ядре Prescott работали с тактовой частотой 2,4÷3,8 ГГц, частотой системной шины — 533 или 800 МГц.

Дальнейшее усовершенствование процессоров Pentium 4 привело к появлению у них 64-разрядной шины адреса, после чего процессоры, исходя из классификации, стали относиться уже к восьмому поколению.

Процессоры восьмого поколения. Пока фирма *Intel* занималась усовершенствованием своего процессора Pentium 4, фирма *AMD* разработала принципиально новые 64-разрядные процессоры, совместимые с системой команд всех предыдущих процессоров x86. Это процессоры *K*-8 Athlon и Opteron, появившиеся в 2003 г. Новая архитектура процессоров получила название x86—64.

МП, выполненный по 0,13 мкм технологии, стал иметь девять конвейерных блоков для операций с целыми и вещественными числами, а также три декодера команд, улучшенный блок предсказания ветвлений — до 16 000 адресов переходов. Длина конвейера по сравнению с *K*-7 была увеличена с 10 ступеней до 12. Удвоено число регистров общего назначения, ставших 64-битными. Одна из инноваций МП — встроенный контроллер памяти, который позволил существенно ускорить процессы обращения МП к ОЗУ.

Преимущества процессоров *K*-8 стали проявляться при работе с большими базами данных и системами САПР (*CAD*-системы). По сравнению с 32-битной адресацией в 4 Гбайт памяти, при 64-битной адресации компьютер получает в свое распоряжение 16 Тбайт. Сегодня большие базы данных уже значительно превысили объем в 4 Гбайт. Процессоры могут работать как в режиме совместимости с 32-битными программными приложениями, так и с 64-битными. Однако для полной реализации преимуществ МП *K*-8 потребовалась новая 64-битная операционная система Windows.

Очень важным преимуществом МП *K*-8 стала возможность организации высокопроизводительной многопроцессорной структуры благодаря новому трехшинному высоко-

скоростному интерфейсу HyperThransport, через который процессоры связываются между собой.

Процессоры Athlon 64 были предназначены для использования в персональных компьютерах, а Opteron — в серверах. Благодаря совместимости с системой команд x86 процессоры Opteron быстро заняли сектор производства серверов.

Отметим, что ранее, в мае 2001 г., фирма *Intel* совместно с фирмой *Hewlett Packard* разработала свой 64-разрядный процессор Itanium, предназначенный для использования в серверах, но он был несовместим с системой команд x86. Сама фирма *Intel* обозначила его как представителя новой архитектуры IA64 (*Intel Architecture* 64 *bit*). Однако этот проект был признан неудачным. Проблема заключалась в том, что огромное число самых различных программных комплексов и приложений было ориентировано именно на традиционную систему команд x86. Для эффективного использования процессора Itanium требовалась разработка специального программного обеспечения. Для того, чтобы процессор можно было использовать в системе команд x86, были разработаны программные средства адаптации. Хотя в режиме IA64 это был самый быстрый процессор для вычислений с плавающей точкой, в целочисленных вычислениях он лишь немного превосходил процессоры равной частоты с системой команд x86. А при выполнении неоптимизированного под процессор программного кода x86 его производительность была в восемь раз меньше, чем у традиционных процессоров на той же частоте. Вследствие этого продажи процессора оказались гораздо менее успешными, чем предполагалось.

Свой ответ в области 64-разрядных процессоров, совместимых с системой команд x86, фирма *Intel* сделала только весной 2004 г. — процессор Pentium 4F. Право использования архитектуры x86—64 фирма *Intel* получила на основе кросслицензионного соглашения с фирмой *AMD*, согласно которому обе фирмы должны передавать друг другу инновационные решения в области архитектуры и систем команд процессоров (это было выгодно обеим фирмам). Эта архитектура получила название EM64T — 64-разрядная технология с расширенной памятью (*Extended Memory 64 Technology*). В дальнейшем технология EM64T стала неотъемлемой частью всех последующих процессоров, выпускаемых фирмой *Intel*.

Процессоры девятого поколения. Прежде чем перейти к описанию процессоров девятого поколения, необходимо сделать небольшое отступление. В начале зарождения микропроцессорной техники поколения процессоров просматривались достаточно четко, при этом сами фирмы-разработчики, и в первую очередь *Intel*, указывали сведения о процессорах, которые позволяли однозначно отнести тот или иной процессор к соответствующему поколению. Если взять процессоры *Intel*, то это был ряд I8086, I80286, I80386, I80486. У фирмы *AMD* соответствующий процессорный ряд выглядел следующим образом: Am8086, Am80286, Am80386, Am80486. В самой цифре обозначения процессора были заложены сведения о его поколении. Процессоры пятого поколения стали последними, которые также однозначно можно было идентифицировать. У фирмы *Intel* это был процессор Pentium, который также обозначался как I80586, у фирмы *AMD* — процессоры *K*-5. Вплоть до пятого включительно поколения отличались четким набором усовершенствований.

Далее, с появлением процессора Pentium Pro фирма *Intel* отошла от четкой нумерации, и отнести тот или иной процессор к соответствующему поколению стало сложнее. Фирма *AMD* продолжила практику нумерации поколений своих процессоров — *K*-5, *K*-6, *K*-7, *K*-8.

К шестому поколению у фирмы *Intel* относились сразу целых три типа процессоров: Pentium Pro, Pentium II и Pentium III, так как, несмотря на отличия в производительности, тактовой частоте, объемах кэш-памяти, все они имели одну и ту же внутреннюю структуру и логику работы.

С седьмым поколением процессоров Intel сначала все было ясно, так как они были выполнены на основе принципиально новой микроархитектуры NetBurst (см. параграф 5.2). В процессе развития этой микроархитектуры создавались новые процессорные ядра, и хотя в них вносились некоторые усовершенствования, коренным образом микроархитектура не изменялась.

Фирма *AMD* первой выпустила процессоры восьмого поколения — 64-разрядные процессоры *K*-8. Свой ответ фирма Intel сделала, выпустив процессор Pentium 4F. Технология, которая обеспечила появление этого процессора — EM64T, впоследствии стала использоваться во всех процессорах Intel, но сама фирма *Intel* не стала идентифицировать ее как новое поколение процессоров. Более того,

усовершенствованная микроархитектура NetBurst в ядрах Prescott, в которой число ступеней конвейера было доведено до 31, также никак не дистанцировалась как новое поколение. Таким образом, если шестое поколение Intel пришлось на целых три типа процессоров, то в рамках процессоров Pentium 4 можно было бы выделить сразу два поколения.

Первые двухъядерные процессоры фирмы *Intel* выполнялись как на основе микроархитектуры, использованной в процессорах седьмого поколения Pentium 4, так и микроархитектуры, ставшей основой всех процессоров шестого поколения. Фирма *AMD* использовала для своего первого двухъядерного процессора микроархитектуру *K*-8. Сразу необходимо отметить, что первые многоядерные микропроцессоры обоих ведущих производителей выполнялись простым объединением двух кристаллов одноядерных процессоров в одном корпусе либо путем их совмещения в одном кристалле. Поэтому их с большим трудом можно считать процессорами нового поколения.

К девятому поколению будем относить многоядерные процессоры, для которых была разработана специальная многоядерная микроархитектура. Такими процессорами у фирмы *Intel* стала серия Intel Core 2, у фирмы *AMD* — серия Phenom, выполненная на базе микроархитектуры *K*-10.

Первые процессоры Intel Core 2 были представлены 27 июля 2006 г. Они были выполнены по 65 нм технологии. В основу их архитектуры была положена архитектура процессоров не седьмого, а шестого поколения (Pentium Pro). Путь развития за счет увеличения числа ступеней конвейера и тактовой частоты был признан тупиковым, так как это вело к возрастанию энергопотребления без заметного прибавления в производительности. Основными направлениями совершенствования в процессорах Intel Core 2 стали:

- эффективная работа исполнительных узлов за счет аппаратной реализации дополнительного набора команд;
- снижение энергопотребления;
- эффективная работа кэш-памяти;
- 64-разрядное адресное пространство;
- технология поддержки виртуальных машин (одновременное обслуживание процессором двух разных операционных систем);

• аппаратная поддержка технологии защиты информации (*La Grande Technology*);

• технология, позволяющая удаленно управлять настройками и безопасностью компьютера независимо от наличия питания и состояния операционной системы (*Active Management Technology*).

Intel вела одновременную разработку нескольких вариантов процессоров семейства Intel Core 2. Первыми появились двухъядерные процессоры Conroe и Merom, имеющие 4 Мбайта кэш-памяти и 291 млн транзисторов. Процессоры Conroe были предназначены для настольных систем (ПК) и выпускались с тактовыми частотами 1,86÷3 ГГц. Процессоры Merom были ориентированы на использование в мобильных системах (ноутбук), поэтому особое внимание при их разработке было уделено пониженному энергопотреблению, соответственно и тактовые частоты в них были ниже — 1,06÷2,4 ГГц.

Фирма *Intel* 13 декабря 2006 г. представила четырехъядерные процессоры Kentsfield, представлявшие собой по сути два процессора Conroe в одном корпусе. Ядра этих процессоров обладали теми же характеристиками, что и у Conroe, но сам процессор содержал уже 582 млн транзисторов.

В 2007 г. фирма *Intel* разработала новую технологию изготовления транзисторов, которая была реализована в процессорах семейства Intel Core. Они имели ту же архитектуру, но уже с более низкой технологической нормой — 45 нм. Был увеличен объем кэш-памяти и внесены некоторые улучшения. В частности, был реализован новый расширенный набор команд. Модели процессоров с двумя ядрами и 6 Мбайт кэш-памяти имели 410 млн транзисторов, а четырехъядерные модели с 12 Мбайт кэш-памяти — 820 млн транзисторов. Эту линейку процессоров, отличающихся различными тактовыми частотами (2,1...3,33 ГГц) и частотой системной шины (800, 1066 и 1333 МГц) фирма *Intel* выпускала в течение 2007 и 2008 гг.

Ответом фирмы *AMD* в области процессоров девятого поколения стали процессоры семейства Phenom, реализованные на базе микроархитектуры *K*-10. Они, как и первые процессоры семейства Intel Core, создавались на основе технологической нормы 65 нм и появились в продаже в конце 2007 г. Эти процессоры четырехъядерные, причем в отличие от всех своих предшественников, все четыре ядра

расположены на одном кристалле. Кроме того, в них внедрен целый ряд новых решений, во многом аналогичных внедренным в процессорах Intel Core:

- технология прямого соединения контроллера памяти и канала ввода/вывода с ядром процессора, обеспечивающая сокращение задержек, связанных с обращением к памяти;
- общая для всех ядер кэш-память третьего уровня объемом 2 Мбайта;
- усовершенствованный 128-битный блок обработки чисел с плавающей точкой;
- усовершенствованная технология *HyperTransport*;
- высокоскоростной интегрированный контроллер памяти;
- технология поддержки виртуальных машин;
- система управления питанием, обеспечивающая пониженное энергопотребление.

Процессоры десятого поколения. Первые процессоры десятого поколения были разработаны фирмой *Intel* в конце 2008 г. Это были процессоры Intel Core i7, реализованные на основе микроархитектуры Nehalem и выполненные на базе технологической нормы 45 нм с числом транзисторов не менее 731 млн. Они стали логичным развитием процессоров Intel Core, в них использовалась та же базовая архитектура ядра процессора, но вместе с тем был внедрен и целый ряд инноваций, основными из которых стали:

- встроенный контроллер памяти, поддерживающий несколько каналов для передачи данных;
- новая усовершенствованная высокоскоростная шина для обмена данными;
- наличие (у некоторых моделей) графического процессора;
- модульная структура, позволяющая конструировать на одном кристалле процессорные ядра разной конфигурации;
- усовершенствованная технология управления питанием, обеспечивающая эффективное энергосбережение;
- расширенный набор команд, ориентированный на широкий круг задач;
- наличие кэш третьего уровня;
- возможность реализации двух логических ядер на одном физическом.

В целом, процессоры десятого поколения Nehalem стали существенным шагом вперед, благодаря которому фирма Intel вновь стала занимать лидирующее положение в области разработки процессоров.

Следующим шагом в области совершенствования своих процессоров у *Intel* стал переход на новую технологическую норму 32 нм. Эти процессоры получили новое название Westmere и выпускались в течение всего 2010 г.

У фирмы *AMD* процессорами десятого поколения стало семейство Phenom II, первые представители которого появились в самом начале 2009 г. Это были четырехъядерные процессоры, выполненные по технологическому процессу 45 нм, Phenom II *X*4 (цифра 4 указывает на число ядер).

Процессоры используют вновь разработанное ядро Deneb (*Shanghai*), которое относят к микроархитектуре *K*-10.5. Процессор Phenom II *X*4 состоит из 758 млн транзисторов и имеет увеличенную кэш-память третьего уровня (2...6 Мбайт).

За счет оптимизации архитектуры и перехода на технологическую норму 45 нм фирме *AMD* удалось повысить тактовую частоту своих процессоров до 3 ГГц, обеспечив при этом более низкое энергопотребление, что позволило приблизиться к характеристикам процессоров Intel в части цены, производительности и энергосбережения.

Процессоры одиннадцатого поколения. Одиннадцатое поколение процессоров, созданных фирмой *Intel*, реализовано на основе новой микроархитектуры Sandy Bridge. Эта микроархитектура — дальнейшее развитие микроархитектуры Nehalem. Процессоры выполняются по технологии 32 нм. Количество ядер у различных модификаций процессоров может колебаться от 1 до 8. Как и Nehalem, Sandy Bridge имеет три уровня кэш-памяти, причем объем кэш-памяти третьего уровня у разных поредставителей может колебаться от 1—1,5 Мбайта (у простейших одноядерных процессоров Celeron) до 20 Мбайт (8-ядерные процессоры для серверов). Процессоры имеют интегрированные усовершенствованные двухканальный контроллер памяти, контроллер системной шины (образуют встроенный в чип так называемый северный мост набора системной логики) и графический контроллер.

Все элементы процессора (ядра, графический контроллер, кэш 3 уровня) объединены с помощью 256-битной межкомпонентной кольцевой шины. Шина состоит из четырех

32-байтных колец: шины данных, шины запросов, шины мониторинга состояния и шины подтверждения. Производительность кольцевой шины достигает 96 Гбайт в секунду на соединение при тактовой частоте 3 ГГц, что фактически в четыре раза превышает показатели процессоров предыдущего поколения.

В процессорах реализован новый расширенный набор векторных команд *AVX* (*Advanced Vector Extension*), который обеспечивает более высокое быстродействие в мультимедийных задачах и операциях с плавающей точкой. В процессорное ядро добавлена кэш-память инструкций нулевого уровня, которая содержит до 1500 декодированных микроопераций. Если обнаруживается, что выполняемая микрооперация в декодированном виде имеется в кэше, она берется оттуда. Это приводит к дополнительной экономии энергии и улучшает пропускную способность инструкций.

В целом процессоры Sandy Bridge имеют вдвое большую производительность, чем их предшественники Nehalem (при равной тактовой частоте).

На основе микроархитектуры Sandy Bridge также разработаны процессоры Ivy Bridge, которые имеют точно такую же структуру, но выполняются по технологии 22 нм.

У фирмы *AMD* представителем одиннадцатого поколения стал процессор Bulldozer. Он имеет полностью переработанную архитектуру по сравнению с предыдущими поколениями процессоров *AMD*, до 16 ядер, увеличенный объем кэш-памяти третьего уровня. Как и процессоры Sandy Bridge, они поддерживают набор команд AVX и имеют интегрированное в кристалл графическое ядро. В процессорах введена улучшенная технология передачи данных на основе четырехканальной шины *HyperTransport* 3.0.

4.8. Микроконтроллеры

Микроконтроллеры — *устройства, имеющие структуру микропроцессорной системы, выполненные в виде БИС и предназначенные для решения задач управления техническими устройствами и системами.* Поскольку в составе таких БИС имеются все узлы, присущие ЭВМ, они называются также однокристальными микроЭВМ.

Первые микроконтроллеры появились в 1976 г., когда в одной БИС стало возможным совместить процессор, ОЗУ, ПЗУ и порты ввода/вывода. В отличие от универсальных микроЭВМ в микроконтроллерах небольшие по размерам память и простой интерфейс для связи с внешними устройствами. Это объясняется прежде всего спецификой решаемых задач. Микроконтроллеры не используются в универсальных вычислительных системах, а предназначены для создания высокоэффективных и дешевых управляющих и регулирующих систем. Микроконтроллеры реализуют несложные алгоритмы, и для размещения программ им требуется емкость памяти на несколько порядков меньшая, чем у микроЭВМ универсального назначения. Внешние устройства также значительно проще, а их набор существенно уже. Микроконтроллеры применяются в устройствах управления приборов, бытовой аппаратуры, автомобилей, станков, в частности, в устройствах числового программного управления.

Первый микроконтроллер, выпущенный фирмой *Intel*, был 8-разрядным. В настоящее время выпускаются также 16- и 32-разрядные микроконтроллеры, но наибольшим спросом все равно пользуются 8-разрядные (рис. 4.18).

Емкость ОЗУ микроконтроллера может меняться от десятков до сотен байт. Она используется не для хранения программ, а только для временного хранения данных и промежуточных результатов, поэтому этой емкости для простых задач управления вполне достаточно. Для хранения программ служит флэш-память, емкость которой обычно не превышает 20 Кбайт. В микроконтроллере предусмотрена возможность расширения объема памяти за счет подключения внешних БИС ПЗУ и ОЗУ.

Управление работой всех узлов осуществляет центральный процессорный элемент (ЦПЭ). Генератор тактовых импульсов вырабатывает тактовые импульсы, синхронизирующие работу ЦПЭ. Их частота задается кварцем, подключаемым к внешним выводам микросхемы. ЦПЭ может работать с невысокой тактовой частотой (12÷24 МГц).

Для передачи адресов и данных между ЦПЭ и всеми остальными узлами микроконтроллера служит мультиплексная шина адреса/данных. Ее разрядность определяется разрядностью обрабатываемых слов в микроконтроллере.

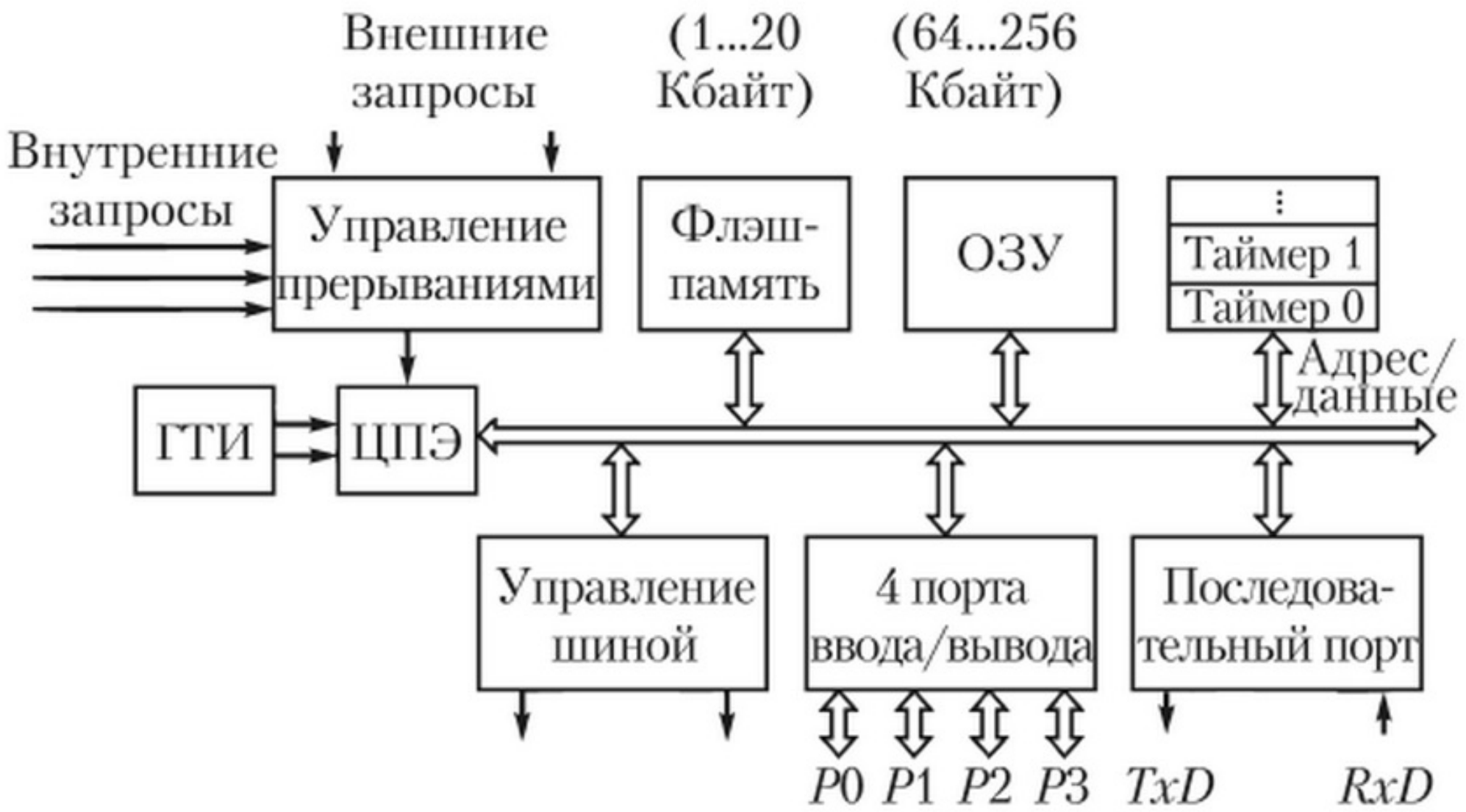


Рис. 4.18. **Типовая структура микроконтроллера**

Средства ввода/вывода микроконтроллера представлены четырьмя параллельными портами $P0 \div P3$ и последовательным портом ввода/вывода. Через порты $P0 \div P3$ осуществляется обмен информацией с внешними устройствами по 8-разрядным шинам, при этом каждый порт обеспечивает взаимосвязь с одним устройством. Для сокращения физической ширины интерфейса (числа внешних выводов микросхемы) функции линий параллельных портов совмещены, и в разных режимах имеют разное значение. Через вход *RxD* осуществляется прием данных в последовательном коде, а через выход *TxD* — передача данных.

Микроконтроллер может обслуживать до пяти запросов прерывания. При этом два запроса — собственно сигналы запроса прерывания от внешних устройств $\overline{INT0}$ и $\overline{INT1}$ и два прерывания формируются таймерами, а еще один — по сигналу на входе последовательного порта *RxD*. Таймеры-счетчики, число которых для разных микроконтроллеров колеблется от одного до трех, предназначены для отработки интервалов времени или подсчета внешних синхронизирующих сигналов. Они обеспечивают синхронизацию работы микроконтроллера в системе.

Система команд микроконтроллера специально ориентирована на решение задач управления, поэтому наряду с обычными командами, характерными для всех микропроцессоров, в ней есть и некоторые специфические. Поскольку все ресурсы микропроцессорной системы — ОЗУ, ПЗУ,

порты ввода/вывода — находятся внутри микросхемы, то ЦПЭ при выполнении команд не требуются циклы обращения к внешним по отношению к нему микросхемам. Поэтому большинство команд — короткие, аналогичные командам микропроцессора, в которых данные находятся в регистровой памяти. Время выполнения команд зависит от тактовой частоты и при частоте 12 МГц составляет 1 мкс, некоторые команды выполняются за 2 мкс.

4.9. Интерфейсные устройства

В процессе работы микропроцессорной системы постоянно происходит обмен информацией между различными устройствами системы с помощью так называемого *интерфейса*. «Интерфейс» — очень широкое понятие. В зависимости от области знаний под интерфейсом могут понимать процесс диалога человека с компьютером, программно-аппаратные средства для этого диалога; различают понятия «дружественного интерфейса», «пиктографического интерфейса», «системного интерфейса» и т.д. *В микропроцессорных системах под интерфейсом понимают совокупность аппаратных, программных и конструкторских средств, обеспечивающих информационный обмен между устройствами системы.* Это определение включает в себя три составляющие: аппаратную, программную и конструкторскую. К *аппаратной* составляющей интерфейса относят отдельные узлы микропроцессорной системы, через которые осуществляется обмен. К *программной* составляющей относят строгий порядок, алгоритм взаимодействия устройств, реализованный в виде специальных программ. Порядок следования информационных и служебных сигналов в процессе обмена информацией называют протоколом обмена. *Конструкторскую* составляющую образуют линии связи, объединенные в шины, по которым осуществляется передача информации. Каждая шина обеспечивает передачу какого-либо определенного вида информации, например адреса, данных или управляющих сигналов.

Параллельный и последовательный способы передачи информации. Приемники и источники информации могут находиться как в непосредственной близости, так и на значительном удалении. В зависимости от удаленности источника и приемника информации в микропроцессорных

системах могут использоваться либо *параллельный* способ передачи информации (*параллельный интерфейс*), либо *последовательный* (*последовательный интерфейс*). При параллельной передаче информация передается по шинам данных (магистралям), состоящим из *n* параллельных проводников.

Передача информации осуществляется целыми машинными словами, причем все разряды слова данных передаются одновременно. Такой способ передачи информации может использоваться, если длина линий связи между приемником и источником обычно не превышает полутора, а с применением специальных магистральных приемопередатчиков — нескольких метров. Это связано с тем, что на частотах, с которыми осуществляется передача информации (десятки мегагерц), проявляются волновые свойства электрических сигналов и проводников. В результате сигналы, передаваемые по разным проводникам шины, доходят до источника не одновременно, происходит «размывание» сигнала, и приемники такой информации становятся неприемлемо сложными и дорогими.

При последовательной передаче информация передается по одной линии связи разряд за разрядом. Для этого необходимо преобразование данных из параллельного кода в последовательный (при передаче) и обратное преобразование из последовательного кода в параллельный (при приеме). Расстояние, на которое может передаваться информация последовательным способом, уже не ограничено волновыми свойствами сигналов и проводников. Оно определяется лишь мощностью передатчика. В частности, во всех информационных сетях передача осуществляется в последовательном коде.

Прием и передача данных в устройствах микропроцессорной системы осуществляется через специальные буферные узлы, называемые *портами*. Порты могут быть параллельными и последовательными. Микропроцессоры, предназначенные для использования в системах управления, обычно имеют оба типа портов.

Синхронный и асинхронный способы обмена. Существует два основных способа обмена: синхронный и асинхронный. При *синхронном* обмене темп выдачи информации определяет источник, который сопровождает выдачу импульсами синхронизации. Приемник информации при этом должен принимать данные в темпе, задаваемом источ-

ником. Если приемник не успел принять информацию, она будет потеряна, так как между источником и приемником нет обратной связи. Поэтому темп передачи данных должен учитывать быстродействие приемника и рассчитывается на наихудший случай.

При *асинхронном способе* обмена между источником и приемником существует обратная связь. В основе этого способа лежит *метод квитирования*. Сущность метода в том, что источник при каждой посылке данных должен получать от приемника подтверждение о том, что данные приняты, т.е. своеобразную «квитанцию». Сигналы, обеспечивающие такой диалог между источником и приемником, называются сигналами квитирования. При посылке данных источник сообщает об этом приемнику сигналом «Данные выданы». Приемник, получив этот сигнал, считывает слово данных и выдает приемнику сигнал «Данные приняты». Только после получения этого сигнала источник приступает к посылке следующего слова данных. При таком обмене интервал времени приема—передачи данных будет переменным, в зависимости от быстродействия источника и приемника, длины линий связи.

Радиальная и магистральная структура интерфейсов. В интерфейсах с *радиальной структурой* каждое из устройств системы связано с центральным устройством, управляющим обменом (*концентратором*), через индивидуальную группу шин с одинаковым составом линий для каждого устройства, т.е. каждому устройству выделяется собственный набор шин. Концентратор получает заявки от устройств и соединяет их между собой. Если заявки поступают одновременно от нескольких устройств, концентратор определяет очередность обмена. Радиальные интерфейсы применяются для связи с удаленными внешними устройствами.

Если интерфейс одним и тем же набором шин обеспечивает работу множества устройств микропроцессорной системы во главе с микропроцессором, то его называют *системным*, а шины — *системными*. Набор системных шин, обеспечивающих работу интерфейса, называют *магистралью*. В интерфейсах с магистральной структурой все шины являются шинами коллективного пользования и к ним подключены все устройства системы. В микропроцессорных системах обычно используется магистральный интерфейс. Для реализации обмена к шинам одновременно может под-

ключаться только два устройства — приемник и источник. Порядок использования общей магистрали для организации обмена между множеством различных устройств определяется контроллером магистрали.

4.9.1. Интерфейсы и интерфейсные БИС

Как только микропроцессоры стали использоваться в качестве процессоров ЭВМ, возникла задача разработки БИС для реализации интерфейсных функций. Первые микроЭВМ были 8-разрядными, работавшими с невысокой тактовой частотой, и интерфейсы, обеспечивающие работу микропроцессорных систем, обладали соответствующими техническими характеристиками. Эти интерфейсы были, как правило, параллельными, магистральными, асинхронными, с 8-разрядной шиной данных и 16-разрядной шиной адреса. В некоторых интерфейсах была мультиплексная шина адреса/данных.

С ростом разрядности и быстродействия микропроцессоров изменялись и характеристики интерфейсов. Сначала использовались главным образом параллельные интерфейсы. Для 8- и 16-разрядных процессоров был разработан интерфейс (шина) ISA, который для 32-разрядных процессоров, начиная с 80386, был доработан, став шиной EISA. Для процессоров Pentium был разработан новый параллельный интерфейс PCI. Постепенно параллельные интерфейсы перестали удовлетворять, и им на смену пришли последовательные — PCI Express, USB, SATA.

Первые интерфейсные БИС были ориентированы на работу с 8-разрядной шиной данных и 16-разрядной шиной адреса. Простейшими микросхемами были шинные формирователи и буферные регистры (порты ввода/вывода). Более сложные операции обслуживались адаптерами и контроллерами. Каждое из таких устройств выполнялось в виде отдельной БИС. Сейчас уровень интеграции ИС позволяет в одном кристалле объединить целый ряд устройств, выполняющих различные интерфейсные функции. Однако в структурном плане современные интерфейсные ИС до сих базируются на «простых» ранее разработанных ИС.

В маркировке первых интерфейсных БИС первыми были цифры 82, после которых стояли еще две цифры, обозначающие вид конкретной схемы. При описании функциональных возможностей и структуры современных

интерфейсных БИС обычно идет перечисление ранее разработанных БИС семейства 82XX, структурно входящих в современную БИС. Например, о современном периферийном контроллере 82C206 сказано: содержит две ИС 8259, две ИС 8237, одну ИС 8254 и др. Более того, даже в библиотеках схемных решений новейших СБИС программируемой логики присутствуют структуры традиционных БИС 82XX.

4.9.2. Шинные формирователи и буферные регистры

Шинные формирователи. Шинные формирователи включаются между источником информации и шиной. Так как к шине может подключаться множество устройств, сигналы, выдаваемые в шину, должны быть достаточно большой мощности, которую обычная БИС обеспечить не в состоянии. Шинные формирователи усиливают сигналы источника по мощности, отключают источник от шины, когда он не участвует в обмене, формируют при необходимости требуемые уровни сигналов лог. 1 и лог. 0. *Двунаправленные шинные формирователи* позволяют не только передавать сигналы источника в шину, но и принимать их с шины и передавать приемнику данных. Работа ИС определяется управляющими сигналами. Так, для двунаправленного формирователя требуются два сигнала управления. Первый — разрешающий прием/передачу данных или отключающий от шины путем перевода выходных усилителей в третье состояние. Второй — указывающий направление передачи данных: в шину или с шины. Обычно в двунаправленном шинном формирователе по каждому разряду работает два усилителя. Один усиливает сигнал источника и выдает его в шину, другой принимает сигнал этого же разряда с линии шины и выдает усиленный сигнал приемнику. При этом мощность первого усилителя, нагруженного на шину, обычно больше второго, нагрузкой которого является одна ИС. Так как шина обладает емкостью, с которой элементы, выполненные по МОП-технологии справляются плохо, ИС шинных формирователей выполняются по биполярной технологии ТТЛШ.

Буферные регистры. Буферные регистры служат для подключения к магистрали внешнего устройства. В отличие от шинных формирователей, которые только усиливают сигналы, буферные регистры способны хранить

данные. Благодаря этому они могут выполнять функцию портов. Буферные каскады с тремя состояниями на выходах регистра обеспечивают портам возможность отключения от магистрали под действием управляющих сигналов, а также необходимую нагрузочную способность.

Через порты ввода данные от внешнего устройства поступают в магистраль, а через порты вывода данные с магистрали передаются тому или иному модулю. Порты ввода/вывода могут выполнять обе указанные операции.

4.9.3. Параллельные периферийные адаптеры

Шинные формирователи и порты обеспечивают лишь непосредственную передачу данных между системной шиной и устройством (МП или ВУ). Более сложные операции выполняют периферийные адаптеры. Эти устройства могут выполнять разнообразные функции обмена данными, причем режим работы устройства задается путем их программирования командами МП.

Для связи с системной ШД ВУ, работающих с параллельными кодами, используются контроллеры ввода/вывода параллельной информации — *параллельные периферийные адаптеры.*

Таким адаптером является БИС Intel 8255A (российский аналог К580 ВВ55А). Адаптер (рис. 4.19) подключается к 8-разрядной системной ШД через канал данных. Этот канал имеет три состояния, благодаря чему адаптер может отключаться от системной ШД. Адаптер содержит три 8-разрядных порта *A*, *B* и *C*, причем порт *C* может разделяться на два 4-разрядных канала. Каждый порт имеет три состояния, и через них осуществляется подключение ВУ к системной ШД. Направление передачи (из МП в ВУ или наоборот) программируется МП. Управление работой всех узлов адаптера осуществляет УУ, которое получает управляющие сигналы от МП. К ним относятся сигналы «Чтение» (*RD*), «Запись» (*WR*), «Выборка кристалла» (*CS*), «Сброс» (*Reset*), а также два разряда адреса для выбора порта. Программирование адаптера заключается в том, что МП заносит в УУ специальную команду — управляющее слово, которое определяет режим работы портов, а также направление передачи информации.

Адаптер может работать в трех режимах:

- синхронного обмена под управлением МП;

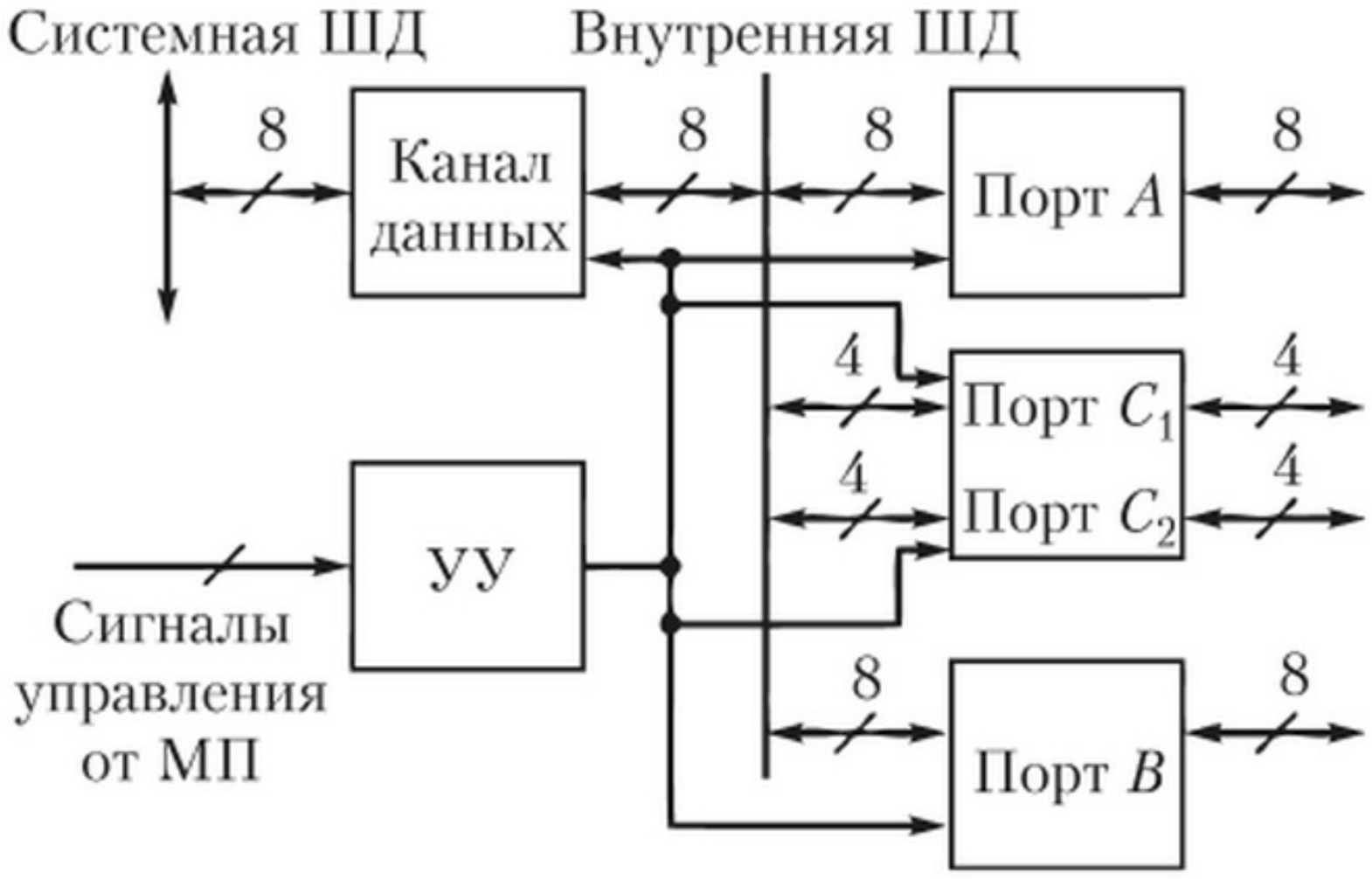


Рис. 4.19. **Структура параллельного интерфейсного адаптера**

• асинхронного однонаправленного обмена, сопровождаемого сигналами квитирования;

• асинхронного двунаправленного обмена.

Наибольшее число ВУ, обслуживаемых адаптером, обеспечивает первый режим. В этом режиме осуществляется программно-управляемая передача данных по четырем независимым каналам: портам A, B, C_1 и C_2. Обмен осуществляется со скоростью, определяемой процессором. Сигналы квитирования, подтверждающие, что обмен происходит без сбоев, не передаются. Адаптер может обслуживать два ВУ с 8-разрядным словом обмена и два с 4-разрядным словом. Направление передачи для каждого ВУ фиксировано и определяется управляющим словом, заносимым в адаптер перед началом обмена. В процессе работы МП двумя разрядами адреса выбирает порт, через который осуществляется ввод или вывод данных.

Во втором режиме происходит асинхронный обмен данными между МП и ВУ. Для этого необходимы линии для передачи сигналов квитирования, которые создаются из линий портов C_1 и C_2. В результате образуются два 12-разрядных канала: портам A и B, через которые осуществляется передача данных, дополнительно передается по три линии портов C_1 и C_2, через которые происходит обмен сигналами квитирования. Таким образом, в этом режиме адаптер может обслуживать два ВУ. Направление обмена, как и в первом режиме, определяется управляющим словом. Поскольку для одного сигнала квитирования требуется

один разряд, в этом режиме возможна индивидуальная установка в «0» и «1» каждого из разрядов портов C_1 и C_2.

Адаптер является приемопередатчиком между МП и ВУ, поэтому сигналы квитирования используются для организации диалога адаптера как с ВУ, так и с МП. Вот, например, какие сигналы квитирования нужны при передаче данных из ВУ в МП. ВУ после выдачи слова данных выдает сигнал «Данные выданы». Чтобы ВУ не выдавало нового слова данных до тех пор, пока данные не запишутся в МП, адаптер выдает ВУ сигнал «Буфер заполнен» и записывает слово данных в порт. После этого адаптер выдает в МП сигнал прерывания «*INT*». МП, получив этот сигнал, переходит к выполнению подпрограммы обслуживания прерывания, в которой имеется команда ввода данных из порта. Получив эту команду, адаптер снимает сигнал прерывания, а по окончании сигнала «*RD*», свидетельствующего о завершении чтения слова данных, снимает сигнал «Буфер заполнен». После этого ВУ может переходить к посылке следующего слова. Аналогичные три сигнала квитирования требуются и для пересылки данных из МП в ВУ.

Для режима двунаправленного асинхронного обмена необходимы уже не три, а пять сигналов квитирования. Поэтому в этом режиме адаптер может обслуживать только одно ВУ. Для передачи слова данных используется порт *A*. Для передачи сигналов квитирования ему выделяются пять линий порта *C*.

Параллельный интерфейсный адаптер используется для связи микропроцессорной системы с удаленными не более чем на 15 м ВУ. Такая связь обычно осуществляется через радиальный параллельный интерфейс ИРПР. Шина данных этого интерфейса может быть 8- или 16-разрядной. Аппаратные средства для реализации этого интерфейса могут быть построены с использованием рассмотренной БИС или ее более современных аналогов.

4.9.4. Программируемые связные адаптеры

При увеличении расстояний, на которые передаются данные, параллельные связи уже не применяют. Передачу данных осуществляют в последовательном коде по одной сигнальной линии. Для аппаратной реализации этого способа передачи данных используют контроллеры последовательного ввода/вывода, называемые *программируемыми связными адаптерами* (ПСА). Тракт передачи последова-

тельных данных в общем случае включает в себя источник и приемник данных, ПСА и модемы (рис. 4.20). Адаптер, подключенный к системной ШД, осуществляет преобразование данных из параллельной формы в последовательную или наоборот.

Модем (модулятор-демодулятор) преобразует двоичные импульсные сигналы в некоторый аналоговый модулированный сигнал, приспособленный к передаче по узкополосным (≈3 кГц) телефонным линиям. Если по такой линии передавать простую последовательность импульсов, соответствующих единицам и нулям, скорость передачи информации будет очень низкой. С помощью различных видов модуляции получают сигнал с более высокой информационной емкостью. Так, современные модемы обеспечивают передачу не менее 38,4 Кбит/с. Если расстояние между источником и приемником не настолько велико, чтобы использовать телефонный канал, то передача последовательных данных осуществляется непосредственно от одного адаптера до другого.

Различают два основных режима последовательной передачи данных: синхронный и асинхронный. В обоих режимах работа приемника и передатчика синхронизируется тактовыми сигналами синхронизации одной частоты. В *синхронном режиме* поток передаваемых символов непрерывный. Для того чтобы приемник мог выделить из потока символов полезную информацию (данные), передатчик в поток информации вводит специально закодированные слова — определенную последовательность единиц и нулей. Такие слова называют *синхросимволами*. Скорость передачи информации определяется частотой синхронизации. Если между словами данных имеются временные промежутки (данные для передачи еще не готовы), они заполняются синхросимволами до тех пор, пока не возобновится передача данных. При синхронной передаче данных приемник находится в состоянии активного ожидания. Он сравнивает каждое принятое слово с символом синхронизации. Как только такое слово будет выделено, приемник продолжает сравнение принимаемых слов с синхросимволом, при



Рис. 4.20. **Структура последовательного тракта передачи данных**

Содержание

этом все слова, отличающиеся от него, считаются полезной информацией и считываются.

В *асинхронном режиме* информационный поток делится на отдельные посылки, соответствующие словам данных. При отсутствии посылок на линии устанавливается единичный уровень. Каждая посылка начинается с нулевого *старт-бита*, затем следуют биты *слова данных* (информационные биты), за ними — *бит четности*. Заканчивается посылка 1; 1,5 или 2 единичными *стоп-битами* (рис. 4.21).

В зависимости от числа единиц в передаваемом слове в бите четности передают «0» или «1» таким образом, чтобы общее число единиц, передаваемых в посылке (не считая старт- и стоп-битов) было четным. Это обеспечивает приемнику возможность проверки достоверности передаваемых данных. Приемник подсчитывает количество единиц в слове, и если оно четно, значит информация в тракте передаче не исказилась. Аналогичным образом может осуществляться контроль достоверности и на нечетность количества единичных битов в посылке.

Выделение информации в посылке осуществляется следующим образом. Тактовая частота, которой синхронизируется работа источника и приемника, в 16 раз выше частоты передачи данных, так что на один бит приходится 16 тактовых импульсов. При поступлении нулевого старт-бита приемник запускает счетчик импульсов и с его помощью определяет моменты, соответствующие серединам информационных битов. В эти моменты определяются значения передаваемых битов. Так как число битов в слове данных известно заранее, после приема всех битов проверяется наличие и длительность единичного стоп-бита. Затем принятое слово записывается в параллельный регистр и проверяется на четность/нечетность. После этого параллельное слово данных готово для дальнейшей работы.

ПСА выполняются в виде БИС. Такая БИС есть и в семействе интерфейсных ИС Intel — 8251A (российский

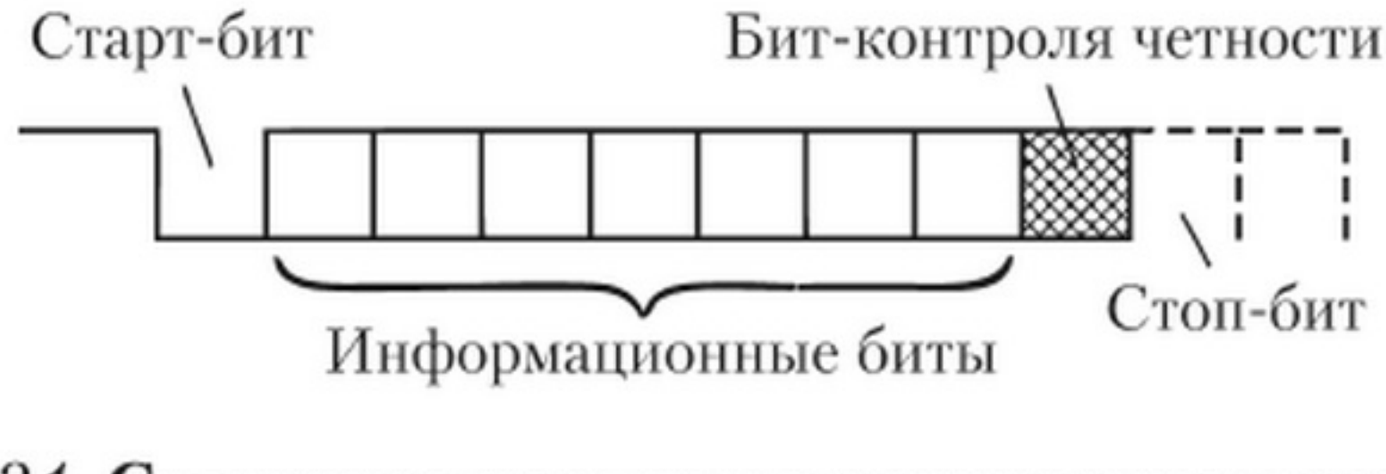


Рис. 4.21. **Структура посылки в асинхронном режиме передачи данных**

Содержание

аналог — К580 ВВ51А). Этот адаптер обеспечивает прием от МП 8-разрядного параллельного кода и преобразование его в последовательный, а также обратное преобразование последовательного потока символов в параллельный код.

Работа адаптера начинается с его программирования. При этом в него заносится следующая информация:

- прием или передача данных;
- режим обмена (синхронный или асинхронный);
- наличие контроля достоверности и его вид (четность или нечетность);
- длина слова данных (5, 6, 7 или 8 бит);
- вид синхросимвола и длина стоп-бита (для асинхронного обмена).

На рис. 4.22 показана функциональная схема взаимодействия при асинхронной передаче двух адаптеров, один из которых является передатчиком последовательных данных, а другой — приемником. По системной ШД адаптер-передатчик принимает слово данных и формирует посылку, присоединяя старт-бит, бит четности и стоп-бит. Готовая к отправке посылка помещается в регистр сдвига. Как только от приемника приходит сигнал о его готовности к приему данных, УУ запускает генератор, который «выталкивает» биты передаваемого слова в линию передачи. После передачи стоп-бита УУ останавливает генератор.

УУ адаптера-приемника по старт-биту запускает генератор и начинается прием слова данных в регистр сдвига. Определение значения очередного принимаемого бита осуществляется по его середине. Когда время, отведенное на прием одного бита, истекает, происходит сдвиг регистра. После приема всех битов принятое слово помещается в выходной буферный регистр, откуда оно может быть считано ВУ или другим МП. Для этого адаптер должен сообщить приемнику информации о своей готовности к выдаче слова данных. Теперь адаптер-приемник может приступать к приему следующего слова данных, о чем он должен сообщить адаптеру-передатчику.

На рис. 4.22 показаны упрощенные структурные схемы адаптеров. В действительности каждый адаптер имеет раздельные линии приема и выдачи последовательных данных, причем каждая линия обслуживается своей схемой. Кроме того, для работы с модемами адаптеры имеют соответствующие схемы управления и линии, по которым передаются сигналы квитирования.

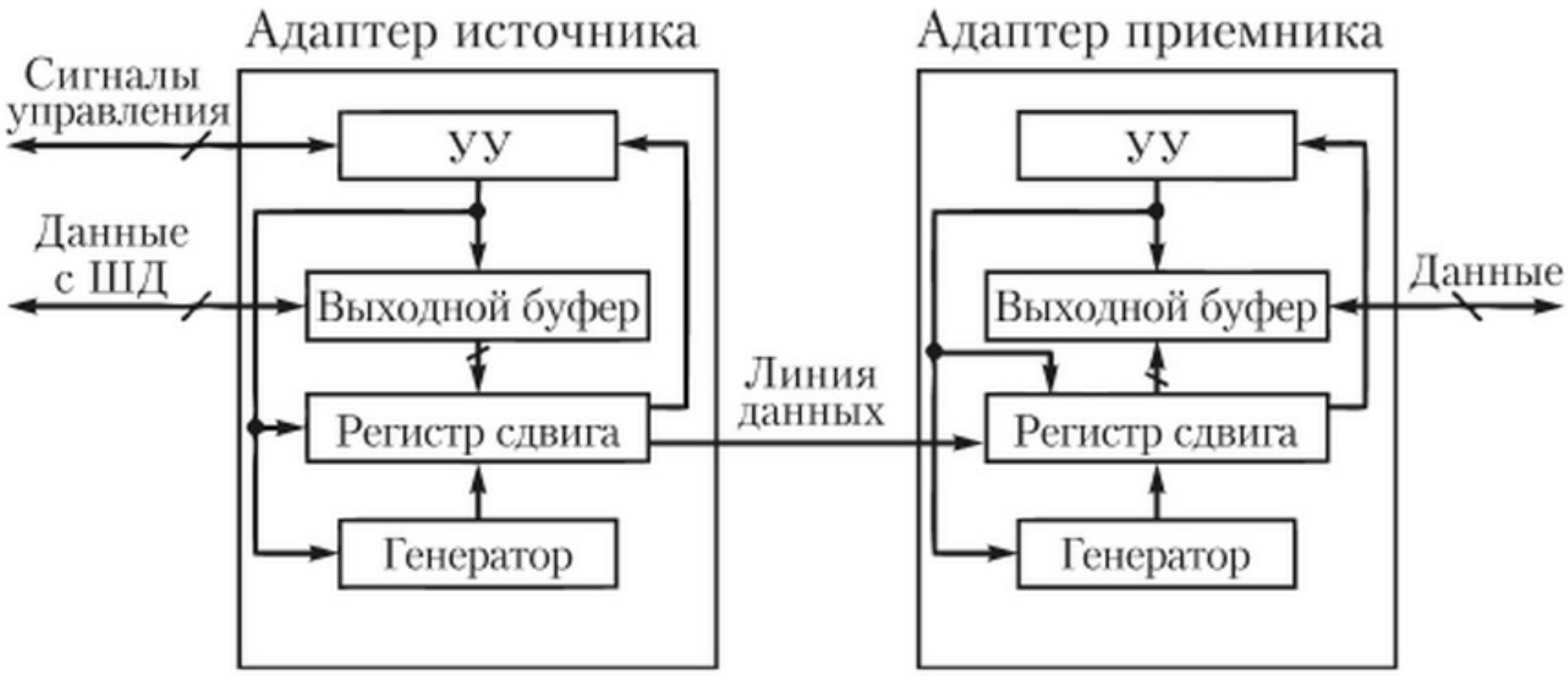


Рис. 4.22. **Функциональная схема работы двух адаптеров в режиме асинхронной передачи последовательных данных**

4.10. Программируемые контроллеры

4.10.1. Программируемые контроллеры прерываний

При работе микропроцессорной системы часто возникают ситуации, требующие немедленной реакции системы. При этом МП прерывает выполнение основной программы и переходит к обслуживанию ВУ, являющегося причиной возникновения этой ситуации. Обмен информацией между ВУ и МП осуществляется по сигналам-запросам прерывания, поступающим от ВУ. Для работы с прерываниями у МП есть два сигнала: *INT* — сигнал запроса прерывания от ВУ, являющийся входным для МП, и *INTA* — сигнал подтверждения прерывания, который выдает МП внешнему устройству, если МП готов к его обслуживанию. В микропроцессорной системе может быть множество различных ВУ, каждое из которых может обращаться с запросом прерывания. Для эффективной работы с прерываниями создаются специальные контроллеры прерываний, обеспечивающие обслуживание ВУ. Эти устройства выполняют всю необходимую работу для организации взаимодействия с ВУ. При этом в начале работы МП осуществляет их программирование, а в процессе работы непосредственно взаимодействует только с контроллером. Таким устройством является БИС Intel 8259A (российский аналог — К1810 ВН59). Этот контроллер может обслуживать запросы прерывания от восьми ВУ, а девять каскадно включенных контроллеров — от 64 ВУ.

При работе микропроцессорной системы запросы прерывания от разных ВУ могут накладываться друг на друга. Для разрешения таких конфликтных ситуаций вводится система приоритетов. При этом ВУ выстраиваются в порядке важности: чем важней ВУ для системы, тем выше его приоритет. Если во время выполнения программы обслуживания прерывания от ВУ с низким приоритетом появляется запрос от ВУ с более высоким приоритетом, МП прерывает выполнение программы и переходит к обслуживанию более приоритетного ВУ. Эта процедура может повторяться несколько раз, если запросы прерывания более приоритетных ВУ следуют один за другим. Такое обслуживание прерываний называется *вложением прерываний*.

Контроллер прерывания (рис. 4.23) через буфер ШД подключается к системной ШД. Все управляющие сигналы, обеспечивающие процессы записи/чтения (ЗП/ЧТ), поступают из МП. Взаимодействие контроллера с ВУ происходит с помощью входов *IR*, на которые подаются сигналы запросов прерывания от ВУ. Сигналы *INT* и *INTA* обеспечивают собственно режим прерывания работы МП.

Работа контроллера прерывания начинается с его инициализации. Контроллер получает команды инициализации по системной ШД от МП. При этом выполняются следующие действия:

- контроллер устанавливается в исходное состояние, в котором все его регистры обнулены;

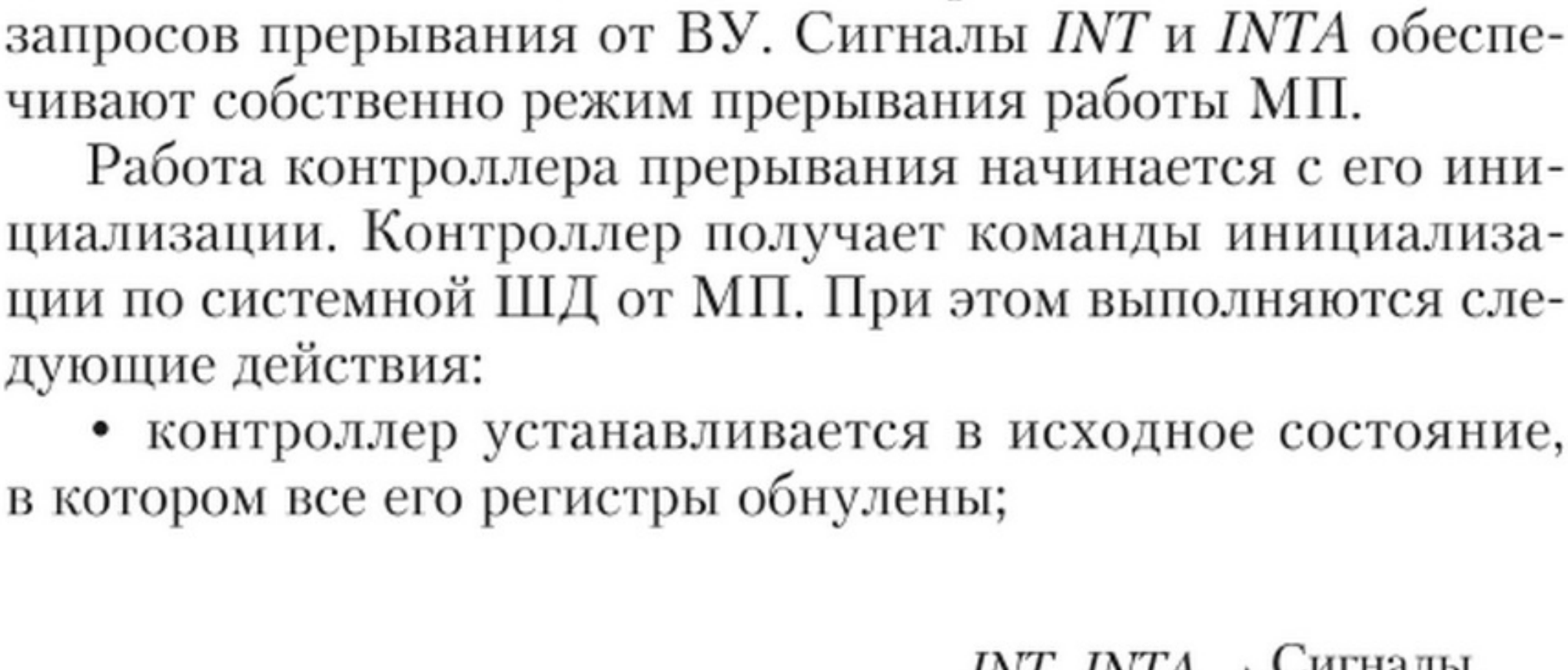


Рис. 4.23. **Структурная схема контроллера прерываний**

Содержание

• устанавливается исходный порядок приоритетов, при котором наивысший приоритет имеет вход IR_0, а самый низкий — вход IR_7;

• определяются начальные адреса программ обслуживания прерываний, причем процессор заносит адрес программы только для запроса IR_0, остальные адреса определяются автоматически, так как располагаются в ОЗУ с интервалом 4 или 8 байтов относительно этого адреса.

Сигналы запросов прерывания поступают в контроллер и запоминаются в регистре запросов прерывания: каждому запросу соответствует свой разряд. Среди полученных запросов выделяется самый приоритетный, который фиксируется в регистре обслуживания запросов. Затем контроллер посылает в МП сигнал прерывания работы процессора *INT*, и если МП отвечает сигналом *INTA* (подтверждение прерывания), он формирует так называемый вектор прерывания — начальный адрес программы обслуживаемого прерывания. МП считывает этот адрес и приступает к выполнению программы.

Один контроллер способен обслуживать не более восьми прерываний. Для расширения числа обслуживаемых запросов применяется каскадное включение контроллеров. При этом используются сигналы каскадирования. При таком включении микросхемы контроллеров разделяются на ведущую и ведомые. Ведущая микросхема собирает запросы прерывания со всех ведомых и взаимодействует с процессором. Входы *IR* ведущей микросхемы подключаются к выходам *INT* ведомых. Адрес программы обслуживания прерывания формирует ведомая микросхема, а выбор микросхемы, передающей адрес процессору, осуществляет ведущая. Для этого код ведомой микросхемы передается по линиям каскадирования.

При работе микропроцессорной системы возможны ситуации, когда какие-то из ВУ временно могут не обслуживаться. Для того чтобы исключить прерывание работы процессора по запросам таких ВУ, МП запрещает прохождение от них сигналов запросов прерывания. Это делается с помощью специальных команд *маскирования прерываний*. Маска прерываний — это 8-разрядное слово, каждый разряд которого соответствует одному из сигналов *IR*. Единица в разряде слова указывает на то, что прохождение соответ-

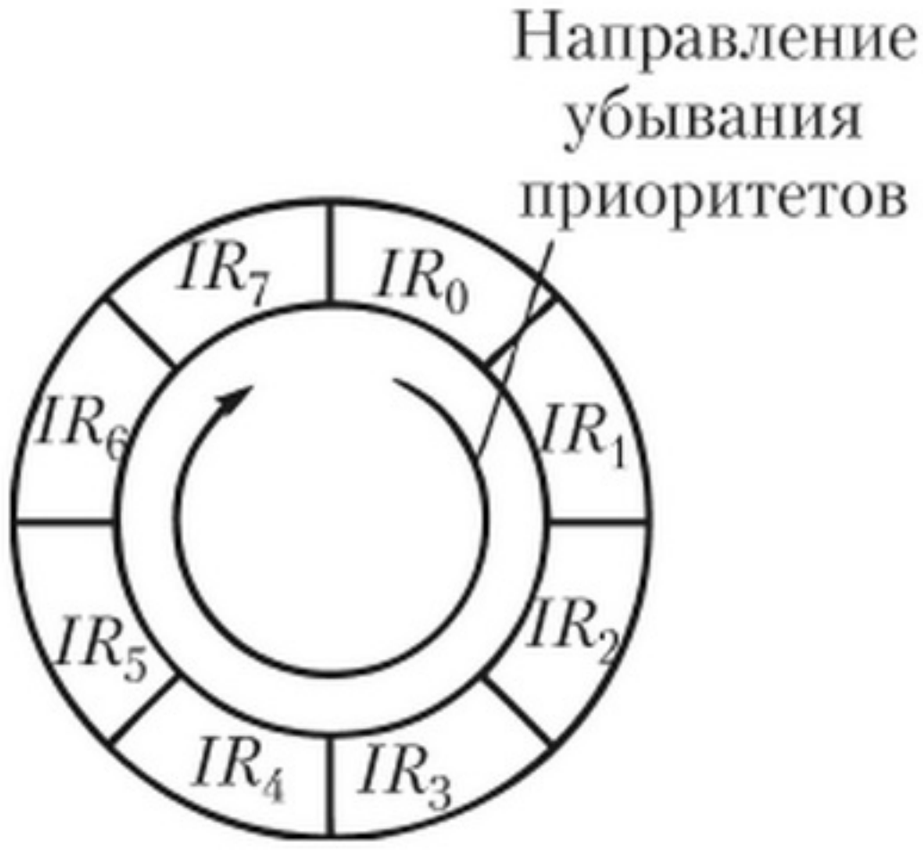


Рис. 4.24. **Приоритетное кольцо**

ствующего запроса прерывания должно быть заблокировано. В процессе работы МП заносит в контроллер маску прерываний, которая фиксируется в регистре схемы маскирования и анализа запросов. Если теперь в контроллер поступит запрос прерывания, прохождение которого запрещено, он зафиксируется в регистре запросов, но в регистр обслуживания не попадет и прерывания не вызовет. После снятия запрета, которое также осуществляется с помощью маски, запрос прерывания поступит в регистр обслуживания и вызовет прерывание работы процессора.

В процессе работы возможно также изменение порядка приоритетов сигналов *IR*. Это осуществляется с помощью установки дна приоритетного кольца (рис. 4.24). В исходном состоянии самый низкий приоритет у сигнала IR_7 и дно приоритетного кольца соответствует этому сигналу. При назначении другого сигнала в качестве дна приоритетного кольца, например IR_3, все приоритеты смещаются циклически, т.е. сигнал IR_4, следующий за IR_3, будет теперь иметь наивысший приоритет. За ним приоритеты расположатся следующим образом: IR_5, IR_6, IR_7, IR_0, IR_1, IR_2, IR_3.

Процессор может также назначать динамическую установку приоритетов. В этом случае дном приоритетного кольца каждый раз становится последний по времени обслуживания сигнал *IR*.

Рассмотренный режим работы контроллера называется режимом векторного прерывания, в котором ВУ является инициатором прерывания работы процессора. Возможно также обслуживание ВУ по инициативе процессора. В этом

случае контроллер осуществляет только прием сигналов прерывания, их фиксацию и выделение наиболее приоритетного запроса. Вся информация о запросах хранится в слове состояния контроллера. Процессор сам опрашивает контроллер и по слову состояния определяет, есть ли запросы прерывания и от каких ВУ.

4.10.2. Контроллеры прямого доступа к памяти

Режим непосредственного обмена данными между памятью и ВУ осуществляется под управлением контроллеров ПДП. ПДП особенно удобен при передачах блоков данных в высоком темпе, например, между внешней памятью и ОЗУ.

Для реализации этого вида обмена разработаны специальные БИС. Одной из таких БИС является Intel 8237A (российский аналог К580 ВТ57). Взаимодействие блоков микропроцессорной системы при ПДП показано на рис. 4.25. МП может выполнять программирование контроллера, настраивая его на определенный режим работы, и чтение его состояния. При ПДП контроллер вырабатывает сигналы управления обменом данными между ОЗУ и ВУ. Этот обмен осуществляется непосредственно через системные ША и ШД, при этом МП от них отключается, переводя свои буферные регистры в третье состояние.

Контроллер содержит четыре канала, к которым подключаются ВУ. Каждый канал обеспечивает работу ВУ в режиме ПДП по его запросу. При поступлении запроса контроллер обращается к МП с сигналом *HOLD* (запрос захвата шин). Если МП отвечает сигналом *HLDA* (подтверждение захвата), контроллер приступает к управлению обменом.

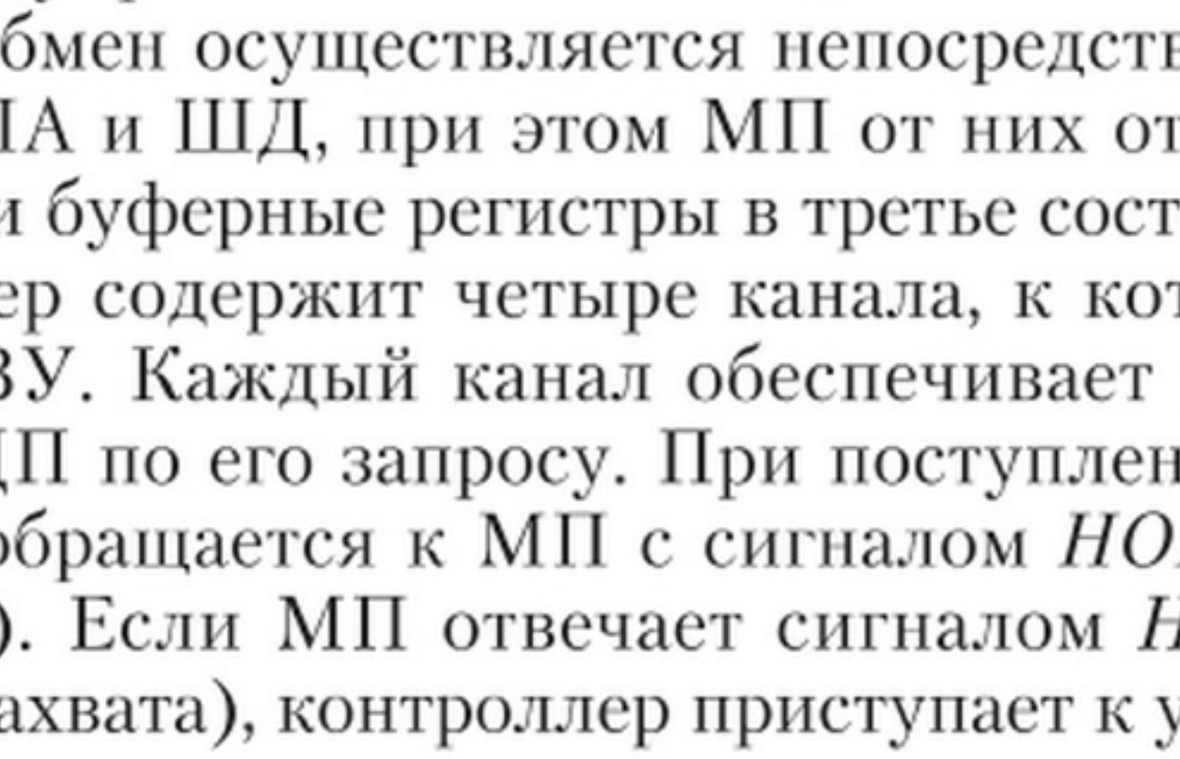


Рис. 4.25. **Взаимодействие блоков микропроцессорной системы при ПДП**

В начале работы МП осуществляет программирование контроллера. При этом для каждого канала заносится следующая информация:

- начальный адрес области памяти, с которой начнется обмен;
- общее число пересылаемых байтов;
- направление обмена (из ВУ в ОЗУ или наоборот).

Пересылка данных осуществляется побайтно, причем данные располагаются в памяти последовательно. Поэтому для формирования адресов ячеек памяти достаточно регистра адреса памяти и счетчика. В регистр адреса памяти заносится начальный адрес, а в счетчик — общее число пересылаемых байтов. При каждой передаче байта содержимое регистра памяти увеличивается или уменьшается на единицу (это определяется при программировании канала), поэтому к моменту окончания пересылки очередного байта уже готов адрес для следующего. Если запрограммирован декремент счетчика, то как только счетчик обнуляется, обмен заканчивается. Счетчик количества пересылаемых байтов 16-разрядный, поэтому максимальный объем блока данных, передаваемых за один запрос ПДП, составляет 64 Кбайт.

Так же как и при обслуживании прерываний, при обслуживании ПДП существует система приоритетов и маскирования запросов. Во время работы МП может запретить обслуживание какого-либо ВУ, выдав соответствующую маску. Система приоритетов в контроллере может быть фиксированной и циклической. При фиксированной системе наивысший приоритет у канала с номером «0», а самый низкий — с номером «3». В циклической системе приоритеты динамически изменяются по приоритетному кольцу. Самый низкий приоритет присваивается последнему обслуженному каналу, следующий за ним канал получает наивысший приоритет, а оставшиеся два канала — более низкие.

4.10.3. Программируемые интервальные таймеры

Работа микропроцессорной системы требует синхронной работы всех ее устройств, при этом их быстродействие может колебаться в самых широких пределах. Для обеспечения такой работы применяются программируемые интервальные таймеры. Эти устройства выполняют операции по выработке временных задержек, формированию

частот и интервалов времени. Для аппаратной реализации таймеров разработаны специальные БИС, подключаемые к системным шинам и программируемые процессором. Такие БИС обычно имеют несколько каналов, работа каждого из которых осуществляется независимо от других (рис. 4.26).

Каждый канал имеет собственный вход частоты синхронизации f_c и вход запуска. Преобразование входной частоты в сигнал с необходимыми временными характеристиками осуществляется с помощью счетчика. Перед началом работы в счетчик канала загружается число n. Каждый импульс, пропущенный на вход канала, уменьшает его содержимое на единицу. Обнуление счетчика является сигналом для формирования выходного импульса.

Возможны шесть режимов работы канала. В *режиме терминального счета* осуществляется формирование одиночного выходного импульса. Передний фронт формируется сразу после программирования канала и записи в счетчик числа. С приходом сигнала запуска импульсы частоты начинают поступать в счетчик, и при его обнулении формируется задний фронт выходного импульса.

В *режиме ждущего мультивибратора* формирование переднего фронта выходного импульса происходит только после прихода сигнала запуска, а заднего — при обнулении счетчика. В отличие от первого режима такой импульс будет формироваться каждый раз с приходом сигнала запуска.

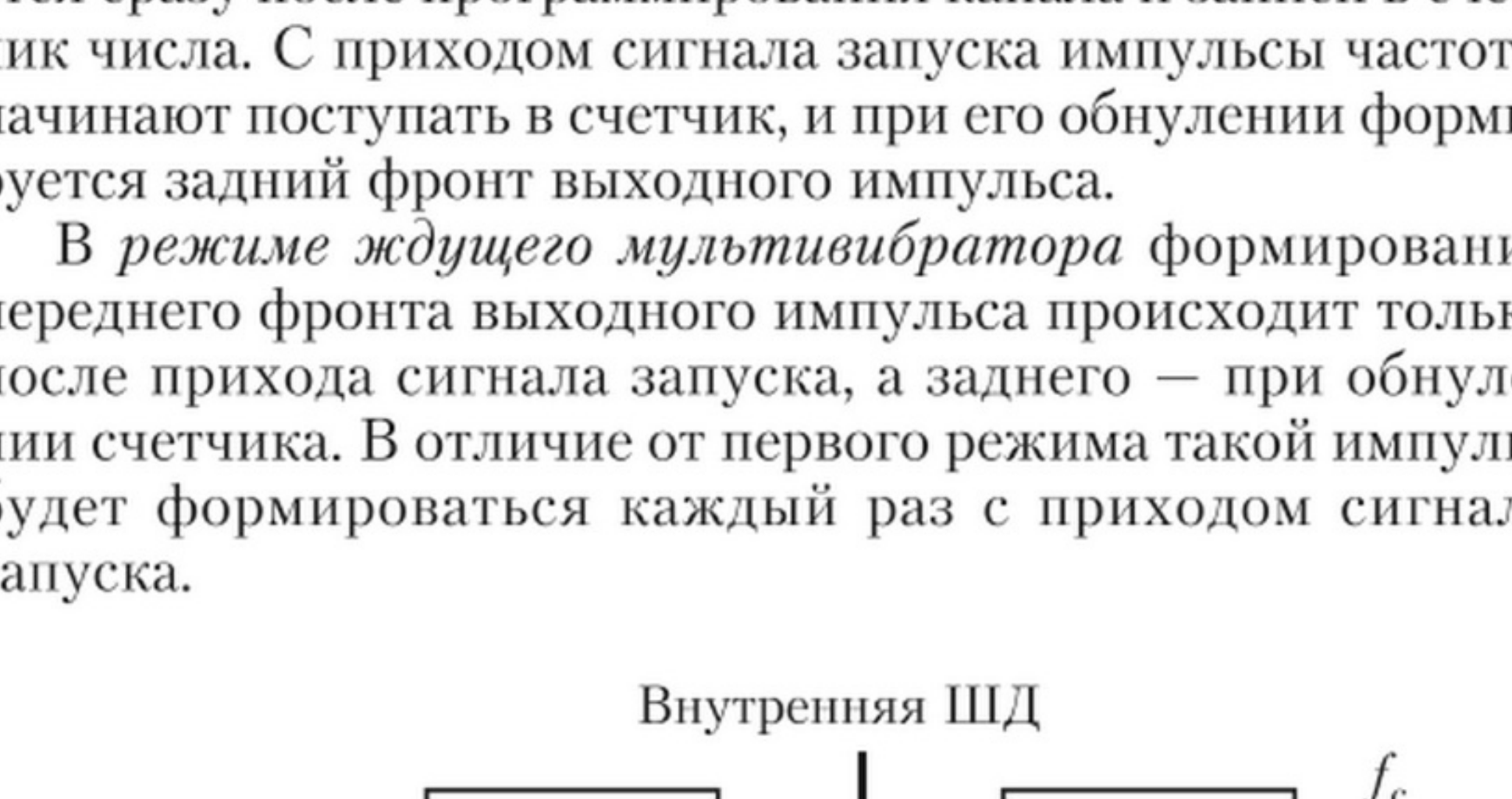


Рис. 4.26. **Программируемый интервальный таймер**

Содержание

В *режиме импульсного генератора* формируется последовательность коротких импульсов, длительность которых равна периоду входной частоты синхронизации, а частота следования определяется делением входной частоты на введенное в счетчик число, т.е.

$$f_{\text{вых}} = f_{\text{с}} / n.$$

Режим генератора меандра аналогичен предыдущему режиму, но отличается тем, что длительности положительного и отрицательного полупериодов выходного сигнала равны по величине.

В *режиме одиночного программно-запускаемого строба* осуществляется формирование короткого импульса с заданной временной задержкой. Содержимое счетчика начинает уменьшаться с приходом сигнала запуска. В момент его обнуления формируется выходной импульс, длительность которого определяется периодом частоты $f_{\text{с}}$. Для повторной выдачи импульса требуется программная загрузка счетчика.

Режим одиночного аппаратно-запускаемого строба отличается от предыдущего тем, что для повторной выдачи импульса программной загрузки счетчика не требуется. Для этого достаточно подать сигнал запуска.

4.11. Средства программируемой матричной логики

4.11.1. Программируемые логические матрицы

Успехи в области интегральной технологии обеспечили возможность разместить в одной БИС многие тысячи элементов, повысить надежность работы электронных устройств и быстродействие, снизить потребляемую мощность и габариты. Однако при этом возник вопрос: как на схемах с таким количеством элементов изготавливать устройства, решающие различные задачи?

Казалось бы, естественно изготавливать специализированные БИС, предназначенные для решения конкретной задачи. Но проектирование таких БИС — дорогой и трудоемкий процесс, поэтому их изготовление экономически оправдано лишь при массовом производстве.

Выходом из создавшегося положения явились *программируемые схемы с матричной структурой*. Такие схемы

представляют собой набор элементов и узлов, не связанных между собой. Для создания конкретного устройства элементы соединяются на заключительном этапе изготовления — в соответствии с заказом, или программируются самим пользователем. Фрагменты схемы соединяются либо созданием, либо устранением перемычек, подобно тому, как это делается при программировании ЗУ.

Первыми из таких устройств были созданные в середине 1970-х гг. *программируемые логические матрицы* (ПЛМ), обеспечивающие реализацию различных логических функций. *Матрица* — это сетка взаимно перпендикулярных проводников (рис. 4.27), в местах пересечения которых находятся полупроводниковые элементы — диоды или транзисторы, включенные через легкоплавкие перемычки к проводникам матрицы. При программировании ненужные перемычки пережигаются импульсами тока.

Рассмотрим, какие функции реализует матрица M_1. Если на горизонтальную шину подана лог. 1 (высокий потенциал), то диод будет заперт, и ток через R не течет. Падение напряжения на R отсутствует, и на вертикальной шине,

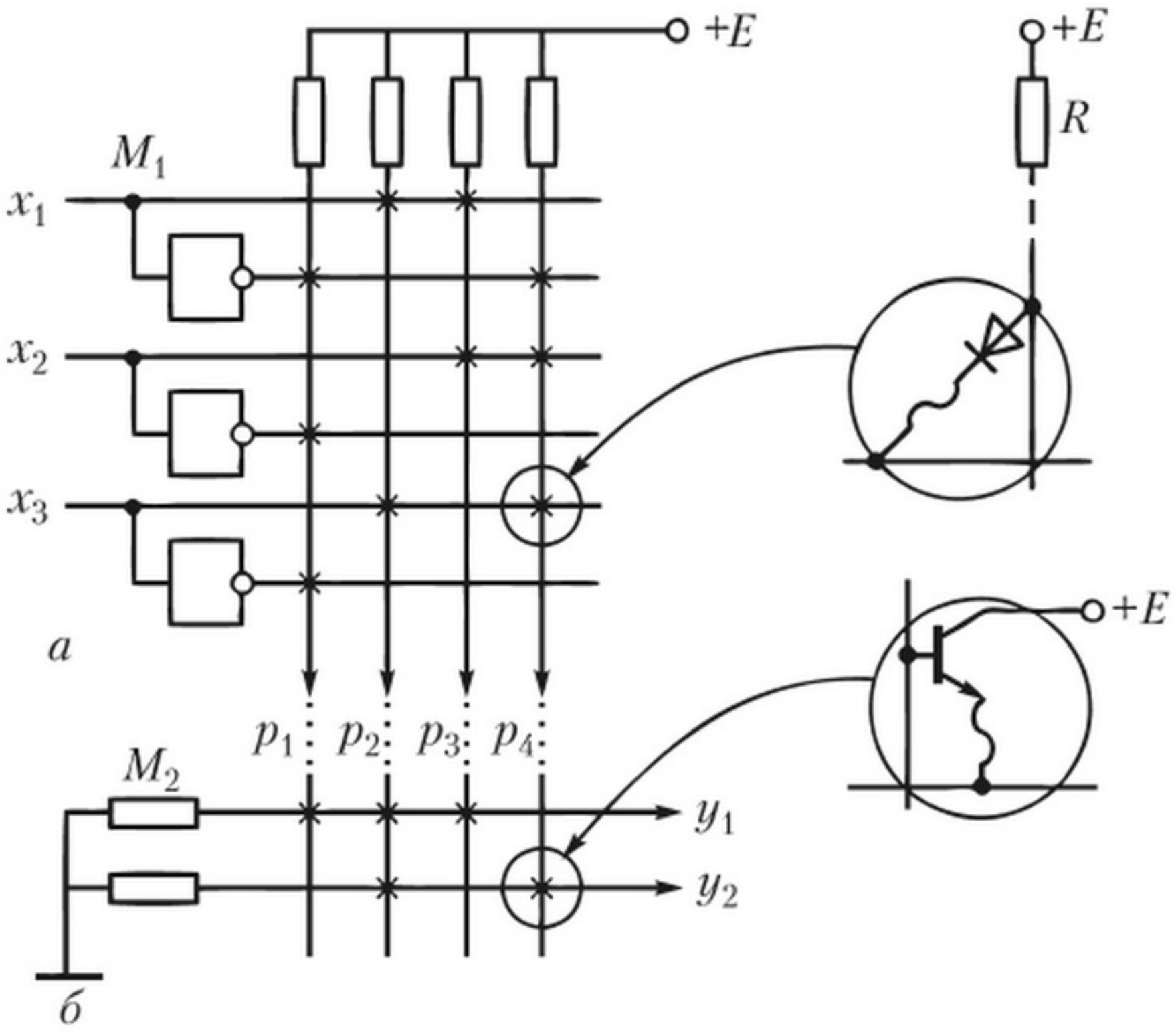


Рис. 4.27. **Программируемые логические матрицы:**
а — матрица И; *б* — матрица ИЛИ

соединенной с анодом запертого диода, остается высокий потенциал, соответствующий лог. 1. Если же на горизонтальной шине низкий потенциал (лог. 0), то протекающий через R и открытый диод ток создаст падение напряжения на R и на вертикальной шине будет лог. 0. Для сохранения на вертикальной шине матрицы M_1 высокого потенциала, необходимо, чтобы все диоды этой шины были заперты, т.е. на всех входах была лог. 1. Таким образом, на вертикальных шинах реализуется операция «И» от аргументов, помеченных крестиками, и матрицу M_1 называют матрицей И. В приведенном примере на выходах P_i реализуются функции:

$$P_1 = \bar{x}_1\bar{x}_2\bar{x}_3;\ P_2 = x_1x_3;\ P_3 \quad x_1x_2;\ P_4 = \bar{x}_1x_2x_3.$$

Конъюнкции, образующиеся на выходах матрицы И, называют *термами*.

Выясним, какие функции реализует матрица M_2 (см. рис. 4.27, *б*). Если на вертикальной шине высокий потенциал (лог. 1), то транзистор открыт и высокий потенциал $+E$ через него поступает на горизонтальную шину y, т.е. на ней будет лог 1. Если транзистор закрыт (на базе потенциал лог. 0), то на шине y будет потенциал корпуса — лог. 0. Очевидно, что открытие любого из транзисторов приводит к появлению на горизонтальной шине y потенциала лог. 1, поэтому на ней реализуется функция «ИЛИ» от входных переменных P_i, помеченных крестиками. Так как каждая из переменных P_i реализует операцию «И» над аргументами x, то в итоге на выходах y реализуются функции:

$$y_1 = P_1 \vee P_2 \vee P_3 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3 \vee x_1x_2 \quad \text{и}$$

$$y_2 = P_2 \vee P_4 = x_1x_3 \vee \bar{x}_1x_2x_3.$$

Таким образом, ПЛМ осуществляет реализацию дизъюнктивной нормальной формы логических функций и используются для аппаратной реализации логических функций. Основными параметрами ПЛМ являются число входов аргументов x, число выходов y и число термов (переменных P), которые могут входить в выходную функцию y.

Обычно наряду с матрицами И и ИЛИ в состав микросхемы ПЛМ входят блоки входных и выходных буферных каскадов. Входные буферы, кроме прямого и инверс-

ного представления значений аргументов x, формируют необходимую мощность сигналов для питания элементов матрицы И. Выходные буферы обеспечивают необходимую нагрузочную способность выходов, разрешают или запрещают выход ПЛМ на внешние шины в зависимости от состояния входного сигнала разрешения, аналогичного сигналу *CE*, используемому в ЗУ.

ПЛМ могут выполняться как на биполярных, так и на МОП-транзисторах. В последнем случае матрица M_1 выполняет функцию «И-НЕ», а матрица M_2 — функцию «ИЛИ-НЕ». Программирование ПЛМ заключается в пережигании перемычек, соединяющих горизонтальные и вертикальные шины матриц, которое выполняется на специальных устройствах — программаторах. В исходном состоянии все перемычки целы. Если в переменную P входит прямое значение аргумента x, то пережигается перемычка, соединяющая вертикальную шину P с инверсной горизонтальной шиной, если значение аргумента инверсное — то с шиной, передающей прямое значение аргумента. Если аргумент не участвует в формировании переменной, пережигаются обе перемычки. Если переменная P входит в функцию y, то перемычка сохраняется, в противном случае — пережигается.

4.11.2. Программируемая матричная логика

Одно из важных применений БИС программируемой логики — замена ИС малого и среднего уровня интеграции при реализации произвольных логических функций. В таких случаях возможность использования выходов любых конъюнкторов (вертикальные шины P) любыми дизъюнкторами (горизонтальные шины y) становится излишним усложнением. Отказ от этой возможности приводит к структуре *программируемой матричной логики* (ПМЛ).

В ПМЛ (рис. 4.28) выходы элементов «И» (выходы первой матрицы) жестко расположены между элементами «ИЛИ» (входами матрицы ИЛИ). В показанной структуре для m входов и n выходов требуется m инверторов, n схем «ИЛИ» и $4n$ элементов «И», поскольку каждому элементу «ИЛИ» придается по четыре конъюнктора. По сравнению с ПЛМ эта схема имеет меньшую функциональную гибкость, так как матрица ИЛИ фиксирована, но их изготовление и использование проще. Преимущества ПМЛ

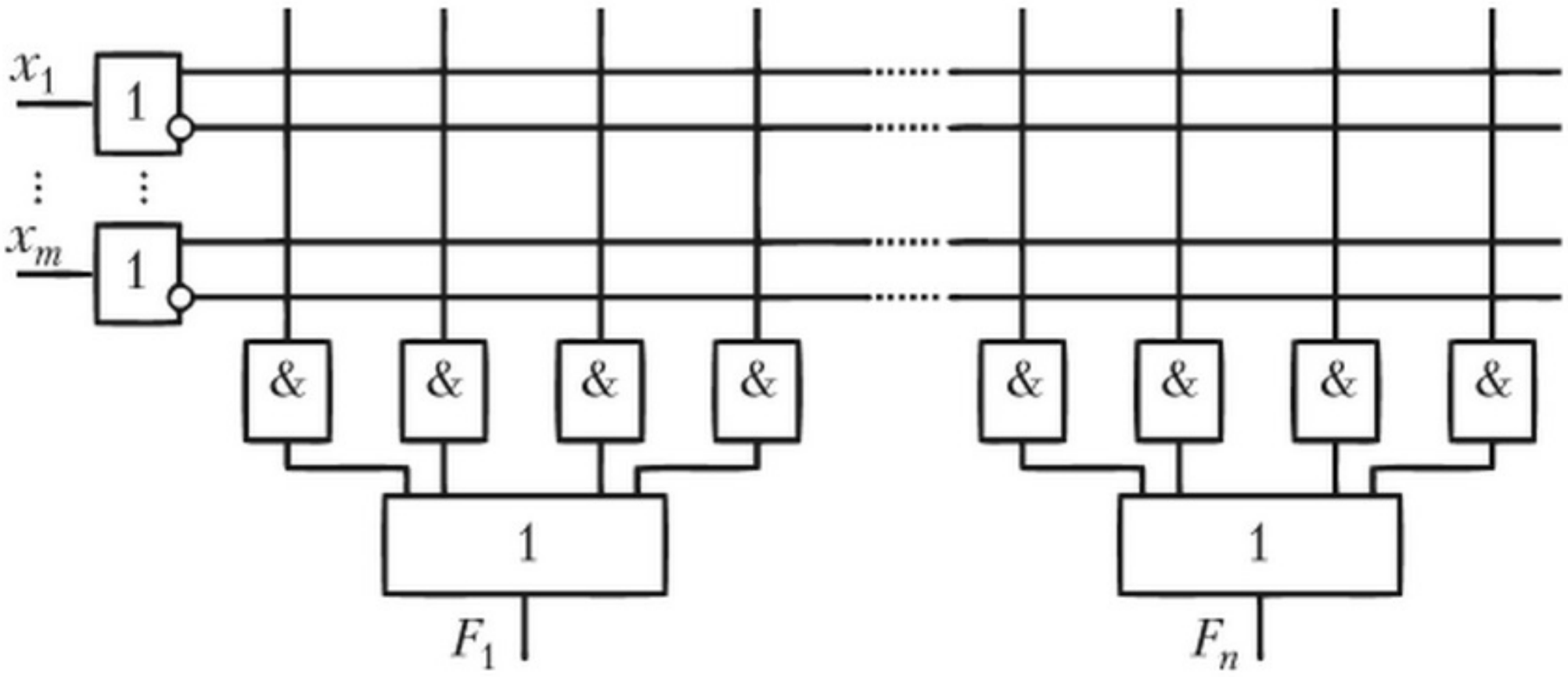


Рис. 4.28. **Базовая структура ПМЛ**

особенно проявляются при проектировании несложных устройств.

Подготовка задач к решению на ПМЛ имеет много общего с подходом к решению задач на ПЛМ, но есть и различия. Для ПМЛ логическую функцию нужно представить таким образом, чтобы число термов было не больше числа входов выходных элементов «ИЛИ».

4.11.3. Модернизация схем ПЛМ и ПМЛ

Схемы с программируемым выходным буфером. Рассмотренные схемы ПЛМ и ПМЛ являются базовыми. Для расширения их функциональных возможностей применяют некоторые схемные усовершенствования. К одним из них относят использование *программируемых выходных буферов*. Такие буферы обеспечивают возможность получения выходных функций не только в прямом, но и в инверсном виде.

В такой схеме (рис. 4.29) разработанные матрицами функции $F_1^* ... F_n^*$ проходят через выходной буфер, имеющий два входа. На первый вход поступают функции F_i^*, а на второй — нулевые сигналы от потенциала корпуса через плавкие перемычки (ПП). Буфер реализует функцию «сумма по модулю 2», поэтому, если перемычка цела, то сигнал с верхнего входа передается без изменения, т.е. $F_i^* = F_i$. Если перемычку пережечь, то на нижнем входе будет сигнал лог. 1 от источника питания через резистор R. Складываясь по модулю 2 с единицей, функции F_i^* инвертируются. Следовательно, в линиях с целыми перемычками функции про-

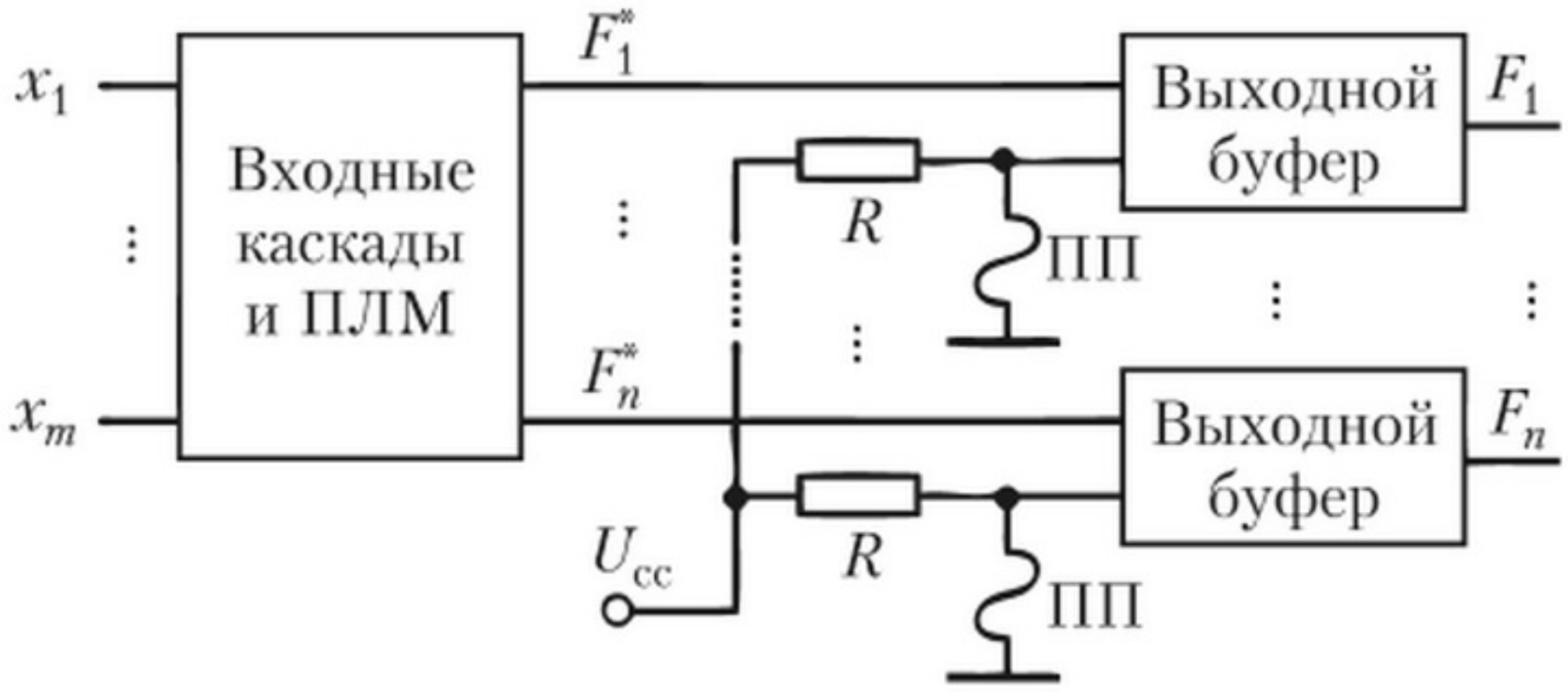
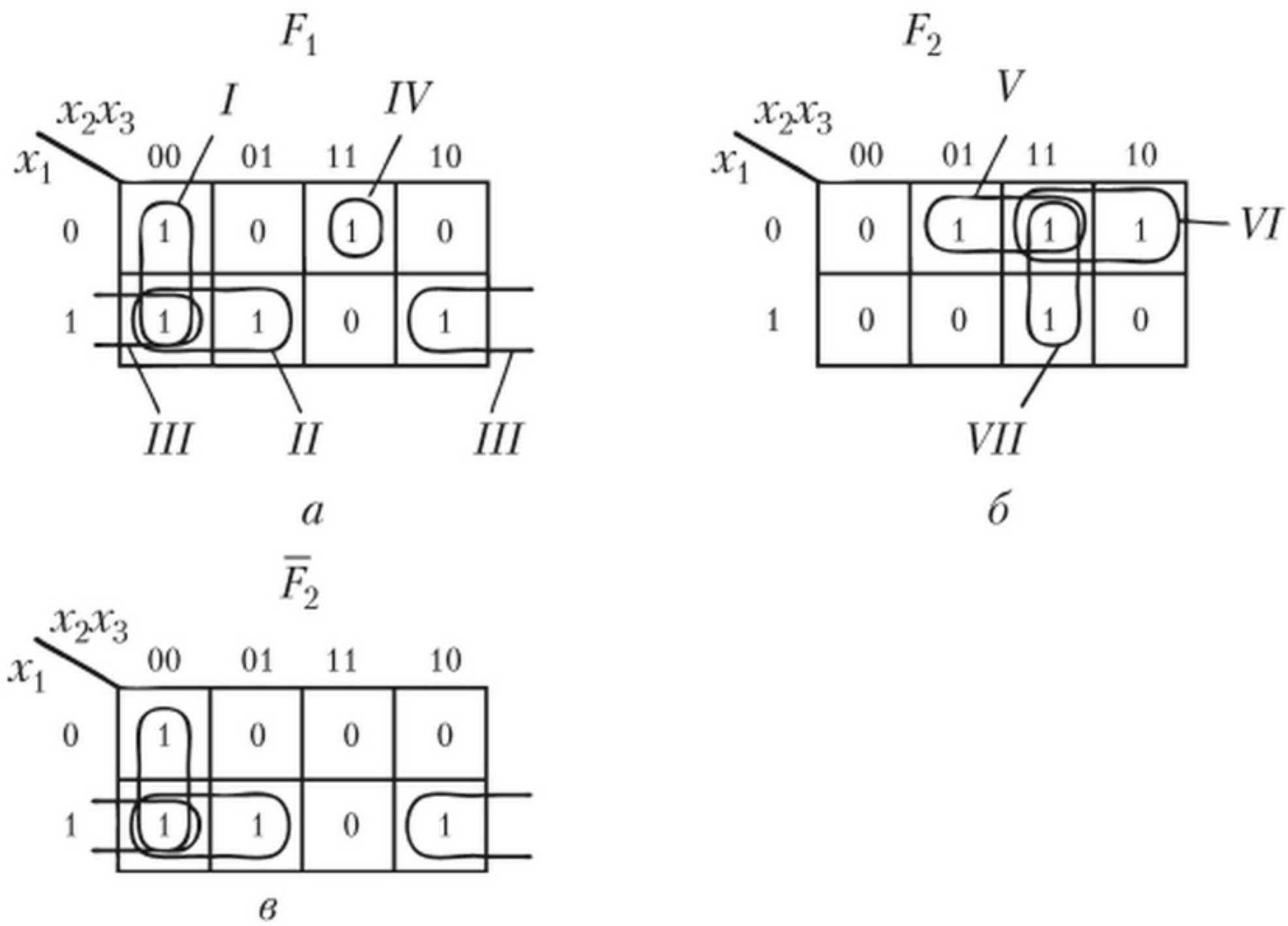


Рис. 4.29. **Схема ПЛМ с программируемыми выходными буферами**

ходят через буфер неизменными, а в линиях с отсутствующими перемычками — инвертируются. Программируемый буфер дает дополнительные возможности для минимизации числа переменных в реализуемой системе логических функций.

Пример 4.1. Пусть с помощью ПЛМ нужно воспроизвести систему из двух функций, представленных картами Карно (рис. 4.30).



а *б* *в*

Рис. 4.30. **Карты Карно для примера воспроизведения функции в ПЛМ с программируемым выходным буфером**

Содержание

Чтобы охватить все «1» функции F_1 контурами понадобится четыре контура, отмеченные на рис. 4.30, *а* римскими цифрами *I*, *II*, *III* и *IV*. Им соответствуют следующие термы:

$$p_1 = \overline{x}_2\overline{x}_3;\ p_2 \quad x_1\overline{x}_2;\ p_3 \quad x_1\overline{x}_3;\ p_4 \quad \overline{x}_1x_2x_3 .$$

Для охвата всех «1» функции F_2 потребуются контуры *V*, *VI* и *VII* (рис. 4.30, *б*), которым соответствуют другие три терма:

$$p_5 = \overline{x}_1x_3;\ p_6 \quad \overline{x}_1x_2;\ p_7 \quad x_2x_3 .$$

Таким образом, для реализации двух функций с помощью ПЛМ понадобится семь различных термов. При инвертировании функции единицы занимают в карте Карно те позиции, которые были нулями, и наоборот. Как это видно из сравнения карт Карно функций F_2 и $\overline{F}_2$ (рис. 4.30, *б* и *в*), количество единиц и охватывающих их контуров при этом не изменилось. Но контуры на карте функции теперь совпадают с контурами для функции F_1 (см. рис. 4.30, *а*, *в*). Таким образом, термы, полученные для функции F_1, могут быть использованы для реализации функции $\overline{F}_2$ и общее число термов, необходимых для реализации двух функций, сокращается с семи до четырех. Возврат от функции $\overline{F}_2$ к функции F_2 осуществляется пережиганием перемычки в линии выхода.

Схемы с памятью. Для построения цифрового автомата необходима не только комбинационная часть, обеспечивающая вычисление логических функций, но и элементы памяти для хранения состояния автомата. Развитием схем ПЛМ

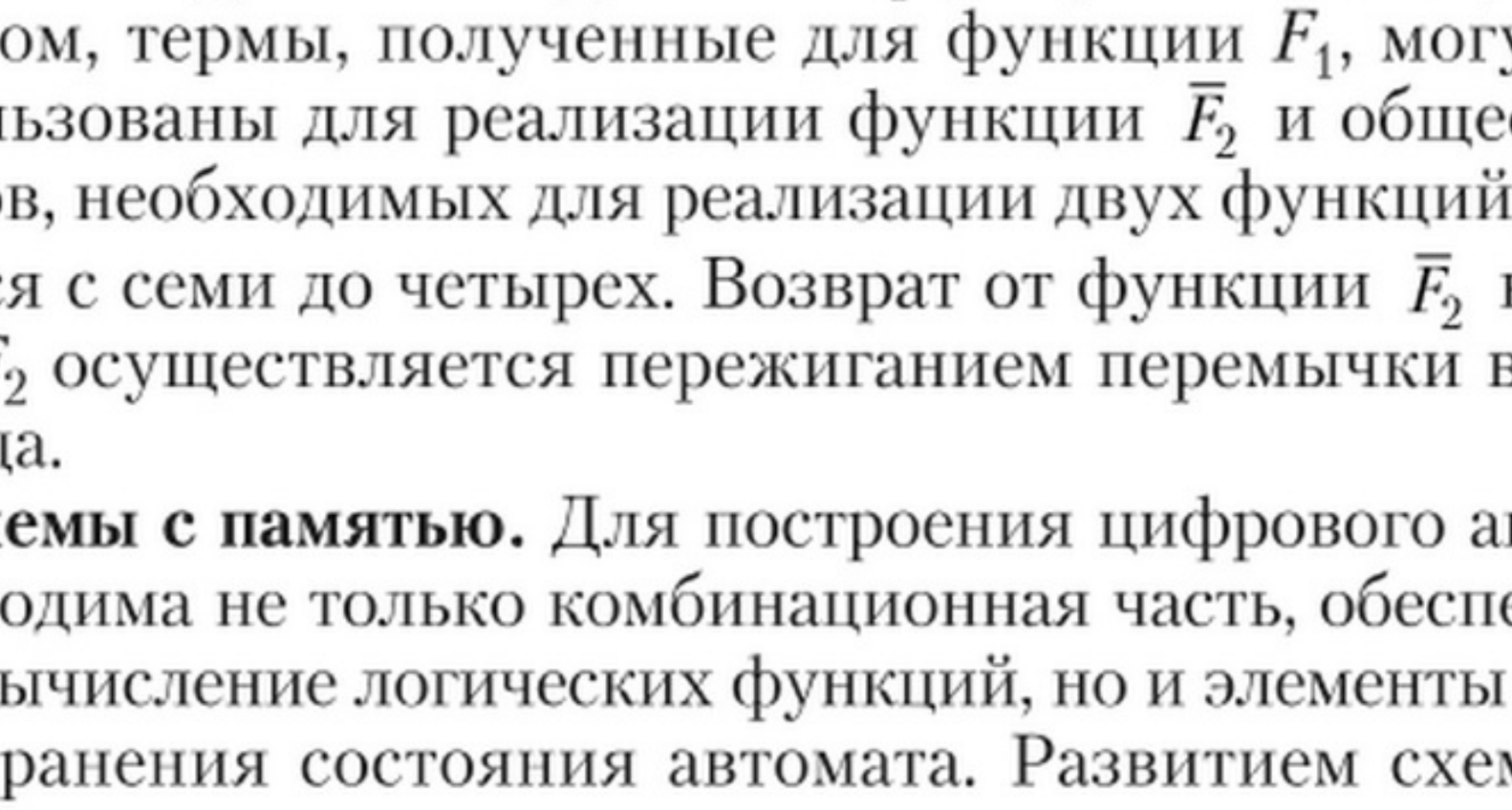


Рис. 4.31. **Структура ПЛМ с памятью**

стали схемы, в которых кроме средств матричной логики содержатся элементы памяти — триггеры или целые регистры. Кроме трех обычных параметров — числа входов, выходов и термов — эти схемы характеризуются числом триггеров или разрядностью регистра. Схема ПЛМ с памятью соответствует структуре цифрового автомата с памятью (рис. 4.31). Регистр обеспечивает хранение состояний автомата Q. Общее число внутренних состояний автомата определяется числом регистров. Данный цифровой автомат — синхронный, так как сигналы с выходов регистров на входы ПЛМ подаются только с приходом тактовых сигналов (ТС).

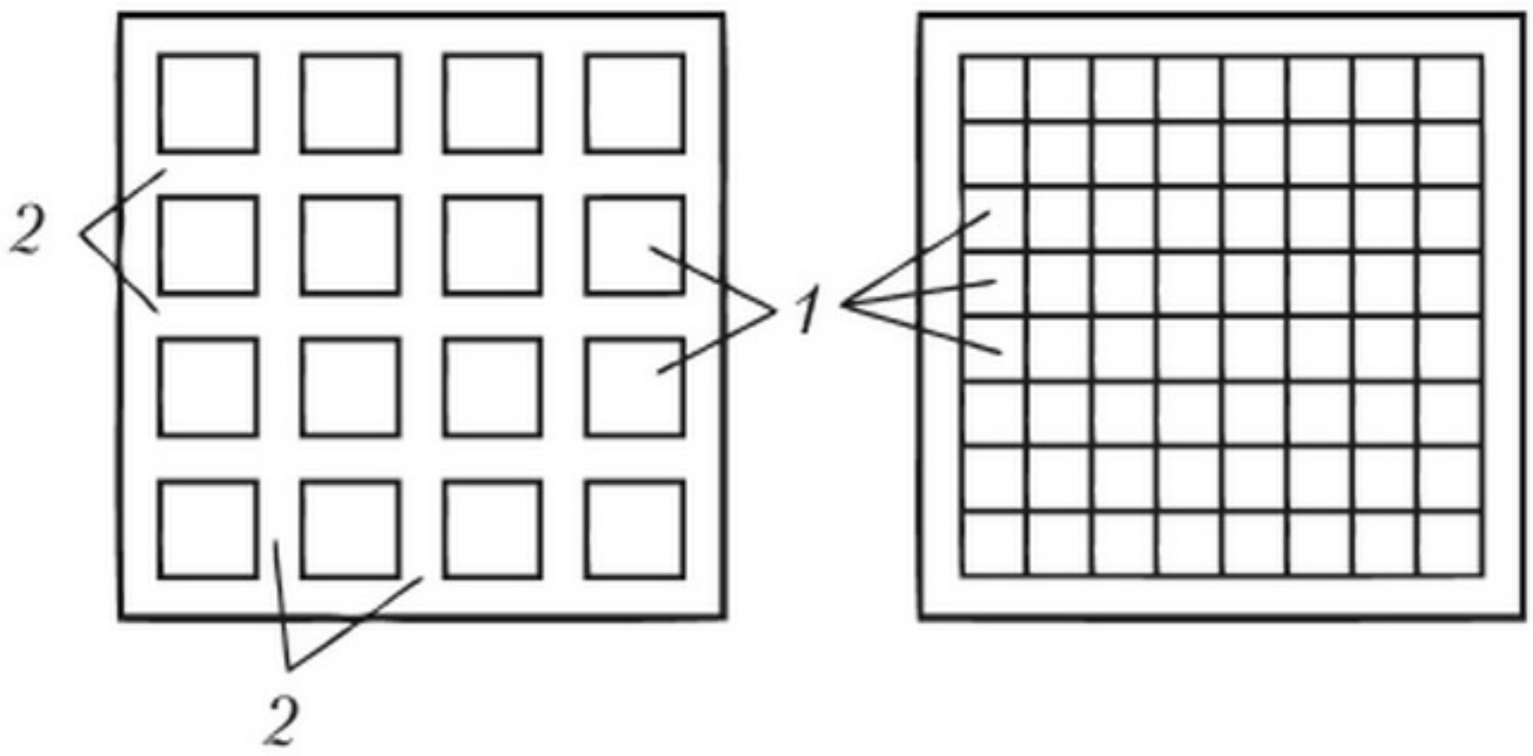
4.11.4. Базовые матричные кристаллы

Проектирование и изготовление БИС/СБИС — весьма дорогостоящий процесс, поэтому их производство экономически оправдано лишь при большой серийности. В то же время возможности интегральной схемотехники позволяют в виде одной микросхемы выполнить целое устройство. Базовые матричные кристаллы (БМК) явились тем средством, на основе которого при сравнительно низких затратах стало возможным выполнение специализированных устройств в виде БИС/СБИС. БМК представляют собой совокупность регулярно расположенных на полупроводниковом кристалле ячеек, каждая из которых содержит типовой набор элементов — транзисторов, резисторов. Такой кристалл является полуфабрикатом, который производится в массовых количествах без ориентации на конкретного потребителя. Чтобы на его основе было получено конкретное устройство, необходимо выполнить требуемые соединения элементов и ячеек. Выполнение соединений осуществляется на заключительных этапах изготовления, а проектирование устройства, таким образом, сводится к созданию рисунка межсоединений. Для упрощения процесса проектирования имеются библиотеки готовых решений. Каждое готовое решение предлагает определенный вариант межсоединений элементов внутри ячейки, позволяющий на базе одной или нескольких ячеек получить готовый функциональный узел — логический элемент, триггер и т.п. Проектирование устройства сводится к выбору готовых узлов и трассировке соединений между ними. Такие БИС/СБИС называются *полузаказными*. Естественно, что процесс проектирования и изготовления такой микросхемы

значительно дешевле, чем разработка и изготовление оригинальной БИС.

Первоначально структура БМК представляла собой матрицу ячеек, называемых *базовыми*, которые располагались в центральной области кристалла и были изолированы друг от друга (рис. 4.32, *а*). Области кристалла, не занятые ячейками, служили для выполнения соединений между ячейками — *трассировки каналов*. Такая структура называется *канальной*. У канальных БМК большие возможности по созданию связей, но низкая плотность упаковки из-за значительных затрат площади кристалла на области межсоединений. Эта структура характерна для БМК, выполненных по биполярной технологии, так как биполярные элементы, в отличие от КМОП-элементов, обладают сравнительно высокой мощностью рассеивания.

В отличие от БМК с канальной структурой у БМК с *бесканальной* структурой (рис. 4.32, *б*) вся внутренняя область кристалла заполнена ячейками. В этом кристалле любая область может быть использована как для создания логической схемы, так и для создания межсоединений. Бесканальные БМК характерны для КМОП-схемотехники, в которой малая мощность рассеяния базовых ячеек позволяет добиваться высокой плотности упаковки. Бесканальные БМК реализуются в вариантах «*море вентилей*» и «*море транзисторов*». Первый содержит массив законченных логических элементов, второй — массив транзисторов.



а *б*

Рис. 4.32. **Структура канального (*а*) и бесканального (*б*) БМК:**
1 — базовые ячейки; *2* — каналы для трассировки

Содержание

Так как в бесканальных БМК положение трассировочных каналов и ячеек не является жестким, и при проектировании конкретной БИС площадь кристалла может распределяться между трассировочными каналами и функциональными ячейками, потери площади кристалла снижаются. Например, в БМК, содержащем массив транзисторов, в некоторых рядах реализуются логические элементы, а другие ряды используются под трассировочные каналы, в них транзисторы остаются нескоммутированными и над ними проходят трассы.

Внутренняя область кристалла бесканального БМК окружена периферийной областью, расположенной по краям прямоугольной пластины БМК. Ячейки в периферийной области отличаются от базовых ячеек внутренней области. Они предназначены для решения задач ввода/вывода сигналов, поэтому они содержат набор специальных элементов и контактные площадки, через которые осуществляется подключение кристалла к внешним контактам микросхемы.

Возможности БМК во многом определяются числом слоев межсоединений (в настоящее время это 2÷6). Многослойность облегчает трассировку и позволяет изготовлять БМК более высокого уровня интеграции. В простейшем случае двухслойной трассировки на нижнем уровне выполняют соединения внутри базовых ячеек и связи по вертикальным каналам. Этот слой делают либо в виде диффузионной области самого кристалла, либо в виде металлических дорожек. Второй слой металлизированных соединений дает разводку горизонтальных трасс и обслуживающих линий.

Рост уровня интеграции ведет к возможностям реализации на одном кристалле все более сложных устройств и систем. Это породило *блочные структуры* БМК. Такие БМК содержат как блоки логической обработки данных, так и память или другие специализированные блоки. Каждый из таких блоков представляет собой как бы «мини»-БМК. Между этими блоками располагаются трассировочные каналы. На периферии блоков изготовляют внутренние буферные каскады для формирования достаточно мощных сигналов, обеспечивающих передачу сигналов по межблочным связям, имеющим относительно большую длину.

В зависимости от назначения БМК подразделяются на цифровые, аналоговые и цифроаналоговые. *Аналоговые* и *цифроаналоговые БМК*, появившиеся позднее цифровых, имеют состав базовых ячеек, позволяющий получать на их

основе такие схемы, как операционные усилители, аналоговые ключи и компараторы и т.д. Для их реализации используется биполярная технология. *Цифровые БМК* выполняют на базе КМОП-технологии.

4.12. Программируемые логические интегральные схемы

Средства программируемой матричной логики получили свое развитие в направлении, называемом «*программируемые логические интегральные схемы*» (ПЛИС). Продолжением ПЛМ и ПМЛ стали микросхемы CPLD (*Complex Programmable Logic Devices — сложные программируемые логические устройства*), а продолжением БМК — микросхемы FPGA (*Field Programmable Gate Arrays — программируемые пользователем вентильные матрицы*, где под вентилем понимается любой логический элемент). Рост уровня интеграции дал возможность размещать на кристалле схемы, сложность которых соответствует целым системам. Эти схемы именуются SOC (*System On Chip — система на кристалле*).

Сфера применения ПЛИС чрезвычайно широка, на них могут строиться не только крупные блоки схем, но и системы в целом, включая память и процессоры.

4.12.1. Средства обеспечения программируемости ПЛИС

Программируемость, т.е. возможность создания конкретного устройства на базе стандартной СБИС, обеспечивается наличием в ней множества ключей, которые в процессе программирования могут стать замкнутыми либо разомкнутыми. Состояния ключей задают ту или иную конфигурацию схемы, формируемой на кристалле. Число программируемых ключей в ПЛИС зависит от ее сложности и может доходить до нескольких миллионов. В ПЛИС используются следующие виды программируемых ключей:

- перемычки типа *antifuse*;
- ЛИЗМОП-транзисторы с двойным затвором;
- ключевые транзисторы, управляемые триггерами памяти конфигурации («теневым» ЗУ).

Пробиваемые при программировании диэлектрические перемычки используются в качестве ЗЭ в однократно программируемых ЗУ (см. параграф 4.2). Перемычки типа *antifuse*, используемые в ПЛИС, также обеспечивают одно-

кратное программирование СБИС. Перемычка образуется трехслойным диэлектриком с чередованием слоев «оксид — нитрид—оксид» и в исходном состоянии имеет очень высокое сопротивление (протекающий через нее ток составляет $\approx 10^{-15}$ А).

Программирующий импульс напряжения пробивает перемычку и создает проводящий канал. Величина тока, создаваемого импульсом программирования, влияет на диаметр проводящего канала, что позволяет управлять параметрами проводящей перемычки (ток 5 мА создает перемычку с сопротивлением 600 Ом, а 15 мА — 100 Ом). ПЛИС с программированием на основе перемычек *antifuse* выпускаются фирмами *Actel* и *Lattice Semiconductor*.

ЛИЗМОП-транзисторы с плавающим затвором, используемые в качестве запоминающих элементов в схемах репрограммируемой памяти, также используются в ПЛИС. Из элементов со стиранием информации ультрафиолетовым излучением выделился вариант вообще без возможности стирания данных. В этих схемах в корпусе ИС прозрачное окошко, через которое происходит облучение кристалла, просто отсутствует, т.е. возможность стирания исключается.

Репрограммируемые ПЛИС на основе схемотехники EPROM требуют длительного (около часа) стирания старой конфигурации под воздействием ультрафиолетового излучения. Из-за деградации свойств материалов под действием излучения число циклов программирования ограничено. При этом необходимо извлекать СБИС из устройства. В ПЛИС на основе схемотехники EEPROM стирание осуществляется электрическими сигналами, что не требует извлечения микросхемы из устройства. Время стирания старой информации и записи новой составляет несколько миллисекунд, а число таких циклов — $10^4 \div 10^6$. В последнее время в репрограммируемых ПЛИС все шире используется схемотехника на основе энергонезависимых ячеек флэш-памяти.

Еще одним схемотехническим решением, используемым в ПЛИС для программирования конфигурации СБИС, стали *транзисторные ключи, управляемые триггером памяти конфигурации* (рис. 4.33).

Ключевой транзистор *VT*2 замыкает или размыкает участок *AB* в зависимости от состояния триггера, выход которого подключен к затвору транзистора *VT*2. При про-

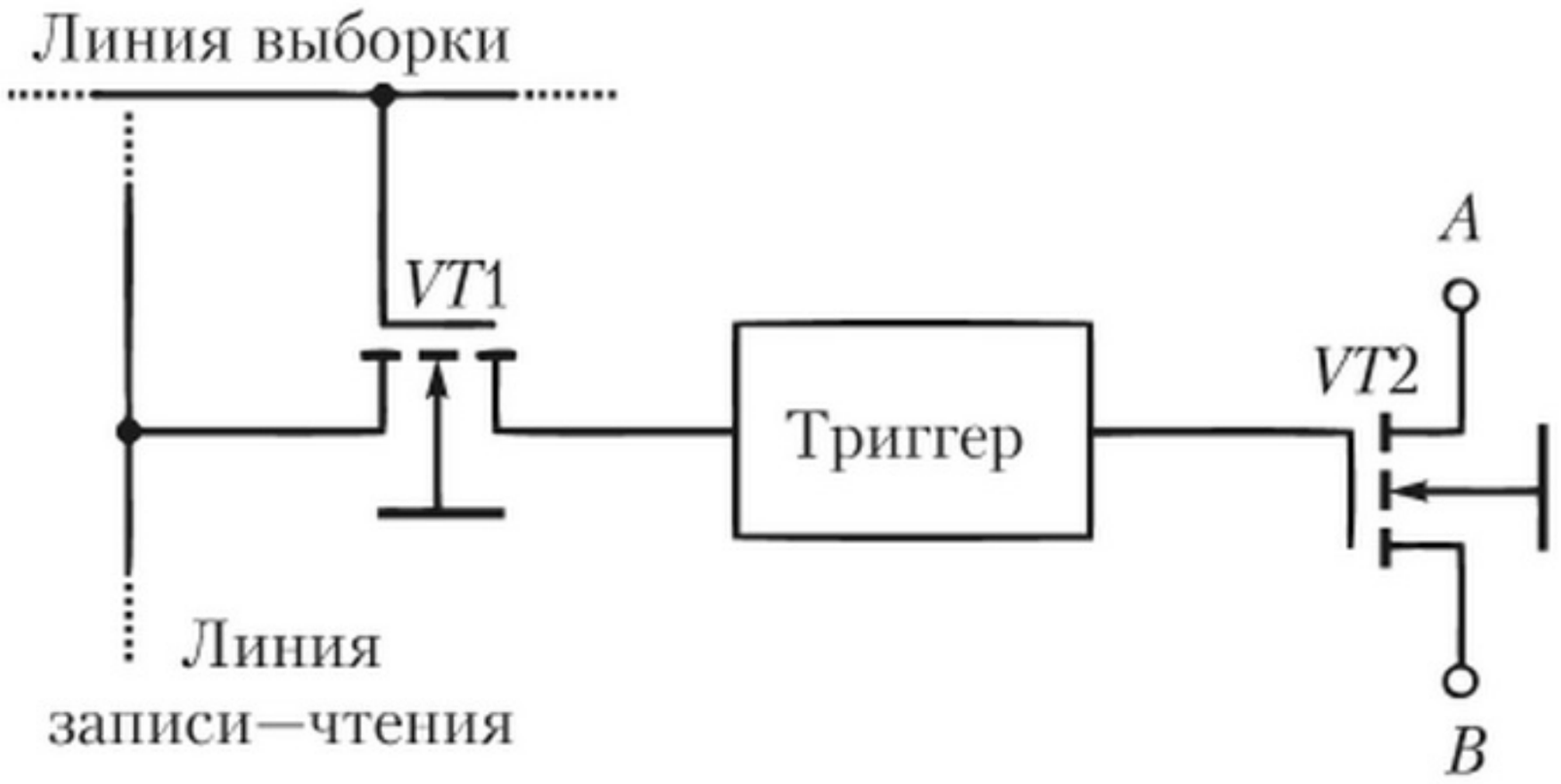


Рис. 4.33. **Ключевой транзистор, управляемый триггером памяти конфигурации**

граммировании на линию выборки подается высокий потенциал, и транзистор *VT*1 включается. С линии записи—чтения подается сигнал, устанавливающий триггер в состояние лог. 1 или лог. 0. В рабочем режиме транзистор заперт, и триггер сохраняет установленное состояние.

Память конфигурации представляет собой статическое ОЗУ, поэтому при выключенном питании все триггеры памяти конфигурации сбрасываются, в связи с чем каждый раз при включении требуется программирование ПЛИС. Этот процесс называется инициализацией и на него уходит время, составляющее десятки и даже сотни миллисекунд. ПЛИС с программированием на основе статического ОЗУ производят такие фирмы, как *Xilinx* и *Altera*.

Говоря об общих свойствах ПЛИС, следует отметить, что благодаря регулярной структуре они реализуются с уровнем интеграции, близким к максимальному. Так как для средств программирования межсоединений требуются затраты дополнительной площади кристалла, ПЛИС по уровню интеграции несколько уступают БМК. Однако в отличие от БМК ПЛИС не требуют каких-то завершающих технологических операций, выполняемых специально под определенного заказчика. Потребитель ПЛИС осуществляет формирование требуемой конфигурации устройства, самостоятельно программируя ее. В результате эти СБИС могут выпускаться большими тиражами, что сразу снижает их цену. Стоимость устройства на основе ПЛИС, таким образом, становится значительно ниже, чем на основе БМК. В связи с этим в большинстве применений ПЛИС практически вытеснили БМК.

4.12.2. Сложные программируемые логические устройства (CPLD)

Архитектурно микросхема класса CPLD состоит из центральной коммутационной матрицы, множества функциональных блоков, в которых осуществляется вся логическая обработка сигналов, и блоков ввода/вывода, расположенных на периферии кристалла и осуществляющих прием и передачу сигналов между кристаллом микросхемы и контактными площадками. Одной из ведущих фирм по производству микросхем класса CPLD является фирма *Altera*, а семейство микросхем MAX этой фирмы является классическим примером данного направления ПЛИС. Первая микросхема этого семейства была анонсирована в 1995 г. и была реализована на основе технологии 0,5 мкм. В настоящее время фирма *Altera* продолжает выпускать микросхемы семейства MAX, которое насчитывает уже пять подсемейств. Это MAX 7000, появившееся в 1995 г., MAX 3000 (2002 г.), MAX II (2004 г.), MAX II Z (2007 г.) и MAX V (2010 г.). Последнее подсемейство было реализовано по технологии 0,18 мкм. Рассмотрим архитектуру микросхем CPLD на примере архитектуры микросхемы MAX 7000.

На рис. 4.34 представлен фрагмент структуры CPLD MAX 7000. Центральная коммутационная матрица здесь именуется *программируемой матрицей связи* (ПМС), функциональные блоки — *логическими блоками* (ЛБ). У младшего представителя CPLD было только два ЛБ. С повышением сложности микросхем число ЛБ увеличивалось, при

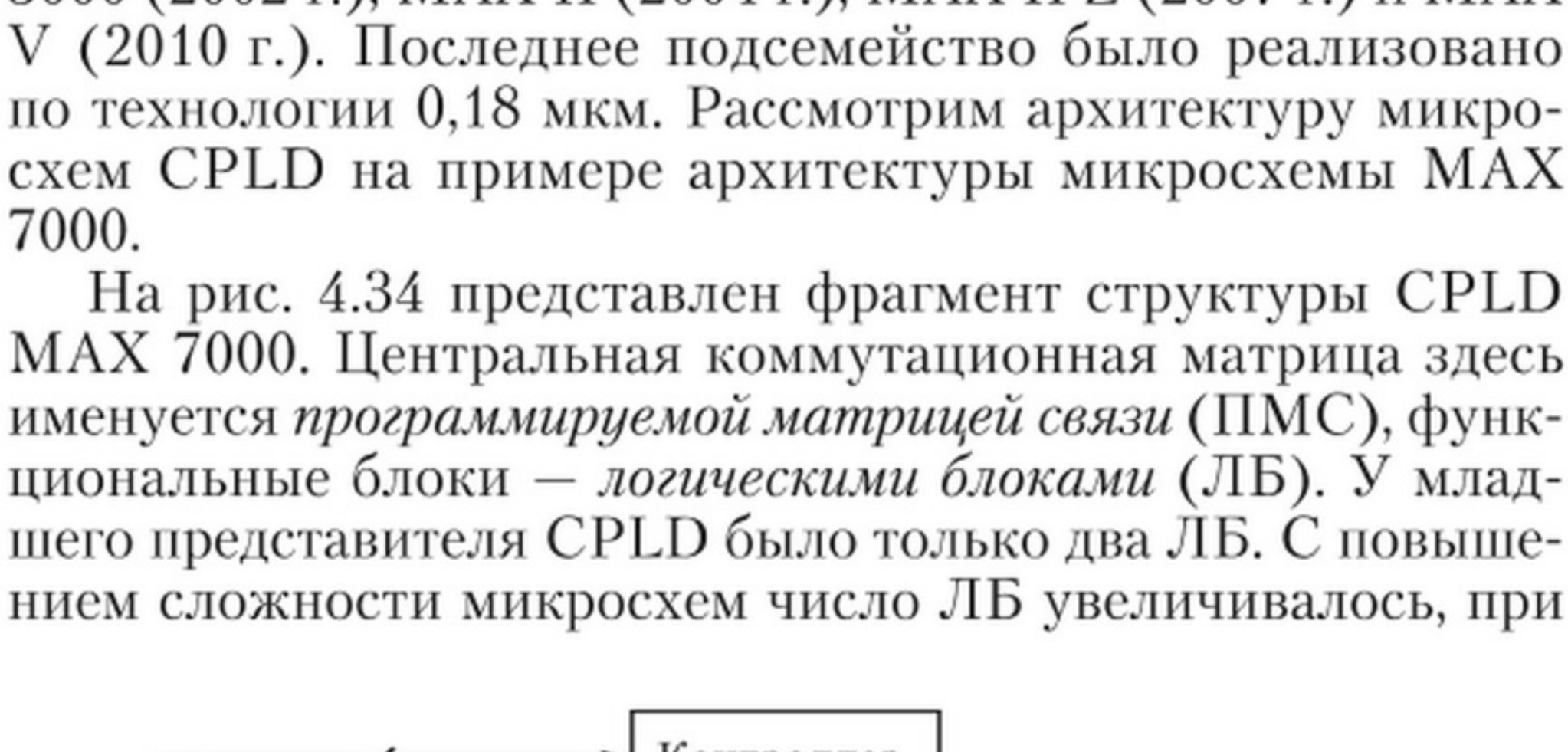


Рис. 4.34. **Фрагмент структуры CPLD MAX 7000**

этом структура микросхемы в целом составлялась повторением изображенного на фрагменте яруса, состоящего из двух ЛБ, двух блоков ввода/вывода (БВВ) и набора контактных площадок, расположенных слева и справа от ПМС. Старший представитель CPLD MAX 7000 содержит шестнадцать таких ярусов, т.е. 32 ЛБ, а последние представители MAX V — 1700 ЛБ.

Вся коммутация сигналов внутри кристалла выполняется ПМС. Эта матрица содержит набор пересекающихся вертикальных и горизонтальных линий связи.

Горизонтальные линии соединены с ЛБ, а также БВВ. Часть горизонтальных линий является входными, а часть — выходными. Сигнал с входных линий на выходные передается через вертикальные линии матрицы соединений, при этом коммутация сигнала с одной линии на другую осуществляется не через программируемый ключ (в отличие от FPGA), а через двухвходовую схему «И». Для этого к одному из ее входов подключается линия-источник, к выходу — линия-приемник, а ко второму входу — программируемый транзистор. Для того чтобы сигнал проходил с одной линии на другую, необходимо запрограммировать транзистор таким образом, чтобы через него на второй вход схемы «И» подавался разрешающий уровень лог. 1. Скорость распространения цифрового сигнала с входа конъюнктора на выход гораздо выше, чем через ключевой транзистор, поэтому использование вместо ключей управляемых конъюнкторов дает выигрыш в быстродействии микросхем CPLD. Вертикальные линии матрицы тянутся через все горизонтальные и могут быть подключены к любой входной линии; таким образом, любой вход матрицы может быть подключен к любому выходу, т.е. ПМС обеспечивает полную коммутируемость блоков CPLD. В микросхемах подсемейств MAX 7000 и MAX 3000 для программирования конфигурации использовались ячейки EEPROM, в MAX II, MAX II Z и MAX V — флэш-память. Поэтому, в отличие от ПЛИС FPGA фирмы *Altera*, ПЛИС CPLD хранят свою запрограммированную конфигурацию и после снятия напряжения питания.

Логические блоки осуществляют вычисление логических функций, но также имеют возможность хранения полученного результата в триггерах. В единую схему

устройства они объединяются с помощью ПМС. ЛБ состоит из программируемой матрицы И и шестнадцати *макроячеек* (*macrocells*) — Я1÷Я16. В программируемой матрице И, так же как и в ПЛМ, осуществляется формирование термов (см. рис. 4.27, *а*). Термы формируются сразу для всех макроячеек, но на вход каждой из них поступают свои термы, необходимые для реализации логической функции путем сборки термов по «ИЛИ». В макроячейке возможна прямая выдача вычисленного значения логической функции на выход, что используется для синтеза чисто комбинационных устройств. Для синтеза последовательностных устройств в макроячейке имеется триггер. В этом случае результат логической операции записывается в триггер и его выход подключается к выходу макроячейки. Синхронизация работы триггера, а также его асинхронная установка и сброс могут осуществляться как контроллером, так и с помощью термов, поступающих от программируемой матрицы. Кроме того, триггер макроячейки можно использовать для ввода внешних сигналов непосредственно с контактной площадки, минуя БВВ и ПМС. Такие цепи называют цепями быстрого ввода.

При синтезе последовательностных цифровых устройств (например, счетчиков) триггеры большей части макроячеек формируют промежуточный результат, который используется для дальнейшей обработки. В этом случае выход макроячейки поступает в ПМС. Если в макроячейке сформировался готовый выходной сигнал синтезируемого устройства, то ее выход подключается к БВВ. Все описанные функции, выполняемые макроячейкой, программируются.

В ПЛИС типа CPLD имеется набор глобальных управляющих сигналов, которые поступают одновременно на все ЛБ. Это сигналы тактирования, сброса и разрешения выхода. С их помощью обеспечивается синхронная работа всего синтезированного на основе CPLD устройства.

4.12.3. Программируемые пользователем вентильные матрицы (FPGA)

Структура программируемых пользователем вентильных матриц FPGA похожа на структуру канальных БМК (см. рис. 4.32, *а*). В их внутренней области, так же как у канального БМК, размещается множество регулярно расположенных ячеек — *логических блоков*, между кото-

рыми проходят трассировочные каналы, а на периферии кристалла расположены ячейки другого типа — *блоки ввода/вывода*. Современные микросхемы FPGA содержат и другие узлы, но знакомство с архитектурой этого класса СБИС начнем с их классической структуры.

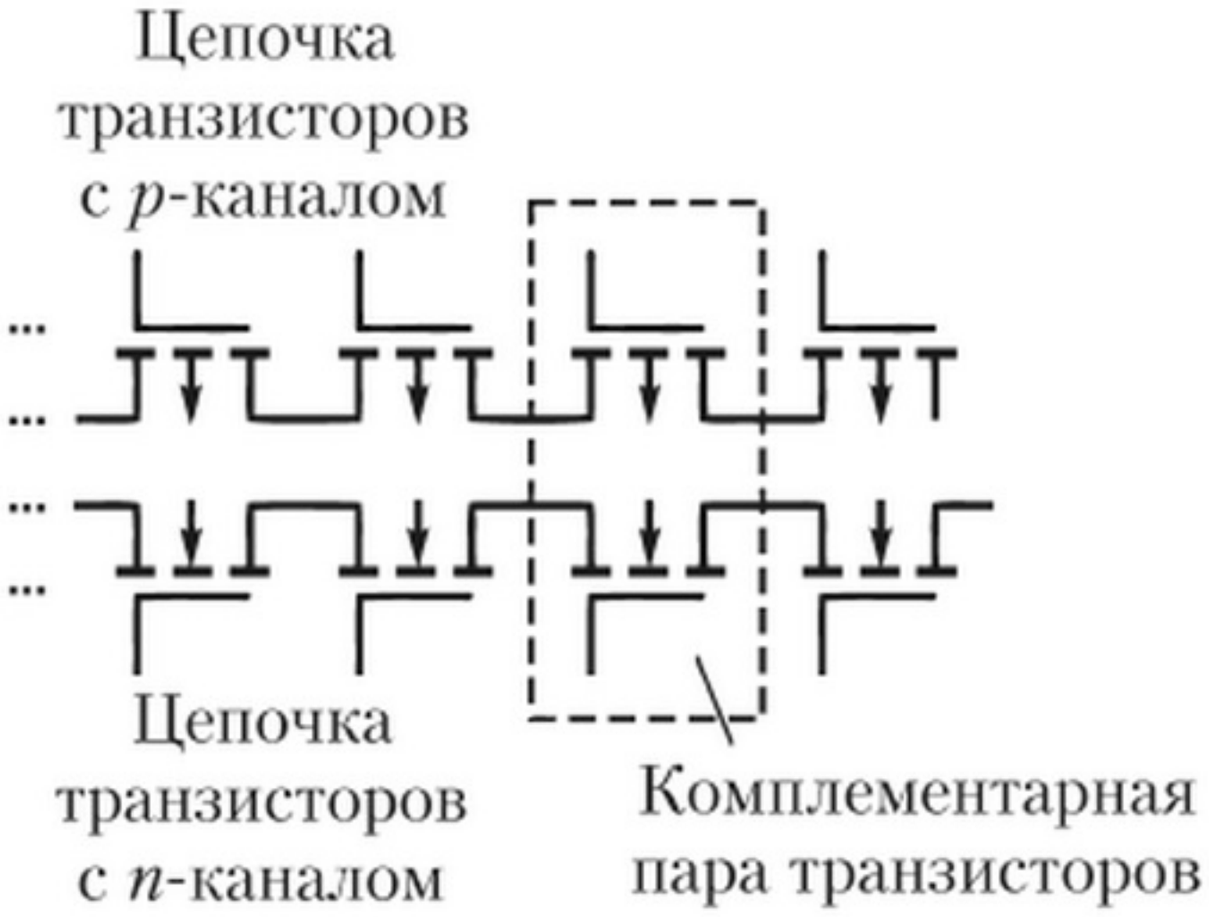
Логические блоки FPGA. В качестве ЛБ используются:

- транзисторные пары, простейшие логические схемы «И-НЕ», «ИЛИ-НЕ» и т.п. Такие ЛБ называются SLC (*Simple Logic Cells*, простые логические цепи);
- логические модули на основе мультиплексоров;
- табличные логические модули на основе программируемых ПЗУ, такие ЛБ называют LUTs (*Look-Up Tables*, таблицы поиска).

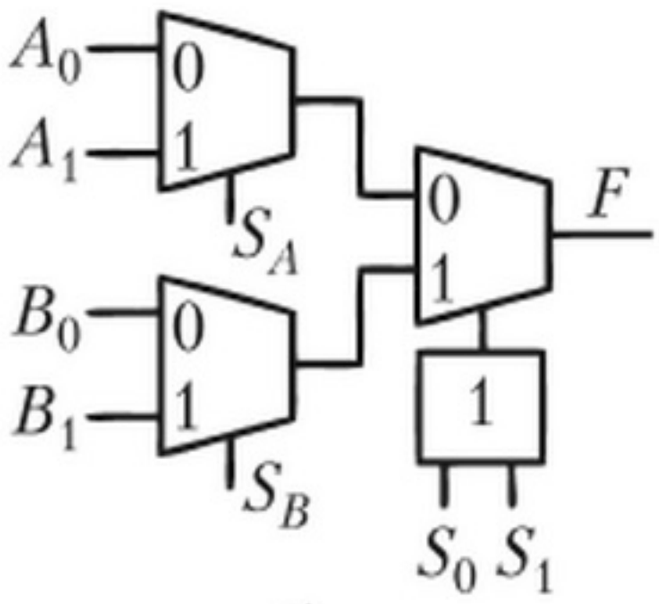
Двумя основными характеристиками ЛБ являются их «*зернистость*» (*Granularity*) и «*функциональность*» (*Functionality*). Эти характеристики противоречивы: чем ниже свойства ЛБ, характеризуемые одной из них, тем выше будут его свойства, характеризуемые другой. «Зернистость» связана с тем, насколько «мелкими» будут те части, из которых можно «собирать» нужные схемы, а «функциональность» — с тем, насколько велики логические возможности ЛБ.

Примером наиболее мелкозернистого может служить ЛБ, который содержит цепочки транзисторов с *p*- и *n*-каналами (рис. 4.35, *а*). Между цепочками транзисторов имеются трассировочные каналы, в которых могут быть реализованы необходимые межсоединения элементов. Каждая пара транзисторов (обведена на рис. 4.35, *а* пунктиром) представляет собой комплементарную пару, из которых можно строить простейшие логические схемы «И-НЕ», «ИЛИ-НЕ» в КМОП-логике. Принцип построения таких схем был подробно рассмотрен в параграфе 3.4. Чтобы «разорвать» цепочки транзисторов, на затворы транзисторов одной из пар подаются запирающие напряжения: напряжение питания — на верхний и корпус — на нижний. При этом транзисторы, находящиеся слева и справа от них, становятся изолированными друг от друга. Затем из полученных элементарных логических схем путем выполнения необходимых межсоединений осуществляется построение более сложных логических устройств.

Мелкозернистость ЛБ ведет к большой гибкости их использования. В то же время мелкозернистость усложняет



а



б



в

Рис. 4.35. **Примеры логических блоков:**

а — мелкозернистый; *б* — крупнозернистый; *в* — обозначение входов и выходов у мультиплексора «2—1»

систему межсоединений FPGA из-за большого числа программируемых точек связи.

Примером более крупного логического блока, используемого в FPGA, может служить ЛБ, состоящий из трех мультиплексоров «2—1» и элемента «ИЛИ» (рис. 4.35, *б*). Подключая к входам такого ЛБ переменные и константы, можно получить комбинационные функции двух переменных, многие функции четырех переменных и некоторые функции большего числа переменных, вплоть до восьми.

В FPGA с триггерной памятью конфигурации применяют крупнозернистые блоки. В таких блоках реализуются более сложные функции. При этом количество необходимых для реализации логического устройства межсоединений существенно сокращается, что упрощает их программирование. В то же время полностью использовать логические возможности такого ЛБ очень трудно, и часть его элементов при реализации конкретной логической схемы просто становится лишней.

Табличные логические модули представляют собой программируемые ПЗУ. Само ППЗУ уподобляется таблице истинности функции, в которой каждая строка — ячейка ППЗУ — соответствует какому-то набору аргументов, т.е. аргументы логической функции служат адресом при выборе ячейки ПЗУ. При программировании в таблицу истинности заносятся значения функции, которые она должна принимать на каждом наборе. Такие табличные преобразователи могут реализовывать любые логические функции n аргументов, где n — разрядность адреса ППЗУ.

Самые сложные ЛБ, используемые в FPGA, могут содержать в своем составе несколько табличных преобразователей на основе ППЗУ, набор мультиплексоров и триггеров. На основе таких ЛБ можно реализовывать достаточно сложные логические устройства с большим числом аргументов.

Блоки ввода/вывода FPGA. Эти блоки обычно содержат два канала — для ввода сигналов и для вывода (рис. 4.36), которые подключены к одной и той же контактной площадке микросхемы.

В каждом канале сигналы могут передаваться прямым путем между контактной площадкой и входом/выходом ПЛИС. Для этого используются верхние входы мультиплексоров. Кроме того, сигналы могут фиксироваться в триггерах, и в этом случае они поступают через нижние входы мультиплексоров.

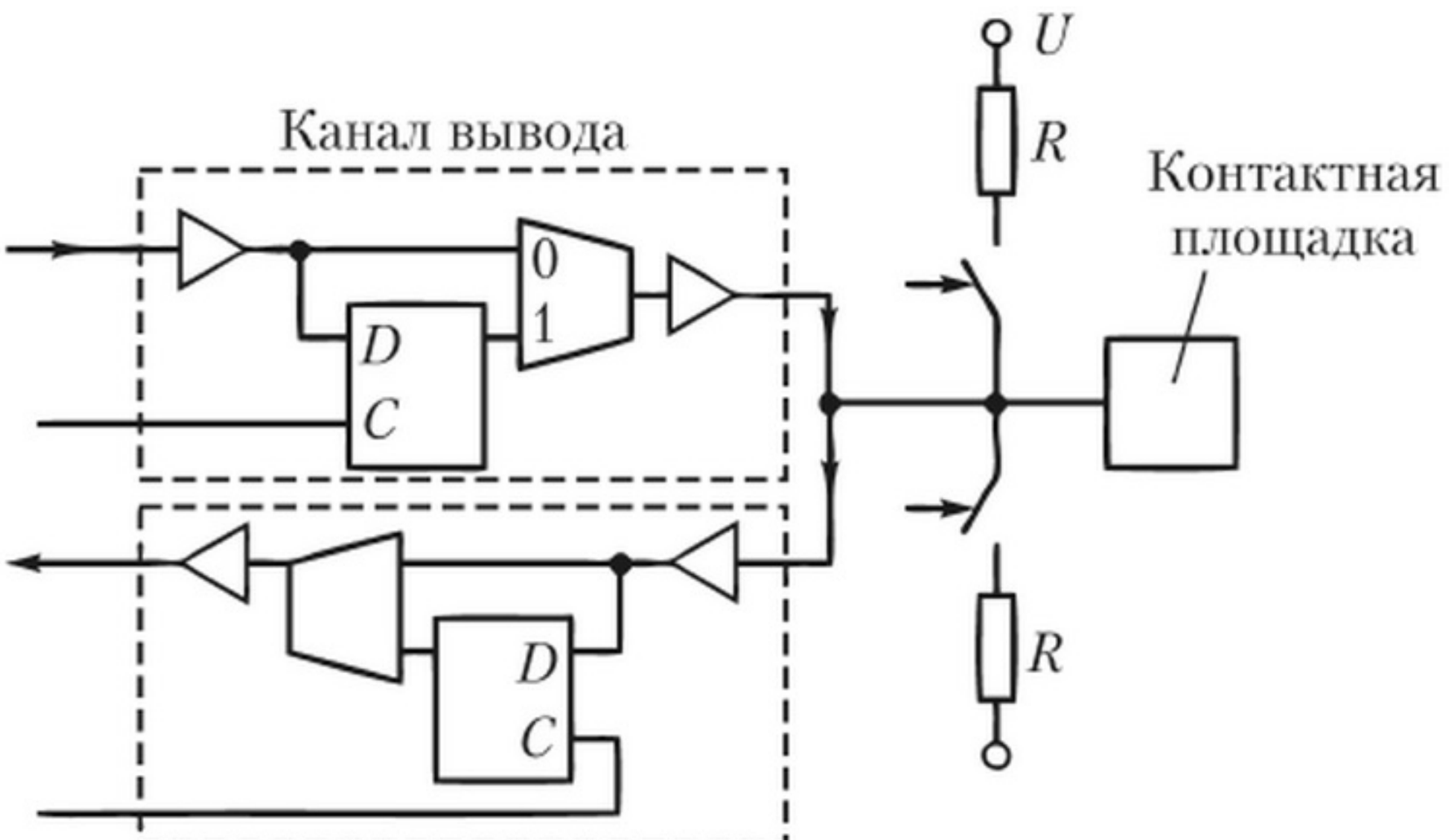


Рис. 4.36. **Блок ввода/вывода FPGA**

Выходной усилитель ПЛИС может переводиться в третье состояние, при этом выходной контакт микросхемы оказывается разомкнутым. Как известно, на разомкнутых контактах в КМОП-устройствах происходит накопление заряда, что приводит к появлению ложных сигналов. Чтобы этого не происходило, в микросхеме предусмотрено подключение контактной площадки через резисторы либо к высокому уровню напряжения, либо к нулевой точке (корпусу). Эту функцию выполняют два транзисторных ключа. Выбор того или иного ключа осуществляется при программировании.

Системы межсоединений FPGA. Линии связей в FPGA составляются из сравнительно коротких участков, не содержащих ключей — *сегментов*. Они могут быть разной длины, вертикальные и горизонтальные. Сегменты между собой могут соединяться с помощью ключей, замыкаемых при программировании. Малое количество сегментов ведет к недостаточно эффективному использованию ЛБ, слишком большое — к появлению большого числа программируемых. Короткие сегменты затрудняют реализацию длинных связей, а длинные — коротких. Поэтому целесообразна система связей с несколькими типами межсоединений — как длинных, так и коротких — для передач на разные расстояния.

В разработке FPGA с однократным программированием на основе перемычек типа *antifuse* ведущую роль играет фирма *Actel*. Логические блоки в FPGA этой фирмы располагаются в виде горизонтальных рядов, между которыми имеются трассировочные каналы (рис. 4.37).

В горизонтальном канале в четыре строки расположены сегменты различной длины. Их пересекают вертикальные сегменты. На пересечении вертикальных и горизонтальных сегментов находятся программируемые перемычки *antifuse*, позволяющие соединять вертикальные и горизонтальные сегменты. На рис. 4.37 они обозначены кружками.

Каждый ЛБ имеет один выход и четыре входа, два из которых расположены в верхней части ЛБ, а два — в нижней. Каждый из входов ЛБ соединен с одним из коротких вертикальных сегментов, причем верхние входы ЛБ соединяются с вертикальными сегментами, расположенными в верхнем ряду, а нижние входы — с сегментами в нижнем ряду. Выход каждого ЛБ соединен с собственным вертикальным сегментом, пересекающим сразу несколько

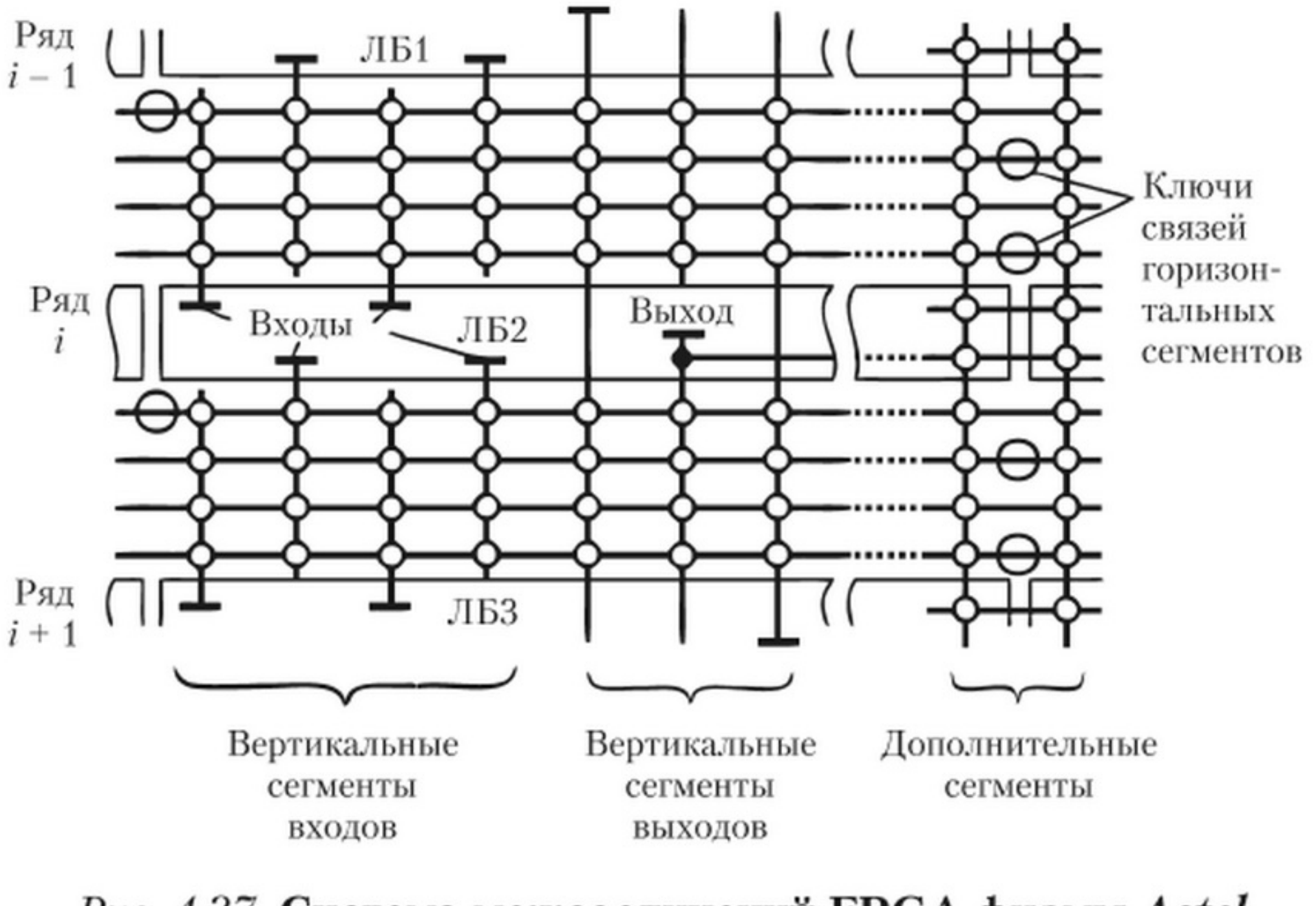


Рис. 4.37. **Система межсоединений FPGA фирмы *Actel***

горизонтальных каналов. Таким образом, для соединения выхода одного ЛБ с одним из входов другого близлежащего ЛБ достаточно замкнуть две перемычки: перемычку, соединяющую вертикальный сегмент выхода ЛБ с горизонтальным, и перемычку в точке пересечения этого горизонтального сегмента с соответствующим вертикальным сегментом входа.

Если ЛБ, выход и вход которых необходимо соединить, расположены не в одном столбце, имеются специальные программируемые ключи, позволяющие соединять концы горизонтальных сегментов друг с другом. Таким образом, можно увеличивать длину горизонтальных линий связи.

Кроме вертикальных сегментов входов и выходов ЛБ в вертикальных направлениях идут также дополнительные сегменты, пересекающие несколько рядов ЛБ и трассировочных каналов. Каждый такой вертикальный сегмент может соединяться с горизонтальными, которые он пересекает. Эти сегменты могут соединяться с выходами соседних ЛБ.

В горизонтальных каналах имеются непрерывные по всей длине сегменты, которые одним из своих концов подключены либо к напряжению питания, либо к корпусу. С помощью таких сегментов на ЛБ можно подавать сигналы лог. 1 и 0.

Таким образом, система коммутации содержит набор сегментов самого различного назначения и средства их коммутации, что обеспечивает большое разнообразие вариантов соединения ЛБ между собой.

Чтобы запрограммировать перемычку, т.е. замкнуть ее, к ней следует приложить повышенное напряжение $U_{пр}$. Так как перемычки расположены на пересечении строк и столбцов, то на одну из этих линий подается напряжение $U_{пр}$, а другая заземляется. Чтобы остальные перемычки остались целыми, перед этим все сегменты микросхемы заряжаются до уровня $U_{пр} / 2$. Таким образом, при подаче на одну из линий напряжения $U_{пр}$, а на другую — «0» только одна перемычка, находящаяся на их пересечении, попадет под напряжение $U_{пр}$, все остальные при этом попадут под напряжение $U_{пр} / 2$, не пробивающее перемычку.

4.12.4. СБИС программируемой логики «система на кристалле»

Современный уровень развития технологии производства интегральных микросхем позволяет разместить в одном кристалле ПЛИС несколько миллионов элементарных логических схем типа «2И-НЕ», «2ИЛИ-НЕ», при этом тактовая частота работы СБИС может достигать 1 ГГц и более. При таких возможностях в одном кристалле можно разместить целую цифровую систему. Это может быть процессор, память или интерфейсное устройство.

Несмотря на большое функциональное разнообразие в цифровых системах самого разного назначения, есть, тем не менее, функциональные узлы, присущие всем устройствам. Для реализации этих узлов, конечно, можно использовать и обычные средства программируемой логики. Но гораздо эффективнее их построение на основе специализированных областей, выделенных на кристалле для выполнения заранее определенных функций. Эти области носят название *аппаратных ядер*. Самыми очевидными функциональными узлами, без которых не может быть построено большинство цифровых систем, являются узлы, осуществляющие вычислительную обработку — сумматоры и, особенно, умножители. Реализация системы цифровой обработки информации также невозможна без ОЗУ. На примере этого узла наиболее очевидными становятся преимущества подхода к реализации СБИС с использованием аппаратных ядер. Так, аппаратное ядро ОЗУ емко-

стью 256—512 бит занимает площадь на кристалле, в десять раз меньшую, чем площадь, которая потребовалась бы для синтеза такого же ОЗУ средствами обычной программируемой логики. Кроме того, при этом в несколько раз повышается быстродействие такого ОЗУ. Ядра, предназначенные для реализации ОЗУ, выполняются с небольшой емкостью. Хотя существуют системы, требующие больших объемов памяти, делать аппаратные ядра ОЗУ большой емкости экономически нецелесообразно, так как это резко снижает степень универсальности таких СБИС и сужает рынок их сбыта.

Также в виде аппаратных ядер эффективно реализуются такие специализированные узлы, как аппаратные умножители. Так, умножитель двух 8-битных слов занимает площадь, равную 1/5 площади, требующейся для реализации такого же умножителя с помощью логических блоков FPGA. Но самым эффективным направлением успешного применения аппаратных ядер являются интерфейсные узлы, контроллеры и процессоры.

Основой для реализации СБИС «система на кристалле» стало направление FPGA. Многие специалисты стали дистанцировать СБИС с аппаратными ядрами как новый вид архитектуры, однако сами фирмы — производители ПЛИС обозначили их как усовершенствованные СБИС с архитектурой FPGA. И если раньше в состав микросхемы FPGA входили, главным образом, ЛБ и БВВ, то теперь пользователям предлагается и целый ряд других функциональных узлов, выполненных в виде аппаратных ядер.

Рассмотрим ряд современных микросхем FPGA на примере ряда СБИС фирмы *Altera*, которая наряду с другой фирмой *Xilinx* занимает до 86% мирового рынка ПЛИС. Фирма предлагает три семейства СБИС FPGA: Stratix, Arria и Cyclone. Первые ПЛИС семейства Stratix были анонсированы в 2002 г. и выполнялись по 120 нм технологическому процессу. Последние (по состоянию на 2012 г.) ПЛИС семейства Stratix, выпущенные в 2010 году, выполнялись уже по технологии 28 нм. Логическая емкость этих СБИС составляет около 1 млн эквивалентных логических элементов. Кроме логических блоков микросхемы этого семейства содержат аппаратные блоки контроллеров интерфейсов и внешней памяти, блоки встроенного ОЗУ, аппаратные блоки для цифровой обработки сигналов (DSP-

Содержание

блоки — *digital signal processor*) переменной точности. Для приема и передачи высокоскоростных (единицы гигагерц) цифровых сигналов в микросхемах есть блоки приемо-передатчиков (*трансиверов*), работающих с различными протоколами.

Первые СБИС семейства Arria появились в 2007 г. и реализовывались на базе технологии 90 нм, последние СБИС реализованы уже по технологии 28 нм. Это семейство относятся к ПЛИС среднего диапазона и предназначены для решения телекоммуникационных задач с повышенными требованиями по стоимости и энергопотреблению. Эти микросхемы имеют тот же набор блоков, что и ПЛИС семейства Stratix. Кроме этого, у последнего подсемейства Arria V, выпущенного в 2011 г., есть аппаратный процессорный блок, основой которого является двухъядерный процессор ARM Cortex A9, и блоки тактирования.

Для решения широкого круга задач выпускаются недорогие микросхемы семейства Cyclone, также обеспечивающие низкое энергопотребление и имеющие в своем составе тот же набор блоков. Первые СБИС этого семейства, реализованные по технологии 0,18 мкм, были выпущены в 2002 г. На сегодня фирмой *Altera* предлагается пять подсемейств Cyclone, последнее из которых Cyclone V появилось в 2011 г. и реализовано по технологии 28 нм. Несмотря на низкую стоимость, микросхемы содержат необходимый набор функциональных блоков для реализации на их основе целой цифровой системы. Чтобы наглядно представить вычислительную мощность этих СБИС, перечислим весь набор ресурсов, предоставляемых разработчику цифровой системы микросхемой Cyclone VSTD6:

- 41 509 адаптивных логических модулей;
- 110 000 эквивалентных логических элементов;
- 166 036 триггеров;
- 514 блоков встроенного ОЗУ;
- встроенное ОЗУ объемом 5140 Кбит;
- 512 математических блоков обработки цифровых сигналов;
- 224 умножителя 18×18;
- двухъядерный процессорный блок ARM Cortex-A9;
- 9 трансиверов со скоростью передачи 9 Гбит/с;
- контроллер внешней памяти;
- 2 контроллера PCI Express.

4.13. Средства автоматизированного проектирования электронных устройств

4.13.1. Автоматизированное проектирование электронных устройств

Нередко электронное устройство, разработанное чисто теоретически, на практике оказывается неработоспособным. Причинами этого могут являться не только ошибки в схеме, но и действие дестабилизирующих факторов, отклонения реальных значений параметров элементов от теоретических. Поэтому процесс разработки электронного устройства, как правило, включает в себя этап макетирования, на котором инженер-проектировщик изготавливает макеты отдельных частей, или устройства в целом, измеряет его реальные параметры. Использование систем автоматизированного проектирования (САПР) позволяет с помощью ПЭВМ провести весь цикл проектирования электронного устройства, который включает в себя следующие этапы:

- разработку структуры и принципиальной схемы устройства;
- анализ характеристик устройства в различных режимах работы с учетом возможного разброса параметров элементов;
- размещение элементов на печатной плате и разводка печатных проводников;
- разработку конструкции устройства и выпуск конструкторской документации.

Современные программы САПР работают в диалоговом режиме, т.е. разработчик — пользователь САПР — осуществляет ввод в ПЭВМ исходной схемы, ввод параметров элементов непосредственно на экране монитора с помощью мыши и клавиатуры. При этом система выводит на экран подсказки в виде меню, варианты продолжения диалога, указания пользователю, какую информацию и как нужно ввести, справочную информацию из файлов «*Help*». Широко применяется технология «*drag and drop*», заключающаяся в том, что пользователь щелчком мыши по пиктограмме осуществляет выбор нужного из предлагаемых системой элементов и установку его в выбранном месте чертежа схемы. Такой интерфейс называют «дружественным» по отношению к пользователю.

Для автоматизированного проектирования создаются библиотеки модулей, обеспечивающих математическое моделирование работы стандартных элементов электронных схем. Модели создаются как для дискретных полупроводниковых элементов — диодов, биполярных и полевых транзисторов, тиристоров, так и для целых устройств. К стандартным аналоговым устройствам относятся ОУ, компараторы, устройства на магнитных сердечниках (дроссели, трансформаторы). Список стандартных цифровых устройств гораздо шире. Сюда входят разообразные логические элементы, комбинационные устройства — дешифраторы, шифраторы, мультиплексоры, демультиплексоры, сумматоры, умножители, а также последовательностные устройства — разнообразные виды триггеров, счетчики, регистры, полупроводниковые ЗУ. На основе моделей отдельных элементов можно создавать макромодели, состоящие из множества элементов. В результате процесс создания принципиальной схемы разбивается на три этапа.

1. Выбор нужных элементов и устройств и размещение их в поле схемы.

2. Установка электрических связей между элементами с помощью проводников.

3. Ввод для каждого компонента схемы численных параметров, позволяющих с помощью имеющейся математической модели описать его работу.

Результаты работы устройства можно получить в виде графиков, временных диаграмм, численных значений выходных параметров. При этом получится оценка работы для номинальных значений параметров элементов схемы. В действительности любой элемент электронной схемы имеет некоторый диапазон изменения параметров, оговоренный в технических условиях. Так, например, действительные значения резисторов и конденсаторов могут принимать любые значения в пределах допустимых отклонений от номинала; для транзисторов оговариваются допустимые пределы изменения коэффициента β, максимальный ток коллектора и т.п.; для цифровых элементов важны пределы изменения задержки прохождения сигнала с входа на выход. Путем ввода крайних значений, соответствующих допустимым границам изменения параметра элемента, можно оценить работоспособность устройства с учетом возможного разброса параметров его элементов при его изготовлении.

Хотя моделирование работы устройств существенно упрощает процесс проектирования, необходимо отметить, что оно все равно не может абсолютно точно описать работу устройства. Это связано с упрощениями в математических моделях, описывающих работу элементов, а также наличием различных паразитных параметров — емкостей, сопротивлений, связанных с конструкцией устройства. Наиболее достоверными получаются результаты моделирования цифровых устройств.

После разработки принципиальной схемы осуществляется автоматизированная разработка печатной платы — одностороннее или двухстороннее размещение элементов и трассировка соединений (разводка проводников печатной платы). Наконец, разрабатывается конструкторская и технологическая документация, включая управляющие программы для технологического оборудования (например, управляющую программу для сверления отверстий в печатной плате).

4.13.2. Обзор современных пакетов автоматизированного проектирования электронной аппаратуры

В настоящее время имеется широкий набор пакетов программ для автоматизированного проектирования электронных устройств. Эти системы носят название системы EDA (*Electronic Design Automation — автоматизация проектирования электронных приборов*). Одним из первых и широко распространенных пакетов класса EDA явилась система P-CAD (*Personal Computer Aided Design — автоматизированное проектирование на персональном компьютере*), которая до сих пор еще применяется на российских предприятиях. В 2008 г. взамен системы P-CAD фирма-разработчик предложила свою новую систему Altium Designer, которая явилась дальнейшим развитием системы P-CAD. Система Altium Designer позволяет выполнять автоматизированное проектирование схем электронных устройств из микросхем и дискретных компонентов и проектирование устройств на основе ПЛИС. В последнем случае формируется программа для реализации межсоединений элементов и ячеек ПЛИС. Электронное устройство проектируется не только в виде принципиальной электрической схемы и печатной платы, но и в законченном конструктивном виде, представленном его 3D-моделью. Система интегрирована с САПР трехмерного твердотельного моделирования, такими, как

Solid Works, Pro/ENGINEER и др., куда можно передавать 3D-модель электронного устройства (электронный модуль, плата, блок).

В отличие от более ранних систем класса EDA, система Altium Designer является комплексной и обеспечивает разработку электронного устройства без использования других программ и систем. Весь цикл проектирования, называемый сквозным, включает разработку электрической схемы, компьютерное моделирование ее работы, разработку печатной платы, разработку комплекта конструкторской документации для производства.

Пакет Altium Designer состоит из двух программных продуктов:

- Altium Designer Custom Board Front-End Design — проектирование ПЛИС, схемотехническое проектирование и моделирование;
- Altium Designer Custom Board Implementation — проектирование печатных плат и ПЛИС.

Описание проектируемого устройства формируется на базе универсального языка VHDL (*Very high speed integrated circuits Hardware Description Language*) — язык описания аппаратуры интегральных схем, который является базовым для всех современных систем автоматизированного проектирования электронных устройств.

Формирование устройства осуществляется с помощью редактора схем, обеспечивающего создание сложных иерархических и многоканальных структур. Редактор позволяет выбирать из библиотеки готовые узлы, а также создавать свои собственные узлы и сохранять их в виде блоков. В результате сложная схема с огромным количеством элементов преобразуется в набор простейших подсхем. Всего библиотека содержит более 90 000 готовых компонентов электронных схем, которая постоянно обновляется. Безусловно, основным видом проектирования в последнее время стало проектирование цифровых устройств, однако редактор Altium Designer позволяет выполнять разработку также и цифроаналоговых схем.

После создания принципиальной электрической схемы можно выполнить моделирование ее работы, для чего в пакет включена библиотека из более чем 20 000 математических моделей компонентов, сигналов. Поскольку пакет ориентирован на разработку устройств, работающих на очень высоких частотах, предусмотрена возможность

проведения анализа с учетом импеданса и отражений сигналов.

Следующим этапом проектирования является разработка многослойной печатной платы, осуществляемая в соответствующем редакторе. Система оснащена программой-трассировщиком печатных проводников, позволяющей вводить правила трассировки и расположения проводников с учетом передаваемых сигналов, требований к толщинам проводников и зазорам между ними. Алгоритмы трассировки учитывают все требования, предъявляемые современными технологиями.

Для разработки ПЛИС система предлагает уникальную возможность отладки готового устройства с помощью специальной платы NanoBoard. Вместе с этой платой Altium Designer фактически является программно-аппаратным комплексом, позволяющим работать не только с виртуальным, но и с «живым» проектом.

Система Altium Designer интегрирована с многими другими системами класса EDA. Она обеспечивает импорт проектов — схем, плат и библиотек из систем PCAD, OrCAD, PADs, DxDesigner, Allegro PCB с преобразованием их в проекты Altium Designer. С помощью экспорта можно передавать результаты проектирования из Altium Designer в другие системы EDA.

Для формирования комплекта конструкторской и технологической документации система Altium Designer имеет широкий набор различных средств. Она поддерживает многие выходные форматы, воспринимаемые другими системами САПР, может генерировать списки соединений, разнообразные отчеты, на основе которых оформляются перечни элементов и спецификации по ЕСКД. В ее состав входит система технологического анализа и предпроизводственной доработки топологий, позволяющая выполнять ряд разнообразных проверок на технологичность проекта с возможностью автоматического устранения большинства ошибок. Также имеются мощные средства редактирования фотошаблонов, файлов сверловки и фрезеровки.

Кроме систем класса EDA, обеспечивающих проектирование и моделирование электронных устройств, имеется целый ряд систем, обеспечивающих автоматизированную разработку чертежей принципиальных электрических схем. Среди них есть как зарубежные пакеты, так и отечественные. Среди зарубежных наиболее известна система

AutoCAD фирмы Autodesk, которая обеспечивает автоматизированную разработку чертежей самого широкого спектра — машиностроительных, строительных, геодезических и многих других. В том числе эта система имеет в своем составе модули для разработки схем электронных устройств. Работы по проектированию чертежей принципиальных электрических схем можно выполнять также в отечественных пакетах автоматизированного проектирования, таких как КОМПАС-ГРАФИК (АСКОН), T-Flex (Топ-системы).

Контрольные вопросы и задания

1. Какие запоминающие устройства применяются для создания оперативной памяти компьютера, а какие для кэш-памяти, и почему?
2. Для какой цели применяются запоминающие устройства типа ROM и для какой — типа RAM, в чем их различие?
3. Сравните статические и динамические запоминающие устройства, укажите их достоинства и недостатки.
4. В чем отличие структуры $2DM$ от структуры $2D$?
5. В какой из запоминающих элементов — диодный или транзисторный — можно перезаписать информацию, и почему?
6. Какими способами можно стереть информацию из транзисторного запоминающего элемента?
7. Каковы основные направления применения флэш-памяти?
8. Каким образом в динамическом запоминающем устройстве выполняется регенерация?
9. В чем принципиальная разница между микропроцессором и последовательностными и комбинационными цифровыми устройствами?
10. Опишите работу микропроцессорной системы на рис. 4.9.
11. Каким образом осуществляется выборка команд программы в микропроцессорных системах?
12. Каково назначение флажков?
13. Объясните взаимодействие устройства управления, регистра команд и дешифратора команд при выработке серии управляющих импульсов.
14. Каково назначение внутренних регистров в микропроцессоре?
15. В чем отличие программно-управляемого обмена от обмена с прерываниями?
16. В чем заключается преимущество режима обмена с прямым доступом к памяти и когда он используется?
17. Каково назначение стековой памяти?

18. Поясните работу микропроцессора Intel 8086 (см. рис. 4.11).

19. Как происходит формирование адресов на основе сегментных регистров?

20. Как происходит мультиплексирование шин?

21. Каким образом определяется местонахождение операнда в косвенных режимах адресации?

22. Что такое трансляция?

23. Какие команды передачи данных используются в микропроцессоре?

24. Существуют ли арифметические команды с одним операндом?

25. Как соотносится разрядность операндов и произведения в двоичной арифметике?

26. Что такое двоично-десятичное число?

27. Как обозначается знак двоичного числа?

28. С помощью какой команды можно обнулить заданные разряды двоичного числа? Установить их в единицу?

29. Чем отличается арифметический сдвиг от циклического?

30. Разряды какого регистра анализируются в командах условного перехода?

31. Назовите основные тенденции развития микропроцессоров.

32. Каким образом кэш-память ускоряет работу микропроцессора?

33. В чем состоит особенность микроконтроллеров по сравнению с микропроцессорами?

34. В каких случаях удобно использовать параллельный интерфейс и в каких — последовательный?

35. Какой недостаток асинхронного обмена устраняет синхронный?

36. Какими возможностями буферных регистров не обладают шинные формирователи?

37. Что такое сигналы квитирования?

38. Каким образом приемник информации распознает начало данных при синхронном и асинхронном режимах?

39. Каким образом осуществляется проверка достоверности передаваемых данных?

40. Для чего в микропроцессорных системах используются сигналы прерывания?

41. Каким образом реализуется очередность обслуживания внешних устройств по сигналам прерывания с помощью приоритетного кольца?

42. Как происходит передача данных в режиме прямого доступа к памяти?

43. В каких режимах может работать программируемый интервальный таймер?

44. В ПЛМ на рис. 4.27 произошел обрыв в левом верхнем резисторе. Какую функцию станет реализовывать схема?

Содержание

45. В чем различие между ПЛМ и ПМЛ? Что общего?

46. Как распределяются базовые ячейки в канальном и бесканальном базовых матричных кристаллах?

47. В чем заключается программирование ПЛИС и какую роль в этом играют ключи?

48. Что такое «зернистость» и «функциональность» микросхемы FPGA? Какая между ними взаимосвязь?

49. Что такое логический блок в микросхеме FPGA? Какие он решает задачи?

50. Опишите работу микросхемы CPLD, изображенной на рис. 4.33.

51. Какова структура СБИС «система в кристалле»? Из каких узлов она состоит?

52. Какие системы для автоматизированного проектирования электронных устройств вы знаете?

Содержание

Глава 5
МИКРОАРХИТЕКТУРЫ ПРОЦЕССОРОВ

В результате изучения главы 5 студент должен:

знать

- основные виды структур современных микропроцессоров;
- принципы взаимодействия узлов в современных микропроцессорах;
- тенденции развития современных микропроцессоров;

уметь

- выполнять сопоставительный анализ работы узлов микропроцессора в различных микроархитектурах;

владеть

- информацией о последних достижениях в области микроархитектуры и технологии процессоров.

5.1. Микроархитектура P6

Развитие микропроцессорной схемотехники — достаточно сложный процесс. Его нельзя описать однозначно, например как постоянное наращивание числа внутренних устройств, рост их сложности. Идеи, которые были положены в основу построения более ранних процессоров, через годы и даже десятилетия могут быть реализованы в новых поколениях. Поэтому для лучшего понимания сути и принципов работы наряду с поколениями МП необходимо также рассматривать их *микроархитектуру*, т.е. принципы построения, состав основных внутренних устройств и их взаимодействие в процессе выполнения команд программы. Разработчики на основе общих принципов микроархитектуры создают конкретную реализацию процессора для данного технологического процесса. Эта конкретная реализация называется *ядром*. Микроархитектура описывает лишь общие принципы построения МП, а ядро представляет собой уже реально существующий кристалл с определенным составом и разрядностью внутренних узлов МП.

На основе одной и той же микроархитектуры может быть реализовано несколько ядер и даже поколений МП.

Ранее (см. параграф 4.5) рассматривалась микроархитектура МП Intel 8086 (она называлась просто «структура», сам термин «микроархитектура» был введен с появлением процессора Pentium Pro). Со времени его появления в микропроцессорной технике произошло много изменений, усовершенствований, в том числе и в принципах организации работы процессоров, однако наиболее значимые изменения произошли, начиная с процессора Pentium Pro.

P6 — суперскалярная, суперконвейерная архитектура, разработанная компанией Intel и лежащая в основе МП шестого поколения Pentium Pro, Pentium II, Pentium III, Celeron и Xeon. На основе доработанной микроархитектуры P6 в 2003 г. были выпущены процессоры с пониженным энергопотреблением Pentium M. Эта же архитектура была положена в основу двухъядерных процессоров Intel Core 2 Duo, которые появились в 2006 г. и пришли на смену процессорам Pentium 4.

Архитектура МП Intel 8086 была реализована на принципах CISC (*Complex Instruction Set Computer — вычислитель с полным набором команд*). Полный набор команд этого процессора составлял около 400. Схема его управления (см. рис. 4.11), приняв команду, декодировала ее и формировала последовательность управляющих импульсов для других узлов процессора. Такой же принцип выполнения команд был характерен для последующих процессоров Intel, вплоть до Pentium. Недостаток такого подхода заключается в том, что программы, состоящие из сложных команд, выполняемых за большое число тактов, трудно распараллелить, т.е. выполнять разными процессорными узлами параллельно.

В отличие от процессоров предыдущих поколений с CISC-ядром, процессоры архитектуры P6 выполняют сложные команды процессоров x86 (т.е. всех предыдущих процессоров Intel, а также других фирм, реализующих тот же набор команд) не напрямую, а предварительно декодируя их в простые внутренние микрооперации (МО). Такие операции выполняются быстро, а значит, и процесс вычислений легче распараллеливать и выполнять несколькими узлами процессора. Эта концепция проектирования процессоров именуется RISC (*Reduced Instruction Set Computer — вычислитель с сокращенным набором команд*).

Первым процессором архитектуры P6 стал Pentium Pro, анонсированный фирмой Intel 1 ноября 1995 г. Его разработка была выполнена под руководством российского ученого В. М. Пентковского — одного из создателей советского суперкомпьютера военного назначения «Эльбрус». При разработке микроархитектуры P6 использовались архитектурные решения, реализованные в этом проекте. Процессоры Pentium Pro выпускались параллельно с процессорами архитектуры P5 (Pentium и Pentium *MMX*). Новый процессор Pentium II был анонсирован 7 мая 1997 г., он реализован по архитектуре P6 и пришел на смену процессорам предыдущего, пятого, поколения.

5.1.1. Структура процессоров P6

Процессоры архитектуры P6 состоят из четырех основных подсистем (рис. 5.1):

- упорядоченной предварительной обработки;
- исполнения с изменением последовательности;
- упорядоченного завершения;
- памяти.

Подсистема упорядоченной предварительной обработки выбирает и преобразует команды x86 в последовательность более простых микроопераций, а также вычисляет возможные адреса переходов (технология предсказания переходов).

Подсистема исполнения с изменением последовательности обеспечивает исполнение МО (из которых состоит сложная команда x86) в оптимальном порядке и организует взаимодействие исполнительных устройств.

Подсистема упорядоченного завершения формирует выдачу результатов исполнения команд в порядке, предусмотренном программой.

Подсистема памяти обеспечивает взаимодействие процессора с оперативной памятью.

5.1.2. Подсистема упорядоченной предварительной обработки

К устройствам этой подсистемы относятся следующие узлы.

Дешифратор команд. Это устройство преобразует команды x86, т.е. сложные команды, реализованные во всех предыдущих процессорах Intel, в последовательность более простых RISC-микроопераций, исполняемых процессором.

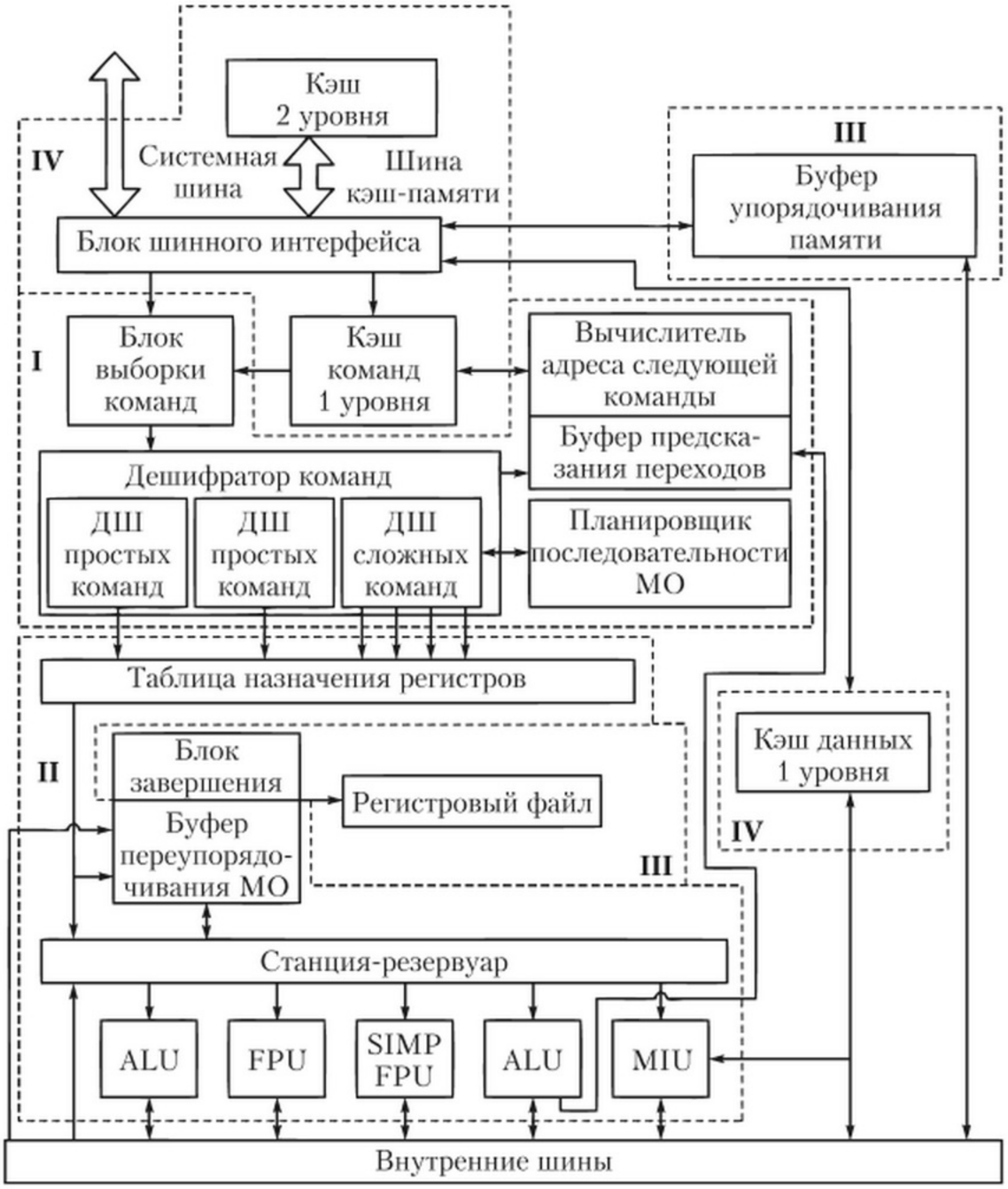


Рис. 5.1. **Функциональная схема процессоров P6:**

I — подсистема упорядоченной предварительной обработки; II — подсистема исполнения с изменением последовательности; III — подсистема упорядоченного завершения; IV — подсистема памяти

Оно включает два дешифратора простых команд, которые могут быть выполнены за одну МО, и один дешифратор сложных команд, для реализации которых нужно несколько (до четырех) МО.

Вычислитель адреса следующей команды. В ранних процессорах Intel адрес следующей команды вычислялся

с помощью регистра — ПС. Напомним, что он правильно указывает адрес следующей команды лишь при естественном (без условных и безусловных переходов) ходе программы. Если в программе появляются условные переходы, то адрес следующей команды вычисляется и загружается в регистр ПС. Эта процедура, как правило, требует времени, между тем операционное устройство процессора простаивает. В процессорах архитектуры P6 роль ПС играет специальный узел — вычислитель адреса следующей команды. Он анализирует прерывания и заранее вычисляет адрес команды, которая должна быть обработана следующей, на основании информации, хранящейся в таблицах переходов.

Буфер предсказания переходов. Это устройство хранит в таблице адреса предыдущих переходов. Проиллюстрируем принцип *предсказания переходов*. В любой более или менее сложной программе всегда имеются циклы, переход к началу которых осуществляется через команды условного перехода. Пусть, например, ветвь программы выполняется циклически до тех пор, пока значение некоторого счетчика изменяется от 1 до 99. При достижении счетчиком значения 100 происходит выход из цикла и переход к дальнейшему исполнению программы по указанному в команде адресу. Это значит, что после первого выполнения этой ветви в 97 случаях из оставшихся 98 переход будет выполнен к одному и тому же адресу программы. Таким образом, можно с большой долей вероятности предсказать адрес следующей команды.

Для предсказания используются как динамический, так и статический методы. Первый, по существу, означает, что в таблицах переходов адрес следующей команды уже имеется (а это означает, что эти команды, скорее всего, имеются в кэш-памяти и их считывать из ОЗУ уже не нужно). Если этих команд нет (они могли быть замещены в кэш-памяти другим набором команд), то подсистема упорядоченной предварительной обработки начинает выбор из памяти с последующим декодированием предсказанных команд. Причем эти действия выполняются процессором параллельно с исполнением ранее выбранных и декодированных команд. Если «предсказания» окажутся неверными, то весь этот кусок программы процессором просто аннулируется.

Статический метод соответствует случаю, когда адрес следующей команды приходится вычислять традиционным способом, как это было в первых процессорах Intel. Он

используется в том случае, если динамическое предсказание невозможно (в таблице переходов необходимая информация отсутствует).

Блок выборки команд. Устройство осуществляет выборку команд из памяти по адресам, подготовленным вычислителем адреса следующей команды.

Планировщик последовательностей микроопераций. Устройство хранит последовательности МО, используемых при декодировании сложных команд x86. При необходимости дешифратор сложных команд обращается к планировщику и получает нужную последовательность МО.

В дальнейшем развитии структуры P6 в подсистему упорядоченной предварительной обработки для дополнительного сокращения времени на считывание команд из ОЗУ был введен блок предвыборки команд. Он осуществлял предварительную выборку на основании таблицы переходов.

5.1.3. Подсистема исполнения с изменением последовательности

Данная подсистема обеспечивает оптимальную загрузку вычислительных узлов процессора и минимизацию их простоев. Это достигается за счет изменения порядка исполнения команд программы, не приводящего к искажению результата.

Рассмотрим возможность изменения порядка исполнения команд программы на примере.

Пример 5.1. Пусть в программе задана следующая последовательность команд:

1) $A = B + C$;
2) $K = A + M$;
3) $Z = X + Y$.

Предположим, что в процессоре имеется два блока для выполнения арифметических операций, которые могут работать параллельно. Как видно, команды 1 и 3 совершенно не зависят друг от друга. В команде 2 один из операндов (A) — результат выполнения команды 1, значит, она может быть выполнена только после нее. В такой ситуации целесообразно изменить порядок выполнения. Сначала лучше выполнить команды 1 и 3 двумя параллельно работающими блоками, а затем команду 2, когда операнд A уже будет готов.

К устройствам подсистемы исполнения с изменением последовательности относятся следующие узлы.

Таблица назначения регистров. Устройство задает соответствие между регистрами архитектуры x86, которые, как известно, указываются в командах в качестве источника одного из операндов и приемника результата, и внутренними регистрами процессора, применяемыми при исполнении МО.

Буфер переупорядочивания микроопераций. Он является ассоциативным запоминающим устройством, хранящим подготовленные к исполнению МО, из которых состоят команды программы. Это устройство называют также *пулом команд* (англ. *pool* — общий фонд, общий котел). Сюда заносятся подготовленные к исполнению МО вместе с операндами, для хранения которых в буфере есть 40 элементов, размером 256 бит. Каждый из элементов может хранить МО, два связанных с ней операнда, результат ее выполнения и несколько битов состояния.

Станция-резервуар. Устройство выполняет функции диспетчера, определяющего порядок выполнения МО, хранящихся в буфере переупорядочивания. Оно оценивает возможность изменения порядка исполнения МО и обеспечивает их выполнение в оптимальной с точки зрения производительности последовательности, направляя их в вычислительные блоки исполнительного устройства.

5.1.4. Набор исполнительных блоков

Для выполнения различных вычислений и действий с памятью в процессоре P6 имеется целый набор исполнительных блоков.

Два блока для выполнения операций над целыми числами — ALU, один из которых выполняет собственно вычисления — арифметические и логические операции, а другой предназначен для выполнения операций условного перехода, связанных с вычислением адреса следующей команды. Двоичные числа, которые рассматривались в параграфе 3.1, являются целыми. Однако, если бы процессоры компьютеров работали только с целыми числами, то невозможно было бы выполнять реальные математические расчеты. Поэтому, кроме целых, в вычислительной технике используются также вещественные числа, или числа с плавающей точкой (ПТ). Принцип их формирования рассмотрим на простом примере. Числу 20,625 в двоичной системе счисления соответствует число $10100{,}101 = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3}$. Его можно записать, выделив так называемые мантиссу — значащую часть и порядок:

$$10100{,}101 = 0{,}10100101 \cdot 2^5,$$

где мантисса — число 0,10100101; порядок — число 5.

Блок арифметики с плавающей точкой — FPU (*Floating Point Unit*) — выполняет операции над числами с ПТ.

Блок арифметики с плавающей точкой над множественными данными — SIMD FPU (*Single Instruction — Multiple Data Floating Point Unit*), т.е. одна команда арифметики с ПТ выполняется с множеством данных. Процессоры Pentium III и выше имеют в своем составе блок, исполняющий операции с ПТ сразу над несколькими числами. Эта технология имеет название SSE (*Streaming SIMD Extensions — потоковое SIMD-расширение процессора*). Набор таких команд был разработан фирмой *Intel* и впервые представлен в процессорах серии Pentium III как ответ на аналогичный набор 3DNow! фирмы *AMD*, который был представлен годом раньше. Технология SSE включает в архитектуру процессора восемь 128-битных регистров, содержимое каждого из которых трактуется как четыре последовательных числа с ПТ одинарной точности (32 бита).

Преимущество в производительности при использовании технологии SSE достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными.

Такая задача возникает, например, при 3D-моделировании, когда трехмерные тела должны отображаться на экране монитора с разных позиций просмотра. При этом отображение осуществляется с привязкой к базовой (мировой) системе координат, само же тело описывается местной системой координат. Чтобы отобразить его при перемещениях и вращениях, необходимо рассчитать координаты точек в базовой системе. Координаты вершин тела (например, параллелепипеда) при этом пересчитываются с помощью выражений типа:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

где X, Y, Z, W — координаты нового положения точки тела в базовой (мировой) системе координат; a_0, a_1, a_2, b_0, b_1, b_2,

c_0, c_1, c_2 — направляющие косинусы углов, образованных местными координатными осями повернутого тела относительно положения осей базовой системы координат; a_3, b_3, c_3 — координаты точки (000) местной системы координат тела в базовой системе координат; d_0, d_1, d_2, d_3 — координаты вектора, определяющего перспективу; x, y, z — координаты точки тела в местной системе координат.

При определении координат для каждой точки необходимо выполнить вычисления типа:

$$X = a_0 x + a_1 y + a_2 z + a_3 \times 1.$$

Аналогичные выражения необходимо вычислить для Y, Z и W. Очевидно, что для вычисления одной координаты необходимо выполнить четыре команды умножения. С помощью команды умножения *SIMD MULTIPLY* эти четыре операции можно заменить одной, которая вычисляет одну из координат радиуса-вектора точки (в данном примере это координата X). Поэтому такие операции также называются *векторными*.

Для реализации векторной операции предварительно необходимо подготовить операнды — в один 128-битный регистр записать четыре 32-битных операнда с плавающей точкой — a_0, a_1, a_2 и a_3, в другой — x, y, z и 1. Для этого в составе набора SSE-команд имеются специальные команды *перестановки* (англ. *shuffle* — переставить, тасовать). Команда *SIMD MULTIPLY* будет выполнена за одну МО, т.е. за один такт. Для дальнейших вычислений понадобится команда разупаковки содержимого 128-битного регистра, хранящего результат операции *SIMD MULTIPLY*, по четырем обычным 32-битным регистрам.

Блок интерфейса памяти — MIU (*Memory Interface Unit*) — вычисляет адреса данных, используемых командами, и формирует запросы к кэш-памяти для загрузки/выгрузки этих данных.

5.1.5. Подсистема упорядоченного завершения

Эта подсистема обеспечивает выдачу результатов выполнения команд в заданном исходной программой порядке. Подсистема состоит из следующих устройств.

Блок завершения. Устройство выдает результаты исполнения команд в той последовательности, в которой они поступили на исполнение, т.е. до ее изменения.

Регистровый файл. Устройство хранит результаты операций в том виде, в котором они хранились бы в регистрах процессора серии Intel x86 в результате выполнения команд программы.

Буфер переупорядочивания памяти. Устройство управляет порядком записи данных в память для предотвращения записи неверных данных из-за изменения порядка выполнения команд.

5.1.6. Подсистема памяти

Подсистема памяти осуществляет хранение данных и команд в быстродействующей кэш-памяти и взаимодействие процессора с оперативной памятью. К ней относятся следующие устройства.

Кэш данных первого уровня. Это память с малым временем доступа объемом 8 (для Pentium Pro) или 16 Кбайт (для более новых процессоров), предназначенная для хранения данных.

Кэш команд первого уровня. Эта память по своему быстродействию и объемам полностью аналогична кэш данных первого уровня, но используется для хранения команд.

Кэш второго уровня. В процессорах архитектуры P6 впервые кэш-память второго уровня была реализована как отдельный кристалл, но интегрированный в корпус процессора и соединенный с кристаллом центрального процессорного устройства отдельной независимой шиной. Такая архитектура стала называться *архитектурой с двойной независимой шиной*. Напомним, что ранее кэш второго уровня реализовывался только на основе внешних микросхем памяти. Новая архитектура позволила обеспечить работу кэш-памяти с тактовой частотой процессора (или вдвое меньшей). При этом число транзисторов у этого кристалла даже больше, чем у самого процессора (для сравнения, кристалл ЦПУ базового процессора P6 содержит 5,5 млн транзисторов, в то время как кристалл кэш-памяти второго уровня объемом 256 Кбайт — 15,5 млн).

Столь большое число транзисторов разъясняется статической природой памяти. Напомним, что у статической памяти для запоминания одного бита необходимо шесть транзисторов, в то время как у динамической памяти достаточно одного. Однако статическая память имеет более высокое быстродействие, нежели динамическая.

Кэш-память второго уровня в последующих процессорах архитектуры P6 составляла 256, 512, 1024 и 2048 Кбайт.

Блок шинного интерфейса. Это устройство обеспечивает управление системной шиной.

5.1.7. Порядок выполнения команд программы

Суперконвейерная архитектура, примененная в процессорах P6, делит ступени стандартного конвейера на более мелкие части, чем в ранних процессорах Intel.

Это значит, что каждая отдельная ступень будет содержать меньше схемотехнических элементов, и время между подачей на их вход управляющих сигналов и получением результата станет меньше. Это, в свою очередь, позволяет повысить тактовую частоту работы процессора.

Конвейер состоит из 12 стадий:

1–4 — определение адреса команды и ее выборка;

5–6 — декодирование команды и преобразование в МО;

7 — переименование регистров;

8 — запись микроопераций в станцию-резервуар;

9 — передача МО из станции-резервуара к исполнительным блокам;

10 — исполнение МО (один или несколько тактов);

11–12 — завершение команды: запись результатов в регистры.

Блок выборки команды считывает поток команд из кэша команд первого уровня порциями по 32 байта за такт. Строки выбираются на основании информации из буфера предсказания переходов. В считанной строке с помощью вычислителя адреса следующей команды определяется начало команды. Если команда находится в конце строки, считывается вторая строка, откуда выбираются недостающие байты этой команды. Выровненные 16 байт команд передаются на три дешифратора.

ДШК состоит из трех параллельных дешифраторов, два из которых — простые (дешифраторы простых команд) и один — сложный (дешифратор сложных команд). Задача каждого дешифратора — преобразование команд x86 в одну или несколько МО.

Простые дешифраторы преобразуют команду в единственную МО. Сложный дешифратор работает с командами, которым соответствуют от одной до четырех МО. Некоторые особенно сложные команды передаются планировщику последовательности МО, который, обращаясь

к таблицам, генерирует необходимую последовательность МО. Сначала все команды передаются дешифраторам простых команд. Если команда оказывается сложной, она передается дешифратору сложных команд либо планировщику последовательности МО. Такая пересылка слегка замедляет дешифрацию, но за счет буферизации с помощью станции-резервуара это незначительно сказывается на производительности процессора.

В принципе при бесперебойной и правильной загрузке командами дешифраторы способны генерировать в общей сложности шесть МО за такт, но, как правило, из всех трех дешифраторов за один такт выдаются три микрокоманды, соответствующие, в среднем, двум-трем командам x86, которые передаются в буфер переупорядочивания МО (пул команд).

Поскольку в классической архитектуре процессора x86 есть только восемь логических регистров общего назначения, то при параллельном выполнении нескольких команд весьма вероятна ситуация, когда одни и те же регистры будут использоваться в разных командах. Отсюда возникает необходимость использовать для разных команд разные физические регистры. Для этого предварительно нужно переназначить указанные в командах логические регистры на реальные регистры процессора, разные для разных команд. Ссылки в командах на логические регистры заменяются ссылками на реальные регистры, находящиеся в элементах буфера переупорядочивания МО (пула команд). Эта процедура выполняется в таблице назначения регистров.

Кроме того, здесь отслеживается порядок выполнения МО, соответствующих командам, таким образом, чтобы команды, использующие в качестве операндов результаты предыдущих команд, выполнялись позже. В случае необходимости записи результата в один и тот же регистр для двух разных команд они выстраиваются в очередь. Таким образом, более поздняя команда не может быть обработана до завершения более ранней команды.

Станция-резервуар выступает диспетчером и планировщиком МО, для чего она непрерывно сканирует буфер переупорядочивания МО (пул команд) и выбирает команды, готовые к исполнению, т.е. имеющие все исходные операнды. Результат выполнения возвращается назад в буфер и сохраняется вместе с МО до вывода. Поскольку резуль-

таты выполнения МО могут быть исходными операндами для следующих МО, они будут взяты непосредственно из пула команд. Порядок исполнения команд основывается не на их первоначальной последовательности, а на факте готовности команды и ее операндов к исполнению. Эта процедура называется *out-of-order — исполнение с изменением последовательности*. Такой порядок работы процессора был впервые реализован в советском суперкомпьютере «Эльбрус», а его ведущий разработчик В. М. Пентковский, ставший после развала Советского Союза руководителем проекта Pentium Pro, использовал эту идеологию при создании микроархитектуры P6. Исполнение команд программы с изменением последовательности их выполнения стало одним из краеугольных камней микроархитектуры P6, обеспечившим резкий подъем производительности процессора.

Всего станция-резервуар может одновременно выдать на исполнительные блоки, работающие параллельно, пять МО. Однако такая ситуация возникает довольно редко. Гораздо чаще на них выдается три МО за такт.

Рассмотрим, как будет работать процессор P6 при выполнении следующего фрагмента программы (номера регистров — условные):

(*r*1) ← (*mem* [*r*0]) (*mem* [*r*0] — ячейка памяти по адресу из *r*0)

(*r*2) ← (*r*2) + (*r*1)

(*r*5) ← (*r*5) + 1

(*r*6) ← (*r*6) – (*r*3).

При выполнении первой МО сначала будет обращение к кэш-памяти. Если фрагмент памяти с нужной ячейкой в кэше присутствует, он будет считан, и процессор продолжит выполнение программы. Если же данных в кэше нет, при традиционном подходе (в том числе и в процессорах Pentium) будет выполнено обращение к памяти, и пока данные через интерфейс не будут считаны, процессор будет простаивать. И это при том, что за это время он мог бы выполнить 20—30 МО, не требующих обращения к памяти.

Идеология процессора P6 заключается в том, чтобы сводить к минимуму возможные простои и выполнять максимально возможный в каждой конкретной ситуации объем полезной работы.

В рассматриваемом случае станция-резервуар обратится к пулу команд и выберет оттуда МО, для которых все операнды уже имеются в наличии. Так как выполнить вторую

МО до завершения первой нельзя, будут выбраны третья и четвертая, для которых все операнды уже готовы. Они будут отправлены на исполнительные блоки. После выполнения МО вместе с результатом будут возвращены в пул команд в свой 256-битный элемент, при этом бит состояния МО будет установлен в состояние «Микрооперация выполнена».

Выполненная МО будет находиться в пуле команд до тех пор, пока все впереди стоящие МО не будут исполнены, т.е. фрагмент программы не будет выполнен полностью. Лишь после этого МО можно удалить из пула команд, а результаты всего фрагмента программы записать в истинные регистры процессора (или в ячейки кэш-памяти — в зависимости от указаний исходной программы). Такой порядок выполнения команд программы называется *опережающим*.

Таким образом, процессор P6 исполняет команды программы, преобразованные во внутренние МО, только исходя из их готовности к выполнению, независимо от исходной последовательности команд в программе.

Чтобы закончить фрагмент программы, нужно из исполненных МО выбрать те, которые уже не будут влиять на ход программы, извлечь их из пула команд и записать результаты работы программы в соответствии с исходной последовательностью команд. Эту работу выполняет буфер завершения.

Процесс завершения занимает два такта. Буфер завершения сканирует пул команд и отыскивает МО, не влияющие на исполнение оставшихся в пуле команд. Затем он выстраивает их в первоначальную последовательность и выводит МО. При выводе результат МО записывается в регистровый файл, содержащий восемь регистров общего назначения и восемь регистров для данных с плавающей точкой. Именно они соответствуют логическим регистрам процессора архитектуры x86. Только после этого МО удаляется из пула команд. Всего буфер завершения может выводить три МО за такт.

Мы рассмотрели работу процессора при линейном порядке следования команд, однако в программах, в среднем, 10—20% составляют команды условного перехода. В этом случае для устранения непроизводительных потерь времени, связанных с обращением к основной памяти, необходимо выполнять команды, следующие после команд условных переходов. Заранее неизвестно, по какой из вет-

вей должен пойти вычислительный процесс, поэтому необходимо определить наиболее вероятный путь продолжения программы, выполнить вычисления, сохраняя их результаты в пуле команд, а после того как будет окончательно определено, правильно был угадан ход выполнения программы или нет, записать полученные результаты или аннулировать их.

Как уже отмечалось, за время обращения к памяти процессор может выполнить до 20—30 микроопераций, среди которых до пяти могут оказаться условными переходами. Определением возможных адресов переходов занимается буфер предсказания переходов. Для предсказания переходов используется расширенный алгоритм Yeh'a, позволяющий с вероятностью 90% правильно определить, будет ли выполнен возврат в начало цикла программы (команды которой уже имеются в кэше) или нет. В буфере хранятся адреса переходов, выполненных ранее. Очевидно, что если на ранних стадиях выполнения циклической программы переход был выполнен по одному и тому же адресу, то, скорее всего, очередной переход будет выполнен туда же.

Если предсказание окажется верным, то исполнение продолжится. Если же предположение ошибочно (цикл программы закончился), то частично выполненные МО удаляются из пула команд, команда с правильным адресом и следующие за ней выбираются, декодируются и отправляются на исполнение. Это приводит к существенному снижению производительности, напрямую зависящему от глубины конвейера — для архитектуры P6 в случае ошибочного предсказания перехода потери составят 4—15 тактов.

Проверка правильности ранее предсказанных переходов осуществляется следующим образом. При дешифрации к команде условного перехода прикрепляются оба адреса. Один из них — предсказанный — предполагается правильным, другой — нет. Как уже отмечалось, один из исполнительных блоков ALU специально предназначен для выполнения операций условного перехода. Он определяет правильный адрес перехода и сравнивает его с предсказанным. Если адреса совпадают, то все МО этой ветви программы, находящиеся в пуле команд, отмечаются как годные, и исполнение ветви продолжается. Если же адреса не совпадают, т.е. адрес перехода был предсказан неправильно, то блок ALU изменяет статус всех команд ветви на «Подлежащие удалению». При этом он передает

правильный адрес в буфер предсказания переходов, а тот, в свою очередь, — вычислителю адреса следующей команды. Работа конвейера процессора начинается с нового адреса команды.

Операции записи в память, в отличие от операций с логическими регистрами, выполненные один раз, уже не могут быть отменены. Поэтому они откладываются до той поры, пока вызвавшая их МО не будет удалена из пула команд. Для этого в P6 предусмотрен буфер переупорядочивания памяти. В нем по командам, выдаваемым блоком интерфейса памяти MIU, сохраняется информация о данных и адресах. Буфер пересылает данные в память только после того, как МО, произведшая запись в память, удаляется.

Таким образом, процессор P6 за счет рационального порядка исполнения команд программы обеспечивает существенное повышение производительности, при этом, несмотря на изменение порядка выполнения команд программы, результаты работы формирует в точном соответствии с программой.

5.2. Микроархитектура NetBurst

Микроархитектура NetBurst пришла на смену микроархитектуре P6. На ее основе был создан ряд процессоров Pentium 4, первые из которых появились на свет в конце 2000 г.

Основным препятствием в дальнейшем повышении производительности процессоров микроархитектуры P6 при тогдашнем уровне технологии в 0,18 мкм было число схемотехнических элементов в одной ступени 12-ступенчатого конвейера. Относительно большая длина ступени приводила к тому, что предельная частота, с которой мог работать такой конвейер, находилась в районе 1,1÷1,2 ГГц. Предвидя эту ситуацию, специалисты фирмы *Intel* заранее приступили к проектированию процессора с новой микроархитектурой, основу которой составил 20-ступенчатый конвейер. Поскольку число схемотехнических элементов в одной ступени конвейера существенно уменьшалось, это позволяло повысить тактовую частоту работы процессора. При проектировании процессора планировалось, что основным его приложением были бы задачи с минимальным числом ветвлений, среди которых основное место занимали бы

приложения, непосредственно связанные с Internet. Отсюда и громкое название микроархитектуры, которое можно перевести как «сетевой взрыв».

В основу микроархитектуры NetBurst вошли все перспективные наработки, применявшиеся ранее в архитектуре процессоров семейства P6, а также были разработаны и новые архитектурные решения.

NetBurst была положена в основу микропроцессоров седьмого поколения Pentium 4, Pentium D, Celeron и Xeon. Первые процессоры архитектуры NetBurst были анонсированы 20 ноября 2000 г., а 8 августа 2007 г. компания *Intel* объявила о начале действия программы по снятию с производства всех процессоров этой архитектуры. Им на смену пришли двухъядерные процессоры семейства Core 2 Duo.

5.2.1. Структура процессоров NetBurst

Процессоры архитектуры NetBurst состоят из четырех основных подсистем (рис. 5.2):

- входной;
- исполнения с изменением последовательности;
- исполнения;
- памяти.

Входная подсистема выполняет предварительную выборку данных, предсказание переходов и преобразование сложных команд x86 в простые внутренние МО.

Подсистема исполнения с изменением последовательности обеспечивает исполнение МО в оптимальном порядке.

Подсистема исполнения состоит из набора блоков, осуществляющих выполнение МО, и узлов, обеспечивающих взаимодействие исполнительных блоков.

Подсистема памяти состоит из группы узлов, обеспечивающих взаимодействие процессора с оперативной памятью.

5.2.2. Входная подсистема

К устройствам этой подсистемы относятся следующие узлы.

Блок предварительной выборки команд. Этот узел выполняет предварительную выборку команд программы (в кодах x86) на основании данных из входного буфера предсказания переходов и преобразование программного адреса команды в физический с помощью таблицы трансляции адресов, входящей в состав этого блока.

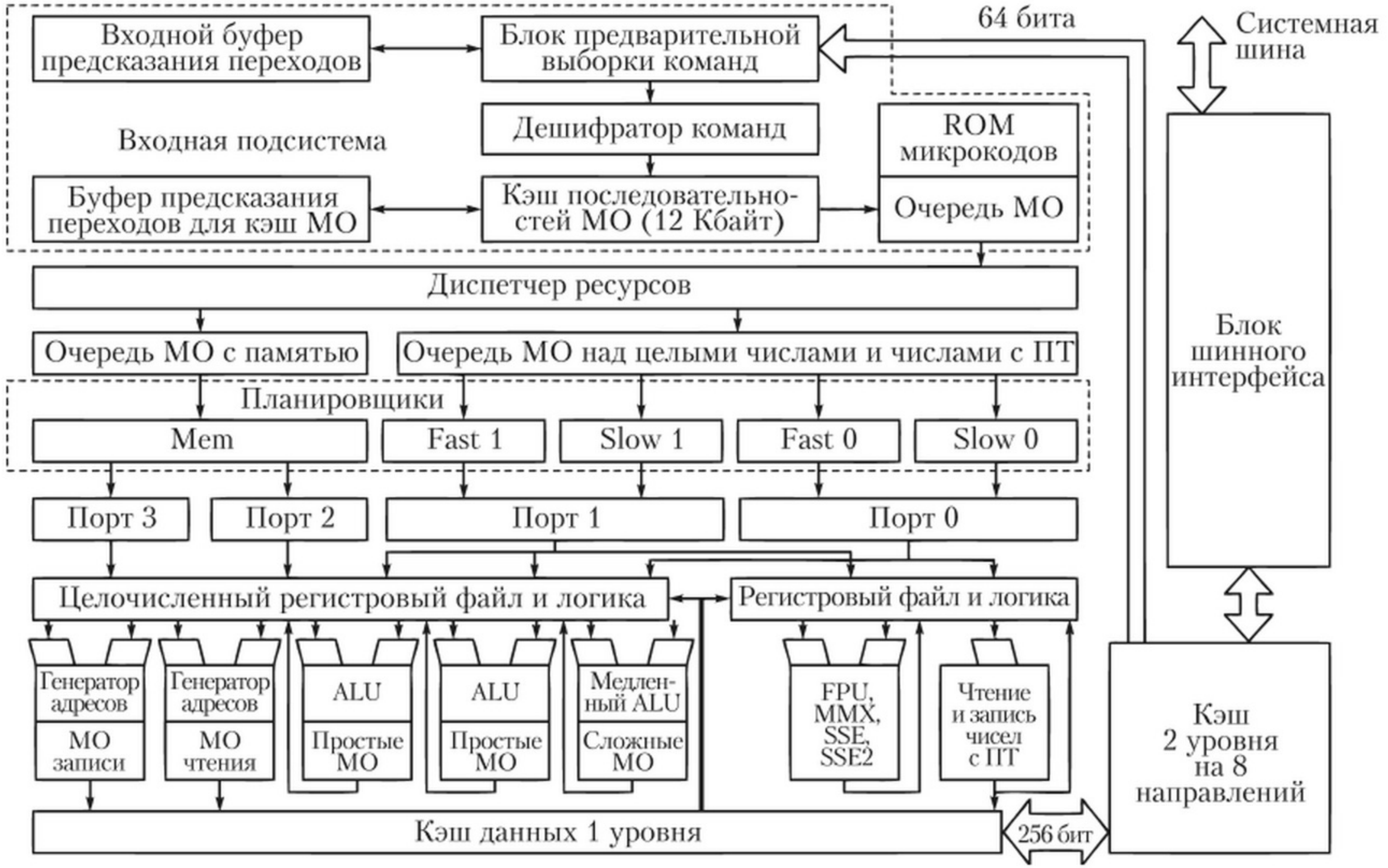


Рис. 5.2. **Функциональная схема процессоров NetBurst**

Входной буфер предсказания переходов. Узел хранит таблицу ранее выполненных переходов объемом 4 Кбайта. Это в восемь раз больше, чем у процессоров P6, что позволяет хранить более подробную историю предыдущих переходов. Также в NetBurst применен более совершенный алгоритм предсказания переходов, что вместе с расширенным объемом таблицы переходов обеспечивает вероятность удачного предсказания следующего адреса перехода около 93–94%.

Дешифратор команд. Как и в процессорах P6, узел выполняет преобразование сложных CISC-команд x86 в последовательность более простых RISC-MO, исполняемых процессором.

Кэш последовательностей MO. Этот узел не имеет аналогов в архитектуре P6, в чем заключается одна из инноваций архитектуры NetBurst. Кэш последовательностей MO обеспечивает исполнительные устройства постоянным потоком микроопераций, что минимизирует возможные простои. В него заносятся последовательности декодированных внутренних MO, готовых к исполнению, т.е. если в архитектуре P6 кэш команд первого уровня находился до дешифратора, то теперь он расположен после него, что существенно сокращает время выполнения команд x86 при их многократном повторном исполнении в циклических программах, поскольку избавляет от необходимости выполнять дешифрацию.

MO упорядочиваются в соответствии с несколькими предсказанными ветвями программы, для чего используется буфер предсказания переходов для кэш MO. Кэш последовательностей MO имеет объем 12 Кбайт и может хранить несколько достаточно длинных фрагментов программы — так называемых *трасс* (рис. 5.3). Он разбит на 2048 блоков, организованных в 256 наборов по восемь блоков. Каждый блок состоит из шести ячеек и может хранить до шести MO (т.е. по одной ячейке на MO). Если же команда x86 использует непосредственный способ адресации, т.е. содержит операнд, или адрес, то соответствующая ей MO может занять более чем одну ячейку.

Как правило, трасса начинается с MO, на которую следует переход в командах условного перехода (на рис. 5.3 начало трассы отмечено тегами *K*). При линейном следовании команд исходной программы трасса естественным

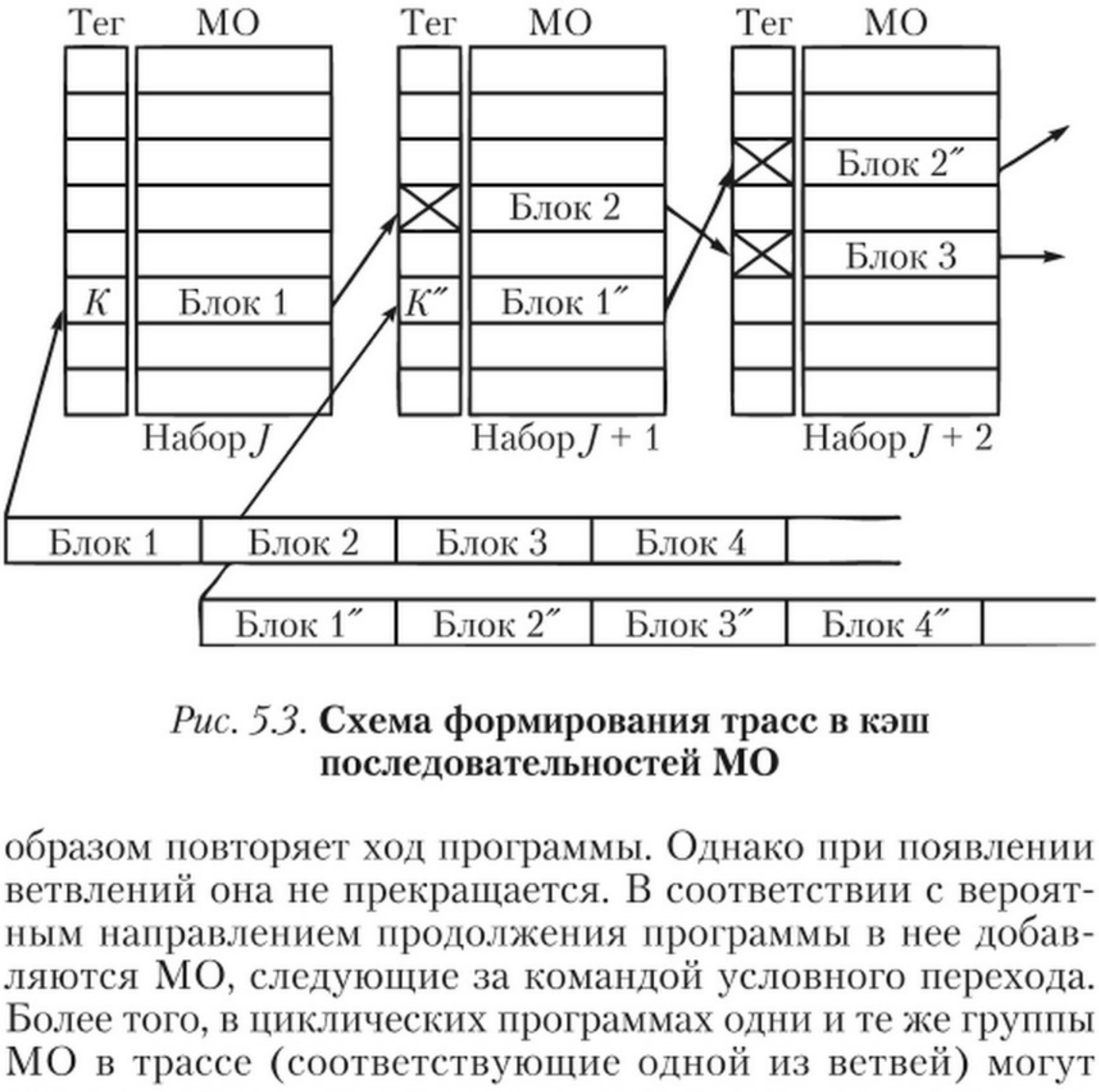


Рис. 5.3. **Схема формирования трасс в кэш последовательностей МО**

образом повторяет ход программы. Однако при появлении ветвлений она не прекращается. В соответствии с вероятным направлением продолжения программы в нее добавляются МО, следующие за командой условного перехода. Более того, в циклических программах одни и те же группы МО в трассе (соответствующие одной из ветвей) могут многократно повторяться и следовать друг за другом, т.е. трасса как бы «разворачивает» циклическую программу в линейную последовательность команд. Взаимосвязь между блоками устанавливается с помощью тегов. Максимальное число блоков в трассе — 64, после чего она обрывается. Если трасса формируется на линейном участке программы, то она «нарезается» на куски этой длины.

Буфер предсказания переходов для кэш МО. Этот узел, так же как и входной буфер предсказания переходов, хранит таблицу ранее выполненных переходов, объем которой меньше — 512 Байт. На основе данных таблицы переходов этого узла формируются трассы.

ROM микрокодов. Здесь хранятся микропрограммы исполнения сложных CISC-команд x86. Если CISC-команда может быть преобразована в набор, состоящий от одной до четырех простых RISC-МО, то дешифратор команд заносит их в кэш последовательностей МО. Для сложных CISC-команд в кэш вместо МО заносится адрес

ROM микрокодов, по которому располагается микропрограмма исполнения сложной команды.

Очередь микроопераций. Этот узел хранит очередь полностью декодированных МО (включая и сложные, декодированные с помощью ROM микрокодов), куда они заносятся из кэш последовательностей МО порциями по три микрооперации за такт. Узел является аналогом пула команд у процессора P6, но объем его существенно больше — до 126 МО. Кроме того, установление связей между логическими регистрами x86 и их физическими аналогами здесь еще не выполнено. Больший поток МО предоставляет процессору более широкие возможности по реализации стратегии динамического исполнения команд программы с изменением последовательности *out-of-order*.

5.2.3. Подсистема исполнения с изменением последовательности

К устройствам этой подсистемы относятся следующие узлы.

Диспетчер ресурсов. Узел распределяет вычислительные ресурсы для организации параллельного исполнения потока МО, устанавливает соответствие между логическими регистрами x86 и некоторыми физическими регистрами, выделенными специально для выполняемой МО, а также отслеживает состояние всех МО, находящихся в обработке. Он выполняет сортировку МО, формируя две очереди — по работе с памятью и всех остальных МО вычислительной обработки.

Очередь МО с памятью. Этот узел хранит очередь МО, связанных с обращением к памяти — чтения и записи.

Очередь МО над целыми числами и числами с ПТ. Этот узел хранит очередь всех остальных МО, которые выполняют вычисления над целыми и вещественными числами (с плавающей точкой).

Планировщики микроопераций. Эти устройства выполняют основную роль в реализации стратегии внеочередного исполнения команд *out-of-order*, обеспечивая оптимальную загрузку исполнительных блоков. Всего в архитектуре NetBurst пять планировщиков, которые специализированы для выполнения определенных наборов МО. Планировщики выполняют просмотр очередей МО и отбор МО,

готовых к исполнению. Микрооперация считается готовой, если ее операнды есть в наличии, независимо от ее порядка в трассе, т.е. команд в программе. Пересылка МО на исполнительные блоки выполняется через четыре порта, два из которых предназначены для МО с памятью, а другие два — для всех остальных. МО отправляются на исполнение таким образом, чтобы исполнительный блок освободился как раз к моменту их прибытия, а операнды были доступными, т.е. работа планировщиков организована так, что простои исполнительных блоков сведены к минимуму.

Специализация планировщиков МО следующая:

FAST 0 и FAST 1 — быстрые планировщики, которые работают с простыми арифметическими и логическими МО над целыми числами, выполняемыми блоками ALU, а также операциями пересылки;

SLOW 0 и SLOW 1 — медленные планировщики, которые работают со сложными арифметическими МО, выполняемыми блоками ALU, FPU, а также операциями пересылки;

MEM — планировщик МО по работе с памятью (чтение и запись).

5.2.4. Подсистема исполнения

К устройствам этой подсистемы относятся исполнительные блоки, собственно выполняющие преобразование данных — вычисления и пересылку.

Целочисленный регистровый файл и логика. Узел обеспечивает непосредственную передачу данных между блоками обработки целых чисел ALU. Регистровый файл содержит набор регистров для хранения результатов операций, которые, как правило, являются операндами для последующих МО. Однако за счет логики данные могут передаваться между исполнительными блоками также и минуя регистровый файл.

Генераторы адресов. Узлы обеспечивают вычисление адресов, по которым проводятся операции чтения и записи. Предназначены для взаимодействия с кэшем данных первого уровня.

Блоки ALU. Еще одна инновация в архитектуре NetBurst заключается во введении двух быстрых блоков ALU и одного медленного. Быстрые блоки ALU предна-

значены для выполнения простых МО, таких как сложение или логические операции. Они работают на частоте, вдвое большей, чем основная тактовая частота процессора. Поскольку именно быстродействие исполнительных блоков во многом определяет быстродействие процессора, эта инновация позволила существенно повысить конкурентные преимущества архитектуры NetBurst. Быстрое ALU состоит из двух «половинок», каждая из которых обрабатывает по 16 разрядов — младших или старших. Это сделано для уменьшения числа схемотехнических элементов в каждой половинке и поднятия частоты работы. За время такта основной частоты быстрое ALU выполняет два полутакта. Вычислительная обработка чисел в половинках ALU осуществляется со сдвигом. В течение одного полутакта младшая половинка осуществляет обработку младших разрядов последующей пары чисел, а старшая — старших разрядов предыдущей, при этом она использует результат (например, перенос), сформированный младшей половинкой при обработке младших разрядов этой пары чисел на предыдущем шаге. В результате, несмотря на то, что одна МО, взятая в отдельности, выполняется за два полутакта, при выполнении потока из n операций понадобится лишь $n + 1$ полутакт. Это дает заметный выигрыш в быстродействии, особенно учитывая, что результат МО может немедленно участвовать в следующей МО, как операнд, благодаря узлу «Целочисленный регистровый файл и логика». Для обеспечения работой быстрых ALU соответствующие им планировщики также работают на удвоенной частоте.

Медленное ALU обеспечивает выполнение всех остальных МО, т.е. тех, которые не могут выполнить быстрые ALU.

Регистровый файл и логика. Этот узел аналогичен узлу «Целочисленный регистровый файл и логика», только он предназначен для хранения операндов и результатов, а также непосредственной передачи данных для блоков обработки чисел с плавающей точкой.

Блоки обработки чисел с плавающей точкой. Для выполнения МО над числами с ПТ в архитектуре NetBurst имеется мощный блок. Также есть блок для выполнения операций чтения и записи.

Блок выполнения вычислений с числами с ПТ может выполнять следующие наборы МО:

FPU — сдвига, сложения и умножения чисел с плавающей точкой;

MMX — по обработке мультимедийных данных, эти МО впервые были введены в процессоре Pentium *MMX*;

SSE — набор микроопераций SIMD FPU архитектуры P6;

SSE2 — дополнительный набор микроопераций SIMD FPU, расширенный еще на 144 микрокоманды.

В процессорах на ядре Prescott, которые *Intel* представила в 2004 г., блок обработки чисел с плавающей точкой получил поддержку инструкций **SSE3**, в котором было добавлено еще 13 новых МО потоковой обработки данных.

5.2.5. Подсистема памяти

Подсистема памяти осуществляет взаимодействие с оперативной памятью. К ней относятся следующие узлы.

Кэш данных первого уровня. Это память объемом 8 Кбайт (в ранних процессорах, таких как Willamette) или 16 Кбайт (в более поздних) с малым временем доступа. В ранних процессорах оно составляло 2 такта, в более поздних с большим объемом кэш (ядро Prescott) время ожидания составляло 5 тактов основной частоты.

Кэш-память второго уровня. Это память большего, чем у кэш первого уровня, объема, но с большим временем доступа. Ее объем у процессоров на ядрах Willamette составляет 128 Кбайт (затем он был увеличен до 256 Кбайт), на ядрах Prescott — был доведен до 2048 Кбайт. Время доступа составляет 7 тактов для процессоров на ядрах Willamette и 18 тактов — на ядрах Prescott. Кэш-память второго уровня соединяется с кэшем данных первого уровня шиной шириной 256 бит.

Некоторые процессоры архитектуры NetBurst содержали также кэш-память третьего уровня объемом 2, 4 или 8 Мбайт.

Основная задача блоков кэш-памяти — обеспечить работой исполнительные узлы процессора. Поскольку передача данных из кэш осуществляется большими блоками, то для обеспечения высокой производительности используются широкие шины, что наряду с частотой работы, соответствующей номинальной тактовой частоте процессора, дает высокую скорость передачи. Так, при номинальной тактовой частоте 3 ГГц пропускная способность шины, связывающей кэш данных первого уровня с регистрами, составляет 48 Гбайт/с, а шины, связывающей кэши первого и второго уровня еще выше, — 96 Гбайт/с.

Блок шинного интерфейса. Этот узел управляет системной шиной. Новая шина, называемая Quad-Pumped Bus, что можно перевести, как шина с учетверенной скоростью передачи, способна передавать четыре 64-разрядных блока данных за такт. При этом изменения коснулись не только данных — она способна передавать два адреса за такт. Это существенно позволило поднять скорость передачи данных между процессором и оперативной памятью.

5.2.6. Описание конвейера

Как уже отмечалось, одной из инноваций архитектуры NetBurst стало хранение в кэш последовательностей МО целых кусков декодированных команд в виде трасс. Это позволило исключить из конвейера по выполнению команд программы этапы выборки команд и их дешифрации. Конвейер насчитывает 20 стадий в ранних процессорах (Willamette), в последних реализациях этой архитектуры (Prescott) число стадий было увеличено до 31. Рассмотрим 20-стадийный конвейер процессоров архитектуры NetBurst.

1–2. На первых двух стадиях происходит поиск в кэш последовательностей МО микрооперации, на которую указывает последняя выполненная. Напомним, что МО уже скомплектованы в трассы.

3–4. В течение следующих двух стадий осуществляется выборка и добавление в очередь МО до шести микроопераций. Если попадается сложная команда, то в очередь заносятся коды микропрограммы исполнения этой команды из ROM микрокодов. При этом кэш последовательностей МО работает на частоте вдвое меньше номинальной.

5. Пятая стадия служит для передачи МО по конвейеру, при этом никаких преобразований с ними не происходит.

6–8. Эти три стадии обеспечивают подготовку МО к исполнению. Диспетчер ресурсов выбирает из очереди МО по три микрооперации. Для них резервируются все необходимые ресурсы процессора: место в очередях — очереди МО с памятью и очереди вычислительных МО, элементы регистрового файла и буфера переупорядочивания инструкций. Подготовленные операции передаются в соответствующие очереди. Логические регистры, указанные в командах x86, переименовываются в физические, общее число которых больше, чем у процессоров P6—128.

9. Девятая стадия служит для формирования очередей МО — с памятью и вычислительных. При этом порядок

следования МО соответствует их положению в исходной программе. После чего они отправляются соответствующим планировщикам.

10–12. Эти три стадии выполняют планировщики. Тип микрооперации однозначно определяет, какому планировщику она достанется. Пять планировщиков выбирают МО из очередей независимо друг от друга, причем отбор осуществляется по принципу FIFO. В планировщиках формируются очереди МО, которые поступают в соответствующие порты, называемые портами запуска. Именно на этих стадиях реализуется стратегия внеочередного исполнения команд. МО выстраиваются в очереди только исходя из принципа готовности операндов к вычислениям.

13–14. На этих двух стадиях МО для исполнения снабжаются соответствующими операндами. Далее они через порты запуска пойдут на соответствующие исполнительные устройства. Порядок их исполнения уже не соответствует расположению в исходной программе, а определяется лишь готовностью.

15–16. Эти две стадии обеспечивают чтение операндов из регистровых файлов. Регистровые файлы работают на удвоенной частоте, т.е. на частоте работы быстрых блоков ALU.

17. Это стадия исполнения. Именно на ней происходит преобразование данных. МО вместе с операндами отправляется в соответствующий вычислительный блок, в котором и происходит обработка. Очень важную роль для ускорения процесса исполнения МО играет логика, имеющаяся в регистровых файлах. Благодаря ей результат только что исполненной МО может немедленно участвовать в следующей МО уже как операнд, т.е. устраняются задержки, которые могли бы быть вызваны отправкой результата в указанный регистр, а затем — чтением его оттуда.

18. На этой стадии вычисляются и устанавливаются признаки результата — флажки, которые необходимы для исполнения следующих МО программы. В частности, они являются входными данными для следующей стадии, на которой будет происходить проверка правильности предсказанных переходов.

19. Эта стадия зарезервирована для работы модуля предсказания переходов. Здесь происходит сравнение предсказанного для только что выполненной команды адреса

перехода с тем, который предсказывался ранее. Если предсказание оказалось ошибочным, алгоритм предсказания будет скорректирован. Таким образом модуль предсказания переходов набирает статистику точности, которая используется для корректировки модели предсказания.

20. Результат проверки, полученный на предыдущей стадии, направляется в дешифратор. После прохождения МО всех стадий конвейера она должна быть удалена, а результаты сформированы и записаны в память в соответствии с исходной программой. Только после удаления МО все ресурсы, зарезервированные за ней, освобождаются. Этот процесс называется отставкой МО. Эта операция осуществляется блоком завершения, аналогичным соответствующему устройству в архитектуре P6 (на рис. 5.3 он не показан). Отставка осуществляется группами по три МО — так же, как они были сгруппированы на стадиях 6—8.

5.2.7. Система повторного исполнения микроопераций

В процессорах архитектуры NetBurst имеется специальная система повторного исполнения МО, которая включается в работу в тех случаях, когда к моменту исполнения МО операнды оказываются не готовы. Такая ситуация может возникнуть, например, если она была отправлена планировщиком в конвейер, исходя из предположения, что операнд должен находиться в кэш данных первого уровня, а на самом деле его там не оказалось.

Как уже отмечалось, основная задача планировщиков состоит в максимально возможной загрузке исполнительных блоков работой. Поскольку конвейер длинный, необходимо формировать очередь из достаточно большого числа МО. При этом они отправляются в конвейер в такой очередности, чтобы к моменту их прихода на исполнительные блоки операнды были готовы. Если длительность исполнения МО (с учетом времени доставки операндов) известна, никаких проблем не возникает.

Пусть, например, идет цепочка МО, в которой результат предыдущей является операндом для последующей. В этом случае, если операнд самой первой операции хранится в регистровом файле, вся цепочка будет выполнена с максимально возможной скоростью. Иная ситуация возникает, если он хранится в ячейке памяти. В этом случае первая операция будет отправлена планировщиком на конвейер из предположения, что операнд находится в кэш данных

первого уровня и для его доставки требуется определенное для данного процессора число тактов, т.е. за время исполнения всех МО, стоящих перед ней, операнд будет доставлен в исполнительный блок. Представим, что нужной ячейки памяти в кэше не оказалось. Значит, операнд не будет доставлен и эта МО будет выполнена неправильно. Более того, поскольку приостановить конвейер невозможно, все последующие МО будут также выполнены неправильно.

Для исправления подобных ситуаций в архитектуре NetBurst имеется *система повторного исполнения МО* (англ. *replay* — переиграть). Она заключается в том, что параллельно с основными очередями планировщиков в процессоре есть резервные очереди системы повторного исполнения, в точности копирующие основные. МО с выхода планировщика попадает в основную очередь и резервную. В резервной очереди она движется параллельно основной. Если МО из основного конвейера выполнена успешно, ее копия в резервной очереди просто уничтожается. Если же к моменту ее исполнения оказывается, что операнд не был доставлен, копия МО повторно направляется в основной конвейер. МО, поступающая на повторное исполнение, имеет более высокий приоритет перед остальными и помещается в конвейере непосредственно после планировщика. Пройдя по цепочке конвейера, она выполняется еще раз. Такое многократное повторное исполнение может происходить десятки раз — до тех пор, пока операнд не будет получен (например, когда он доставляется из оперативной памяти).

Если эта МО оказалась первой в цепочке взаимозависимых команд, то все последующие МО также будут отправлены на повторное исполнение.

Иногда повторное исполнение взаимозависимых МО вызывает взаимоблокировки, приводящие к многократному повторному исполнению больших участков программы. Для исключения подобных ситуаций предусмотрен аварийный выход. После выполнения нескольких десятков циклов повторного исполнения соответствующие наборы МО отправляются в специальный буфер, чтобы освободить конвейер. При этом конвейер заполняется другими командами, а за время их исполнения в кэш данных первого уровня доставляются необходимые для отложенного участка программы операнды.

Наличие системы повторного исполнения — необходимая плата за стремление максимально загрузить рабо-

той исполнительные блоки при длинном конвейере. Хотя повторное исполнение и приводит к снижению производительности, тем не менее это позволяет избежать остановки и сброса конвейера, которые вызвали бы еще более серьезные потери производительности процессора.

5.2.8. Технология Hyper Threading

В некоторых процессорах, реализованных на основе микроархитектуры NetBurst, была введена возможность одновременного параллельного выполнения двух программ, или двух разных участков одной программы. Само название Hyper Threading означает «гиперпоточность», т.е. процессор может одновременно выполнять много потоков команд.

Один физический процессор, поддерживающий технологию Hyper Threading, рассматривается операционной системой компьютера как два разных логических процессора. Каждый из них имеет свой набор логических регистров, включая программный счетчик, которые формируются из группы физических регистров процессора, объединенных в регистровом файле. Так как физический процессор NetBurst имеет большой набор исполнительных узлов, работающих параллельно, включая и два работающих на удвоенной частоте быстрых блока ALU, появляется возможность загружать их командами от двух разных фрагментов программ.

Поскольку многие исполнительные узлы в процессоре имеются только в одном экземпляре, то, безусловно, между двумя потоками команд возможно возникновение конфликтов, когда обоим потокам требуется использовать один и тот же исполнительный узел, однако в целом такая работа позволяет более интенсивно их использовать. Конечно, эффективность такой работы зависит от того, из каких команд состоят потоки программ, т.е. насколько редко в процессе исполнения двух потоков будут поступать команды, одновременно требующие один и тот же исполнительный узел.

Рассмотрим преимущества технологии Hyper Threading на примере. Пусть процессор выполняет две программы, которым соответствуют два потока команд — поток *A* и поток *B*. Команды потока *A* загружают исполнительные блоки процессора так, как показано на рис. 5.4, *а*, а потока *B* — на рис. 5.4, *б*. Как видно, в разные такты оба

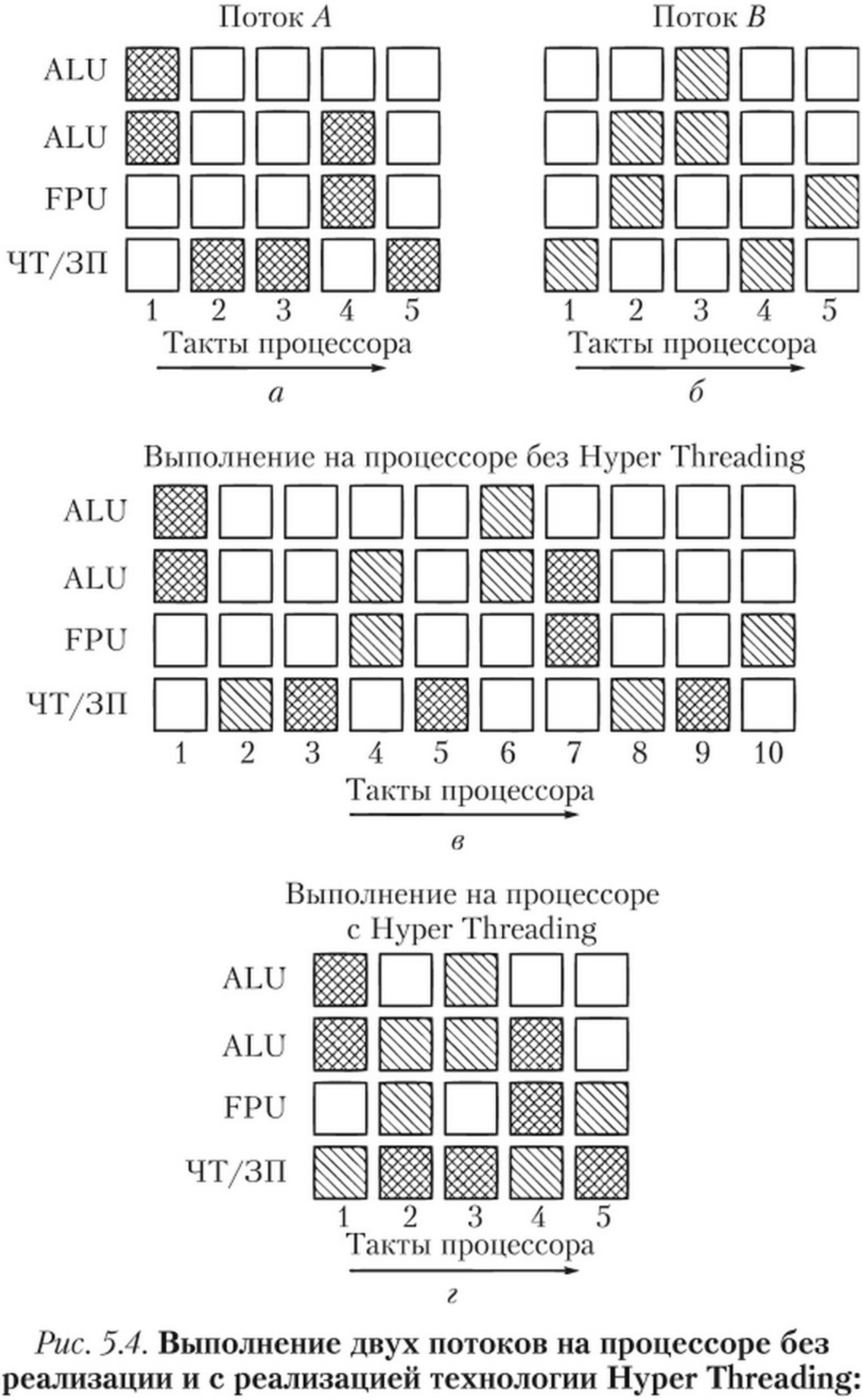


Рис. 5.4. **Выполнение двух потоков на процессоре без реализации и с реализацией технологии Hyper Threading:**

а — исполнение команд потока *A*; *б* — исполнение команд потока *B*; *в* — исполнение команд двух потоков без технологии Hyper Threading; *г* — исполнение команд двух потоков с технологией Hyper Threading

потока используют разные исполнительные блоки процессора. Если процессор не имеет технологии Hyper Threading, то при параллельном выполнении двух потоков он будет поочередно выполнять их команды (рис. 5.4, *в*). В общей сложности для этого понадобится 10 тактов. Если же процессор реализует технологию Hyper Threading, то для выполнения обоих потоков понадобится только 5 тактов (рис. 5.4, *г*).

5.3. Микроархитектура многоядерных процессоров

В начале XXI в. стало ясно, что дальнейшее увеличение производительности процессоров, построенных на основе одного ядра, будет ограничиваться растущим энергопотреблением. В то же время возможности технологии, уменьшение технологической нормы позволяли реализовывать в одном кристалле до несколько сотен миллионов транзисторов. В этих условиях ведущие разработчики процессоров, как *Intel*, так и *AMD*, приняли решение, что дальнейшее повышение производительности возможно за счет совмещения в одном кристалле двух и более процессорных ядер. С учетом того, что режим работы компьютера многозадачный, это позволяло существенно повысить его производительность.

В 2005 г. практически одновременно оба ведущих производителя представили свои двухъядерные процессоры: в марте фирма *Intel* — процессор Smithfield, в мае фирма *AMD* — процессор Opteron 2xx.

Первые двухъядерные процессоры просто объединяли в одном корпусе два ядра, которые ранее реализовывались в виде одноядерных процессоров. Так, процессор Smithfield практически совмещал в одном кристалле кремния два процессорных ядра, аналогичных Prescott, которые были подключены к общей системной шине. Напомним, что в процессоре Prescott была использована микроархитектура NetBurst. Затем *Intel* выпустила еще один двухъядерный процессор Presler с той же микроархитектурой. Он был реализован по технологии 65 нм и схемотехнически просто объединял в одном корпусе два кристалла, на каждом из которых располагался одноядерный процессор. На этом эра микроархитектуры NetBurst закончилась.

Будущее многоядерных процессоров фирмы *Intel* разработчики связали с развитием микроархитектуры P6. Пер-

вым двухъядерным процессором, ядра которого были реализованы на этой базе, стал Yonah, появившийся в январе 2006 г. В нем уже была общая для обоих ядер инфраструктура: кэш второго уровня и контроллер системной шины. Процессор Yonah можно рассматривать как последний представитель базовой микроархитектуры P6. Вслед за ним летом 2006 г. фирма *Intel* представила процессоры, построенные на основе новой микроархитектуры, специально разработанной для реализации многоядерных процессоров, которая во многом опиралась на базовую структуру P6, но имела и много нововведений. Она получила название Intel Core, в ней снова стал использоваться короткий конвейер, состоящий из 14 ступеней (в последних модификациях Pentium 4 на ядре Prescott число ступеней конвейера было доведено до 30).

Наконец, в конце 2008 г. *Intel* представила процессоры, построенные на базе новой микроархитектуры Nehalem, которая также, в свою очередь, стала развитием Intel Core. Разумеется, и Intel Core, и Nehalem — это микроархитектуры новых поколений с большим объемом нововведений. И все же отправной точкой для всех многоядерных процессоров фирмы *Intel*, начиная с 2006 г., стала микроархитектура P6.

5.3.1. Микроархитектура Nehalem

Микроархитектура Nehalem была разработана для многоядерных процессоров фирмы *Intel*, ориентированных на использование в высокопроизводительных персональных компьютерах и серверных системах. На ее базе выпускались двух-, четырех- и восьмиядерные процессоры. Она использует базовые наработки, созданные при проектировании микроархитектуры Intel Core, но имеет и целый ряд усовершенствований.

При разработке схемотехники своих многоядерных процессоров фирма *Intel* стала использовать совершенно новые материалы для реализации транзисторов. С уменьшением технологической нормы и ростом числа транзисторов в кристалле процессора уменьшалась толщина изолирующего слоя — диоксида кремния, пока не достигла значения 1,2 нм, или 5 атомов. По мере уменьшения толщины изолирующего слоя рос ток утечки, а вместе с ним и рассеиваемая мощность. Для снижения токов утечки нужно было увеличить толщину изолирующего слоя, который для сохранения

характера взаимодействия затвора и канала транзистора должен был обладать более высоким коэффициентом диэлектрической проницаемости. В 2007 г. инженеры фирмы *Intel* разработали технологию изготовления изолирующего слоя не из диоксида кремния, а из материала на основе гафния, которая позволила снизить ток утечки в десять раз. Однако новый материал оказался несовместимым с затвором транзистора. Транзисторы с новым изолирующим материалом работали менее эффективно, чем со старым. Тогда было предложено заменить и материал затвора: корпорация *Intel* открыла уникальное сочетание металлов, состав которого держится в строгом секрете. Новые технологии реализации транзисторов позволили перейти на более низкую технологическую норму — с 65 на 45 нм, что обеспечило увеличение числа транзисторов, размещаемых на той же площади. При этом снизилась мощность процессов включения—выключения транзистора и повысилась скорость переключения.

Существенным недостатком процессоров Intel Core стал их немодульный дизайн. Дело в том, что они изначально проектировались как двухъядерные полупроводниковые кристаллы. Поэтому четырех- и шестиядерные представители микроархитектуры Core просто собирались из нескольких двухъядерных кристаллов, а это приводило к затруднению взаимодействия между ними. Обмен данными между разрозненными ядрами организовывался через системную память. Конечно, такой подход к реализации обмена вызывал большие задержки, обусловленные ограниченной пропускной способностью процессорной шины.

В отличие от процессоров Intel Core микроархитектура процессоров Nehalem создавалась с ориентацией на модульный дизайн. Она включает в себя лишь несколько основных электронных модулей, из которых на этапе конечного проектирования и производства, как из строительных блоков, может быть собран итоговый процессор. Этот набор включает в себя процессорное ядро, кэш-память, контроллеры системной шины и памяти и т.д.

На этапе проектирования процессора необходимые модули собирают в едином полупроводниковом кристалле. В зависимости от назначения процессора в него включают то или иное число модулей. Например, процессор для персональных компьютеров может иметь два процессорных

ядра, одну кэш-память, а также один контроллер памяти и контроллер системной шины.

Серверный процессор может включать в себя уже восемь ядер, несколько контроллеров системной шины для объединения в многопроцессорные системы, кэш-память и контроллер памяти.

В соответствии с модульной концепцией в процессорах Nehalem выделяются два уровня. На первом расположены ядра процессора, второй объединяет устройства, общие для всех ядер. Прежде всего, это кэш третьего уровня (у процессоров Intel Core общим был кэш второго уровня). Второй компонент второго уровня у процессоров Nehalem — контроллер системной шины, третий — общий контроллер оперативной памяти.

Для Nehalem был разработан новый интерфейс, связывающий процессор с микросхемами материнской платы, обеспечивающими его взаимодействие с памятью и внешними устройствами (чип-сетом). Этот интерфейс основан на использовании новой системной шины, получившей название Intel QuickPath Interconnect (QPI). Шина QPI является двунаправленной и состоит из 20 пар линий, каждая из пар передает один бит — 16 для данных и 4 — для служебных сигналов. Шина QPI повышает скорость передачи данных по сравнению с ранее использовавшейся, начиная с процессоров Pentium 4, шиной Quad-Pumped Bus в 4—8 раз. Она используется как для связи ядер процессора с чип-сетом, так и для связи их между собой. Шина QPI обеспечивает невиданную для всех более ранних процессоров фирмы *Intel* скорость обмена данными — свыше 25 Гбайт/с.

В некоторые модели процессоров Nehalem во втором уровне также устанавливается общий графический контроллер.

Процессоры Nehalem могут поддерживать технологию многопоточной обработки Simultaneous Multi-Threading, которая известна под названием Hyper Threading. Напомним, что она использовалась еще в некоторых процессорах Intel Pentium 4 с микроархитектурой NetBurst, однако в процессорах с микроархитектурой Intel Core она отсутствовала. Благодаря наличию технологии Hyper Threading операционная система компьютера будет рассматривать двухъядерный процессор Nehalem как четыре, а четырехъядерный — как восемь отдельных логических процессоров.

Очень важной чертой микроархитектуры Nehalem стала высокая эффективность энергопотребления. Все схемотехнические компоненты процессора разрабатывались исходя из критерия энергосбережения. Внедрена инновационная система управления электропитанием ядер процессора. Фактически в него встроен специальный контроллер питания, который контролирует температуру и силу тока потребления каждого ядра и в зависимости от этого управляет их напряжением питания и тактовой частотой. В результате энергопотребление ядер, не задействованных в процессе выполнения программ, резко снижается.

Функциональная схема процессора с микроархитектурой Nehalem показана на рис. 5.5.

Как уже отмечалось, микроархитектуры процессоров Intel Core и Nehalem имеют много общего с P6, но в них также есть и целый ряд принципиальных нововведений, что позволило разработчикам Intel позиционировать их как принципиально новые поколения. Рассмотрим эти новые микроархитектуры на примере процессоров Nehalem.

Цикл выполнения команд здесь, как и в процессорах P6, начинается с работы буфера предсказания переходов (на рис. 5.5 не показан). Блок выборки команд на основе данных буфера осуществляет выбор наиболее вероятной ветви программы и ввод из кэш второго уровня целых блоков команд. Как и в процессорах P6, выборка команд из кэша выполняется 16-байтными блоками, причем время выборки составляет 1 такт. Поскольку блок имеет фиксированную длину, а команды x86 могут содержать разное число байтов, в нем необходимо выделить границы команд (в процессорах P6 выполнялось в процедуре выравнивания). В процессорах Nehalem эта процедура именуется *преддешифрацией*.

После преддешифрации команды порциями по четыре поступают в очередь команд (этот блок в процессорах P6 отсутствовал). Отсюда они с тактовой частотой теми же порциями поступают в дешифратор.

Дешифратор процессора Nehalem очень похож на P6. Только здесь простых дешифраторов не два, а три. Сложные (состоящие из 2—4 МО) команды декодируются в дешифраторе сложных команд. Наконец, команды, для декодирования которых требуется больше чем 4 МО, поступают в ROM микрокодов МО (аналогичный узел в P6 назывался планировщиком последовательностей МО).

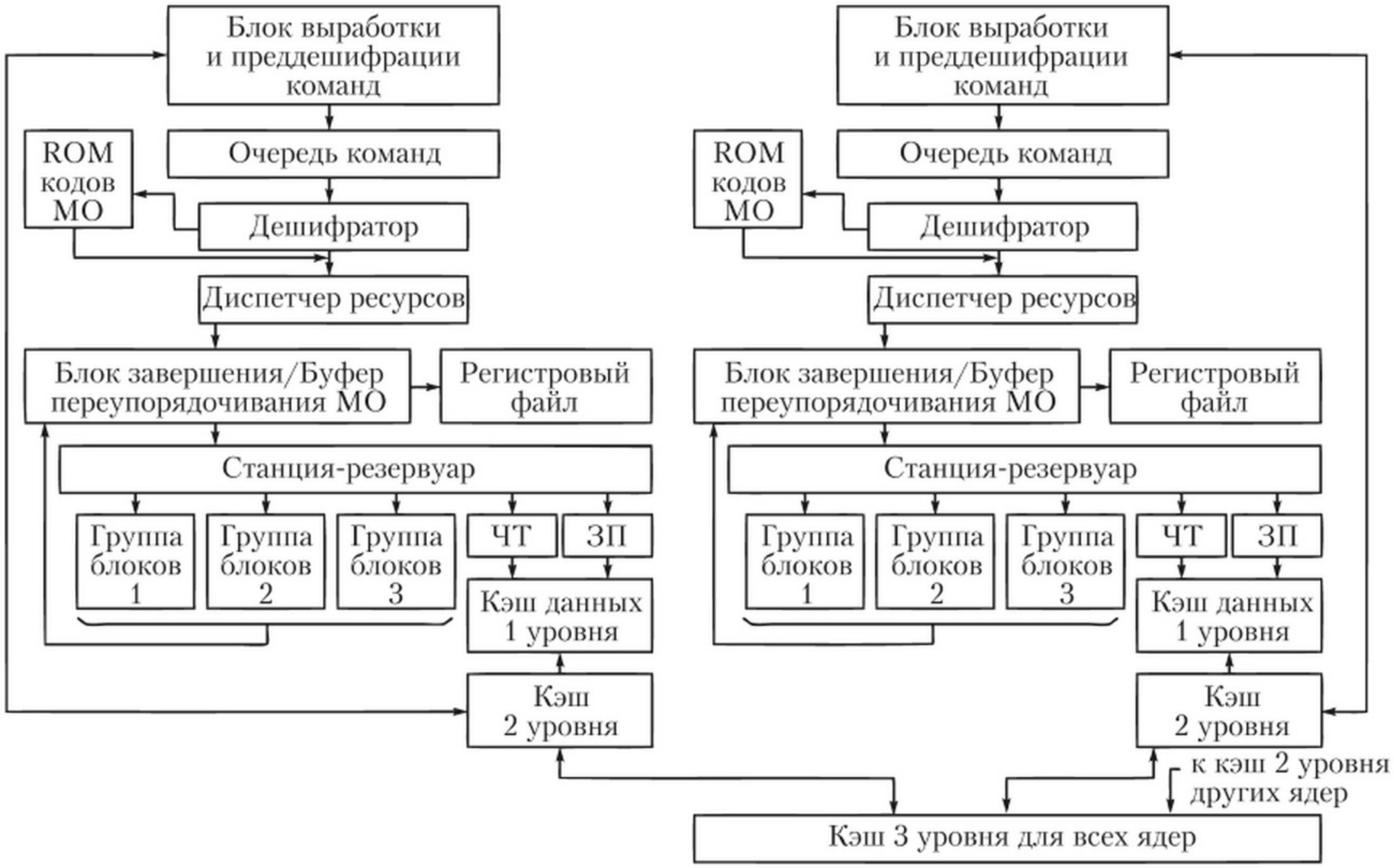


Рис. 5.5. **Функциональная схема процессоров с микроархитектурой Nehalem**

Из дешифратора МО порциями по четыре за такт поступают в очередь МО. Разумеется, это возможно только тогда, когда четыре команды x86 занимают не более 16 байтов.

В процессорах Intel Core и Nehalem введена *технология обнаружения циклов программ*. Для этого в состав блока выборки и преддешифрации команд введен блок обнаружения программных циклов. Он принимает непосредственное участие в процессах выборки команд из кэша и содержит буфер ранее выбранных команд программных циклов. Если блок обнаруживает в программе команды цикла, которые уже имеются в буфере, они непосредственно из буфера поступают на исполнение. В процессорах Intel Core это позволяет исключить из конвейера стадию выборки команд. В процессорах Nehalem технология обнаружения программных циклов была усовершенствована. Буфер ранее выбранных команд был расширен с 18 до 28 команд. Кроме того, его расположили не до, а после дешифратора, и хранить он стал уже декодированные команды. Можно сказать, что эта технология хранения уже декодированных участков программ была взята из архитектуры NetBurst. Это нововведение еще больше сократило цикл выполнения команды.

Еще одной инновацией в процессорах Intel Core и Nehalem стала технология так называемых слияний на уровне команд и МО. *Технология слияния команд* (MacroFusion) заключается в том, что некоторые пары команд, следующие друг за другом, сливаются в одну внутреннюю МО, которая затем будет исполняться именно как одна МО.

Первой командой, подлежащей слиянию, может выступать команда сравнения *CMP*. Напомним, что она выполняет вычитание двух операндов, формируя при этом признаки результата — флажки, но сам результат вычитания при этом не сохраняет. Она выполняется только для того, чтобы получить признаки результата, которые затем нужны для выполнения команды условного перехода, в качестве которой может выступить команда «перейти, если меньше» *JL*, «перейти, если больше» *JNL*, «перейти, если результат равен нулю» *JZ* или любая подобная. По существу, команда сравнения всегда выполняется вместе со следующей за ней командой условного перехода. Значит, их было бы целесообразно объединить в одну. Эту задачу и выполняет технология MacroFusion.

Для реализации таких «сдвоенных» МО в процессоре есть соответствующие исполнительные блоки. Следовательно, теперь за счет технологии слияния пять старых команд x86 преобразуются в четыре МО, которые процессор может передать на исполнение за один такт.

Кроме технологии MacroFusion, в процессорах также была внедрена *технология слияния микроопераций* (Micro-ops fusion). Она заключается в том, что в ряде случаев две МО сливаются в одну, содержащую два элементарных действия. Любая x86-команда, как правило, разбивается на несколько МО. Технология Micro-ops fusion позволяет объединить две одинаковые МО, относящиеся к двум разным командам x86, в одну слитую, которая и будет обрабатываться как одна МО. В результате число обрабатываемых МО снижается, а производительность процессора растет. В дальнейшем результат такой МО расслаивается, чтобы восстановить исходный ход программы.

После декодирования команд x86 и преобразования их в потоки МО начинается этап исполнения. В диспетчере ресурсов выполняется процедура закрепления за МО, использующими в качестве операндов одни и те же логические регистры x86, реальных физических регистров процессора, разных для разных МО (в структуре процессоров P6 эту роль играла таблица назначения регистров). Эта процедура необходима для организации внеочередного выполнения команд.

Затем поток МО порциями заносится в буфер переупорядочивания МО, откуда они поступают в станцию-резервуар, которая осуществляет проверку их готовности (в зависимости от наличия операндов) и отправку на исполнение. Данные для этого берутся либо из буфера переупорядочивания, либо из регистрового файла. Буфер переупорядочивания МО и блок завершения совмещены в одном устройстве, но работа блока завершения начинается уже после того, как МО будут выполнены, т.е. когда становится возможным восстановить исходный порядок их следования в соответствии с порядком следования команд x86 в исполняемой программе.

Из станции-резервуара через специальные порты диспетчеризации, аналогичные портам в микроархитектурах P6 и NetBurst, МО поступают к группам (кластерам — *cluster*) исполнительных блоков (рис. 5.6). Всего портов шесть, каждый связан со своими группами исполнитель-

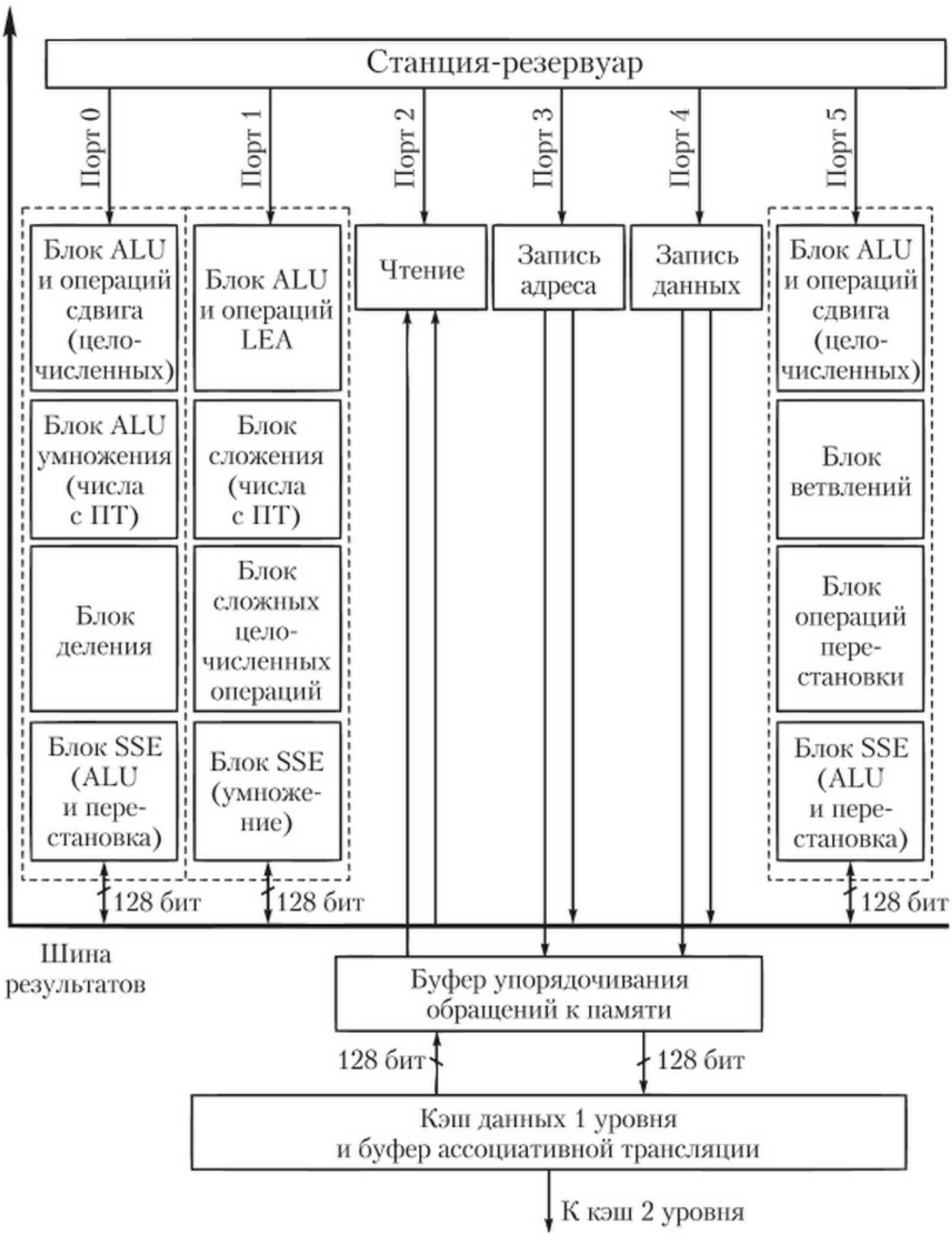


Рис. 5.6. **Группы исполнительных блоков процессоров Nehalem и их взаимодействие с ядром процессора**

ных блоков. Станция-резервуар может выдавать на исполнение по шесть МО за такт, т.е. через все шесть портов (разумеется, если в станции-резервуаре есть необходимые МО и соответствующие им исполнительные блоки свободны).

Порты 0, 1 и 5 обеспечивают загрузку групп вычислительных блоков, порты 2, 3, 4 — блоков для работы с памятью.

Таким образом, всего одновременно могут исполняться три вычислительные МО и три операции по работе с памятью.

Исполнительное ядро многоядерных процессоров Intel было спроектировано первоначально для микроархитектуры Intel Core. В микроархитектуре Nehalem были введены изменения, направленные на улучшение работы исполнительного ядра, но в целом оно базируется на разработках микроархитектуры Intel Core.

Исполнительные блоки в процессорах Intel Core и Nehalem представлены гораздо богаче, чем в предшествующих микроархитектурах даже самых последних модификаций — как P6, так и NetBurst. Именно они в конечном итоге обеспечивают высокую производительность процессора, и вся его структура направлена на обеспечение эффективной загрузки исполнительных блоков.

В каждой группе вычислительных блоков есть блоки для работы с целыми числами, числами с ПТ и выполнения потоковых векторных операций над множественными данными (SIMD).

Для наглядного представления возможностей исполнительных блоков процессоров Nehalem по выполнению различных МО воспользуемся табл. 5.1.

Таблица 5.1

Порт	Тип вычислений	Выполняемые операции	Число тактов
Порт 0	Целочисленный	Арифметико-логические (ALU) Операции сдвига	1 1
Порт 0	Векторный	SIMD — арифметико-логические (ALU) SIMD — перестановка	1 1
Порт 0	С плавающей точкой	Умножение Умножение с двойной точностью Операции FP/SIMD/SSE (пересылка, логические) Операции перестановки Операции деления и вычисления квадратного корня	4 5 1 1 4 бита за такт

Окончание табл. 5.1

Порт	Тип вычислений	Выполняемые операции	Число тактов
Порт 1	Целочислен-ный	Арифметико-логические (ALU) Пересылка Умножение	1 1 1
Порт 1	Векторный	SIMD — умножение SIMD — сдвиг SIMD — вычисление абсо-лютной разности SIMD — сравнение строк	1 1 3 3
Порт 1	С плавающей точкой	Сложение	3
Порт 2	Целочислен-ный	Чтение	4
Порт 3	Целочислен-ный	Запись адреса	5
Порт 4	Целочислен-ный	Запись данных	—
Порт 5	Целочислен-ный	Арифметико-логические (ALU) Операции сдвига Операции перехода	1 1 1
Порт 5	Векторный	SIMD — арифметико-логиче-ские (ALU) SIMD — перестановка	1 1
Порт 5	С плавающей точкой	Операции FP/SIMD/SSE (пересылка, логические)	1

Как видно из табл. 5.1, подавляющее большинство операций, включая и векторные, выполняется за один такт. Исключение составляют традиционно сложные для процессоров операции — умножения и деления. Тем не менее, даже операция умножения с двойной точностью (64 бита) выполняется за пять тактов.

У ядер процессоров Core и Nehalem есть три блока ALU для целочисленных вычислений, каждый из которых может исполнять однотактовые 64-битные целочисленные операции. Простые арифметико-логические целочисленные операции могут выполнять три блока. Все они способны выполнять операции над 64-битными операндами. Поскольку они

расположены на разных портах, то теоретически (при наличии соответствующих команд в исходной программе) ядро процессора Nehalem может за один такт выполнять сразу три операции над целыми 64-разрядными числами.

Для выполнения сложных операций над целыми числами здесь, как и в ядре процессоров P6, есть один 64-разрядный блок.

Пятый порт содержит блок для вычисления адресов переходов в операциях условного перехода (блок ветвлений). Он вместе с блоком ALU может работать параллельно над выполнением команд x86, объединенных в процессе реализации технологии MacroFusion.

Еще одной особенностью процессоров Core и Nehalem стало наличие специального блока для выполнения операций деления. Он реализует операции деления как над целыми числами, так и над числами с плавающей точкой. По сравнению со всеми предыдущими процессорами Intel используется усовершенствованный алгоритм деления, позволяющий за каждый шаг вычислять 4 бита частного, в то время как в более ранних процессорах вычислялись только 2 бита. Наличие специального блока деления обеспечивает повышение производительности научных вычислений, преобразования трехмерной графики и других функций с высоким содержанием математических вычислений.

В ядрах Core и Nehalem используются два исполнительных блока для вычислений с плавающей точкой, способных осуществлять как обычные (с двумя операндами), так и векторные арифметические операции. Блок, расположенный на порту № 1, выполняет сложение и другие простые операции в форматах: для обычных операций — с одинарной (32 бит) и двойной (64 бит) точностью; для векторных операций — с одинарной (для четырех операндов) и двойной (для двух операндов) точностью. Исполнительный блок на порту № 0 осуществляет операции умножения в этих же форматах.

Одно из важнейших улучшений в ядрах процессоров Core и Nehalem — блоки потоковых векторных операций над множественными данными (SIMD). Архитектура обеспечивает полноценную 128-битную обработку во всех векторных блоках. В предыдущих процессорах Intel у SIMD-блоков не было возможности работы с трехоперандными и 128-битными командами. Из-за этого они разбивались на пары команд, работающих с 64-битными операндами,

которые выполнялись за два такта. Теперь этот недостаток был устранен, и SSE-команды, ранее выполнявшиеся за два, стали выполняться за один такт.

SIMD-блоки процессоров Intel Core и Nehalem стали поддерживать новый расширенный набор команд технологии SSE – SSE4. Он представляет собой самое значительное усовершенствование набора команд для мультимедийных приложений с 2001 г. Новый набор позволил существенно повысить производительность программных приложений по обработке графики, трехмерных изображений, видео, систем кодирования, компьютерных игр. Кроме того, он эффективно используется в программных приложениях, связанных с выполнением алгоритмов сжатия звука, изображений, данных и многих других приложений.

Полный набор команд SSE4, введенный у процессоров Intel Core, насчитывал 47 команд, у процессора Nehalem дополнительно было введено еще 7 команд. Все они работают с 128-битными регистрами.

Новый набор SSE4 содержит команды ускорения для кодирования и декодирования видеофайлов, вычисления векторных примитивов на основе целых чисел, вставки и извлечения 8-, 16-, 32- и 64-битных полей в 128-битные регистры для чисел с плавающей точкой, скалярного умножения векторов, смешивания для формирования операндов, проверки отдельных битов и групп, округления.

Чтобы получить представление о сложности действий, выполняемых исполнительными SIMD-блоками, кроме уже известного скалярного произведения векторов, рассмотрим еще одну команду.

При кодировании и сжатии видеоинформации кадры разбиваются на небольшие фрагменты и определяются вектора их смещения на изображении от кадра к кадру. Задача определения векторов смещения решается путем вычисления для группы соседних пикселей (из которых состоят фрагменты) сумм абсолютных значений разностей байтов, кодирующих информацию о пикселях. Для подобных вычисления в наборе команд SSE4 есть команда *MPSADBW*. Она выполняет вычисление восьми сумм абсолютных значений разностей (*SAD*) смещенных 4-байтных беззнаковых групп A и B:

Если $\{A_0, A_1, ..., A_{15}\}$ и $\{B_0, B_1, ..., B_{15}\}$ — указанные в качестве операндов группы байтов, то результатом операции

станут восемь сумм $\{SAD_0, SAD_1, SAD_2, ..., SAD_7\}$, вычисляемых по правилам:

$$SAD_0 = |A_{(s1+0)} - B_{(s2+0)}| + |A_{(s1+1)} - B_{(s2+1)}| + |A_{(s1+2)} - B_{(s2+2)}| + |A_{(s1+3)} - B_{(s2+3)}|;$$

$$SAD_1 = |A_{(s1+1)} - B_{(s2+0)}| + |A_{(s1+2)} - B_{(s2+1)}| + |A_{(s1+3)} - B_{(s2+2)}| + |A_{(s1+4)} - B_{(s2+3)}|;$$

...

$$SAD_7 = |A_{(s1+7)} - B_{(s2+0)}| + |A_{(s1+8)} - B_{(s2+1)}| + |A_{(s1+9)} - B_{(s2+2)}| + |A_{(s1+10)} - B_{(s2+3)}|,$$

где $s1$ и $s2$ — смещения от 0 до 3, определяющие, для каких байтов выполняются вычисления.

Новые команды, введенные в процессорах Nehalem и дополнившие набор SSE (он получил название SSE4.2), в отличие от всех ранее введенных SIMD-команд были ориентированы не на реализацию задач мультимедиа, а на решение более узких и специфических задач, благодаря чему они получили название «ускорители прикладного назначения». Пять новых команд предназначены для обработки текстов и, в частности, XML-файлов. Шестая обеспечивает подсчет 32-разрядных контрольных сумм, которые широко применяются для проверки правильности переданной по каналам связи информации. Седьмая обеспечивает подсчет числа единичных битов.

Важную роль в повышении производительности процессоров Intel Core и Nehalem сыграли буферы ассоциативной трансляции. Эти устройства представляют собой буферы сверхоперативной памяти процессора, которые используются для ускорения трансляции адреса виртуальной памяти (указанного в программе) в адрес физической памяти. Они содержат фиксированный набор записей, в каждой из которых находится соответствие адреса страницы виртуальной памяти адресу физической памяти. Если адрес в буфере отсутствует, процессор вынужден выполнять вычисления для определения физического адреса памяти, что занимает гораздо больше времени.

В процессорах Intel Core использовались два отдельных буфера — для команд и для данных, которые можно рассматривать как буферы первого уровня. В процессорах Nehalem дополнительно введен унифицированный буфер

ассоциативной трансляции для данных и команд, который стал буфером второго уровня. Он рассчитан на 512 записей, каждая из которых хранит адрес страницы объемом 4 Кбайт (малая страница). На первом уровне остались два буфера — для команд и для данных. Первый рассчитан на 128 записей адресов малых страниц и семь записей адресов больших страниц, объемом 2 Мбайт/4 Мбайт, второй — на 64 записи адресов малых страниц и 32 записи адресов больших страниц.

5.3.2. Перспективы развития многоядерных процессоров Intel

В 2006 г. фирма *Intel* заявила о концепции проектирования процессоров, в соответствии с которой каждые два года будет выполняться переход на обновленную архитектуру, которая на следующий год реализуется уже на основе новой более низкой технологической нормы.

Так, в 2006 г. были представлены процессоры Intel Core, реализованные по технологии 65 нм. В течение целого года выпускалась линейка процессоров Intel Core самого различного назначения: для персональных компьютеров, ноутбуков и серверов. Они отличались производительностью, объемами кэш-памяти, число транзисторов в них колебалось от 100 до 820 млн. В следующем 2007 г. микроархитектура Intel Core была уже реализована по технологии 45 нм.

Первый процессор, построенный на основе микроархитектуры Nehalem, был представлен 16 ноября 2008 г. Он был реализован на базе технологии 45 нм. В течение 2009 г. компания *Intel* выпускала целую линейку процессоров, отличающихся числом ядер, назначением, производительностью и, разумеется, ценой. Все они были реализованы на базе той же технологической нормы. Среди этой массы процессоров есть процессоры младшего, среднего и высокого уровней, предназначенные для использования в стационарных (настольных) и мобильных системах, серверах.

В самом начале 2010 г. Intel представила свои первые процессоры, использующие микроархитектуру Nehalem, выполненные на основе новой технологической нормы — 32 нм. Они получили название Westmere или Nehalem-C.

Следующим поколением процессоров фирмы *Intel* стала серия процессоров с микроархитектурой Sandy Bridge. Первые процессоры этой серии, появившиеся начале 2011 г., были реализованы на базе технологии 32 нм. Микроар-

хитектура Sandy Bridge стала эволюционным развитием микроархитектуры Nehalem. Здесь, так же как и у Nehalem, используется трехуровневая кэш-память, в которой третий уровень является общим для всех ядер, но имеет больший объем. В этих процессорах мощное интегрированное с процессором графическое ядро, новый контроллер системной шины, а сама шина — кольцевая и более производительная, специализированный блок для выполнения операций с ПТ. Реализованы новые команды для векторной обработки данных. Существенно снижено энергопотребление. В 2012 г. стали выпускаться процессоры микроархитектуры Sandy Bridge на основе технологии 22 нм. Она получила название Ivy Bridge.

Начиная с 2013 г., фирма *Intel* планирует перейти к выпуску процессоров на основе новой микроархитектуры Haswell. Эти процессоры будут реализованы на основе технологии 22 нм. Планируется коренная перестройка микроархитектуры, снижение энергопотребления, наличие сопроцессора для векторных вычислений и кэша 4 уровня. Повышение производительности процессоров будет как за счет архитектурных решений, так и за счет введения новых команд векторной обработки данных, а также более эффективной работы с памятью. Следующими за Haswell будут процессоры Broadwell, которые будут иметь ту же микроархитектуру, но реализованы будут на основе более низкой технологической нормы.

Такой ступенчатый процесс развития должен привести к переходу к 2022 г. на технологический процесс 4 нм. Необходимо отметить, что выпуск микросхем по нормам 16 и даже 11 нм теоретически возможен уже сейчас. Однако переход на еще более тонкий 8 нм, а тем более 4 нм технологический процесс возможен лишь при реализации совершенно новых технологических решений.

Контрольные вопросы и задания

1. Что такое суперконвейерная структура процессора? Суперскалярная структура?
2. В чем разница между CISC-процессором и RISC-процессором? В чем разница между командой и микрооперацией?
3. Какую задачу решает дешифратор команд в микроархитектурах, начиная с P6?
4. В чем состоит принцип предсказания переходов?

Содержание

5. Каким образом в процессоре реализуется процесс исполнения команд с изменением последовательности?

6. Какие устройства обеспечивают изменение последовательности выполнения микроопераций?

7. В чем отличие блока SIMD FPU от узлов с обычной арифметикой с плавающей точкой?

8. Опишите порядок выполнения команды в процессоре с микроархитектурой P6.

9. Что происходит, если предсказанная последовательность команд оказалась неверной?

10. Объясните процесс формирования трасс в микропроцессоре NetBurst (см. рис. 5.3).

11. Какую роль играет планировщик команд в микропроцессоре NetBurst?

12. В чем отличие микроархитектуры Pentium Pro от NetBurst?

13. Как работает система повторного исполнения микроопераций в микропроцессоре NetBurst?

14. Объясните суть технологии Hyper Threading.

15. Что общего у микроархитектуры Nehalem с P6?

16. В чем заключается технология обнаружения циклов программ в микроархитектуре Nehalem?

17. В чем суть технологий слияния команд и микроопераций?

18. За счет чего процессоры микроархитектуры Nehalem обеспечили снижение энергопотребления?

19. За счет чего многоуровневая кэш-память обеспечивает повышение производительности процессора?

20. Сравните состав исполнительных блоков в процессорах P6 и Nehalem.

21. Каковы перспективы развития процессоров Intel?

22. Какая из микроархитектур, P6 или Netburst, по вашему мнению, внесла наибольший вклад в развитие микропроцессорной техники? Обоснуйте ответ.

Содержание

Заключение

В настоящее время во всех сферах человеческой деятельности в той или иной степени используются достижения полупроводниковой электроники. Это, прежде всего, промышленное производство, в котором электроника выполняет очень важные функции — управления, координации работы всех технологических объектов. Непосредственное управление работой станков, роботов, автоматических транспортных и складских систем, технологических установок — все это выполняется современными средствами автоматизации технологических процессов, выполненных на базе полупроводниковых электронных элементов, устройств и целых систем.

В основе всех современных информационных технологий также лежат достижения полупроводниковой электроники: компьютерная техника, информационно-вычислительные сети, спутниковые каналы связи и многое другое. В информационных технологиях очень важную роль играют средства оптоэлектроники, рассмотрение которых не входило в задачи этого учебника. Передача огромных объемов информации обеспечивается благодаря применению волоконно-оптических линий связи. Для взаимодействия человека и информационно-вычислительной системы все шире применяются плоские цветные жидкокристаллические мониторы, позволяющие не только отображать информацию, но и вводить ее непосредственно с экрана с помощью пленочной клавиатуры, расположенной на поверхности экрана. На их основе выполняются ноутбуки, промышленные компьютеры и компьютеры специального назначения, конструктивно представляющие собой моноблок.

Основой элементной базы для разработки самых различных электронных устройств и систем становятся БИС/СБИС микропроцессорных комплектов и программируемые ИС, среди которых наиболее перспективными представляются ПЛИС «система на кристалле». Совершенствуется технология СБИС, повышается степень инте-

Содержание

грации, тактовая частота и производительность СБИС. В области микропроцессорных средств продолжает действовать закон, сформулированный одним из основателей фирмы *Intel* Гордоном Муром еще в 1965 г.: «Мощность процессора удваивается каждые полтора года при сохранении его стоимости».

Возрастает роль аналого-цифровых и цифроаналоговых преобразователей, обеспечивающих взаимодействие вычислительной сети с объектом управления. Они могут выполняться как в виде отдельных БИС/СБИС, так и встроенными в СБИС по аналоговым входам и выходам. Такие СБИС получили название аналоговых микропроцессоров.

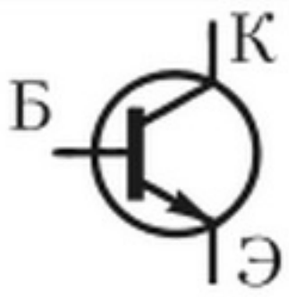
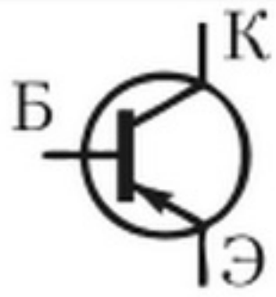
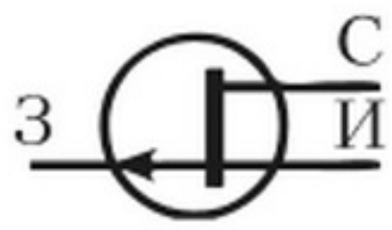
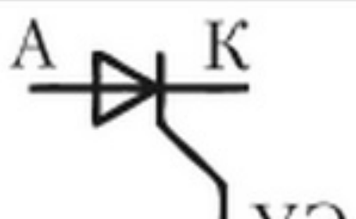
Главным направлением развития средств силовой электроники становятся импульсные преобразователи постоянных напряжений, реализованные в интегральном исполнении, обеспечивающие на основе сети постоянного тока формирование гальванически развязанных питающих напряжений конкретных устройств. Частота преобразования их достигает 500 кГц, КПД — до 90%, а удельная мощность — свыше 5000 Вт/дм^3.

Содержание

Приложения

Приложение 1. Примеры условных обозначений некоторых активных и пассивных элементов на принципиальных электрических схемах

Таблица П1

№ п/п	Наименование элемента	Условное обозначение	Примечание
1	Биполярный транзистор *n-p-n*-типа		Б – база, К – коллектор, Э – эмиттер
2	Биполярный транзистор *p-n-p*-типа		
3	Полевой транзистор с управляющим *p-n*-переходом с каналом *n*-типа		З – затвор, С – сток, И – исток
4	Полевой транзистор с управляющим *p-n*-переходом с каналом *p*-типа		
5	Полевой транзистор с изолированным затвором и встроенным каналом *n*-типа		П – подложка
6	Полевой транзистор с изолированным затвором и индуцированным каналом *n*-типа		
7	Тиристор		А – анод, К – катод, УЭ – управляющий электрод

Содержание

Окончание табл. П1

№ п/п	Наименование элемента	Условное обозначение	Примечание
8	Резистор		
9	Конденсатор		В обозначениях электролитических конденсаторов обкладка, подключаемая к положительному потенциалу, помечается знаком «+»
10	Дроссель		
11	Выпрямительный диод		А – анод, К – катод
12	Стабилитрон		Полярность рабочего напряжения у стабилитрона обратна полярности выпрямительного диода

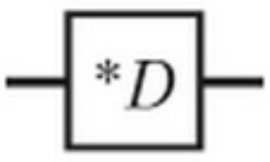
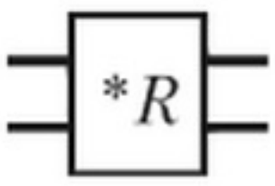
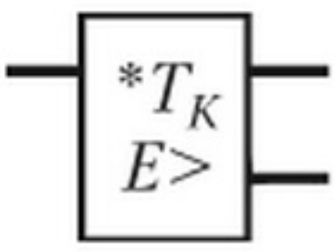
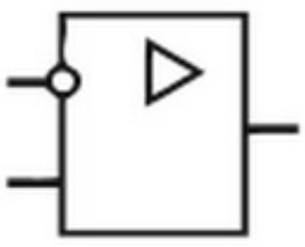
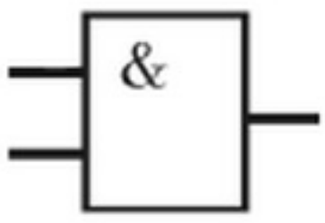
По направлению стрелки в транзисторах определяется их тип, так как она всегда направлена от *p*-области к *n*-области. Так, например, если в биполярном транзисторе стрелка направлена от базы к эмиттеру (строка 1), значит, база — *p*-типа, а транзистор — *n*-*p*-*n*-типа. Стрелка в обозначениях полевых транзисторов с управляющим *p*-*n*-переходом расположена между затвором и каналом транзистора, связывающего исток и сток. Следовательно, обозначение в строке 3 показывает, что затвор у транзистора — *p*-типа, а канал — *n*-типа. Обозначение в строке 4, напротив, соответствует полевому транзистору с каналом *p*-типа. В обозначениях

Содержание

полевых транзисторов с изолированным затвором стрелка располагается между подложкой и каналом. Поэтому обозначения в строках 5, 6 соответствуют транзисторам с подложкой *p*-типа и каналом *n*-типа.

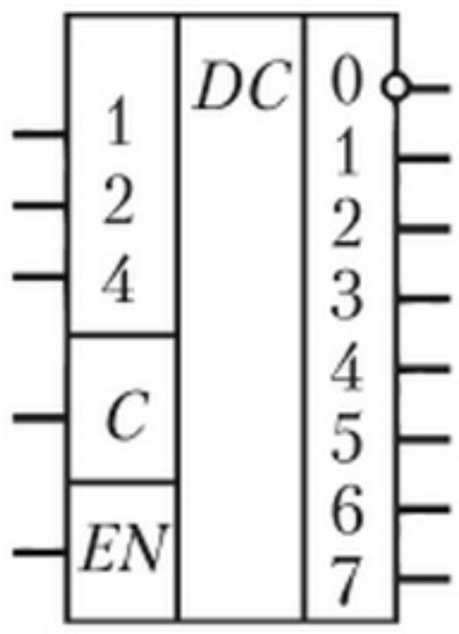
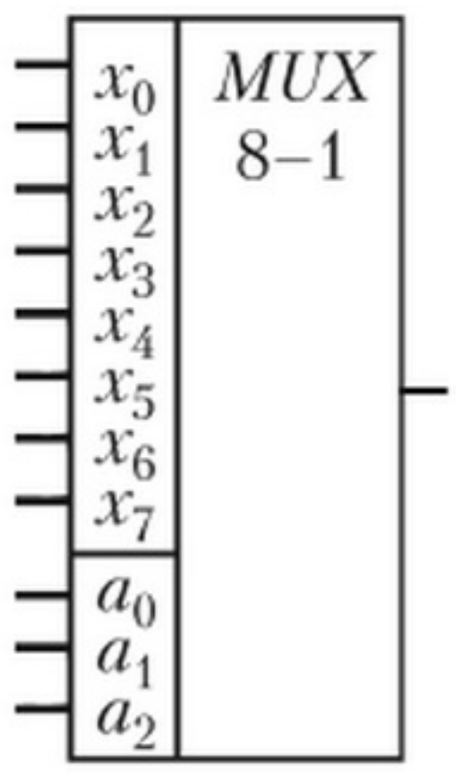
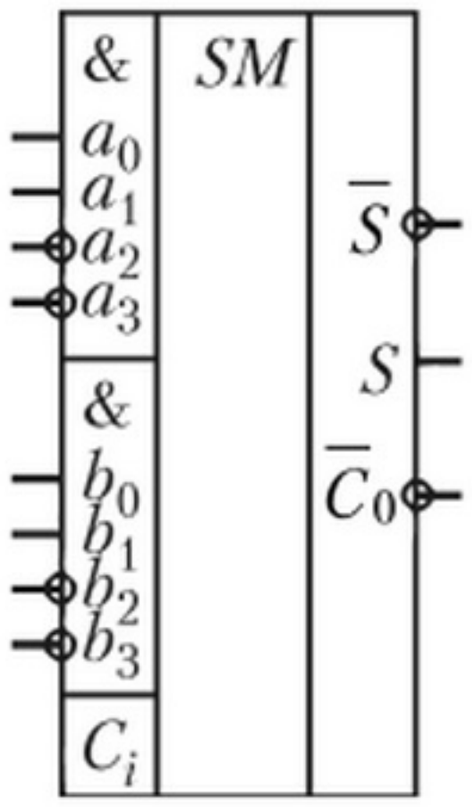
Приложение 2. Примеры условных обозначений некоторых аналоговых и цифровых интегральных элементов на принципиальных электрических схемах

Таблица П2

№ п/п	Наименование элемента	Условное обозначение	Примечание
1	Диод из диодной сборки		Знак «*» — признак сборки элементов, выводы элементов нумеруются в соответствии с номерами выводов корпуса ИС
2	Два резистора в резисторной сборке		Число выводов может не соответствовать удвоенному числу резисторов, если в сборке есть внутренние соединения между резисторами
3	Транзистор из транзисторной сборки		Буквами К и Е обозначаются коллектор и эмиттер; знак «>» указывает направление стрелки база—эмиттер
4	Операционный усилитель		Обозначение ○ соответствует инвертирующему входу; символ ▷ — усилительному устройству
5	Логический элемент «2 И» Логический элемент «3 ИЛИ-НЕ»		Для обозначения инверсных входов/выходов может использоваться не только вариант ○, но и другие, например ⊖ или ⊳

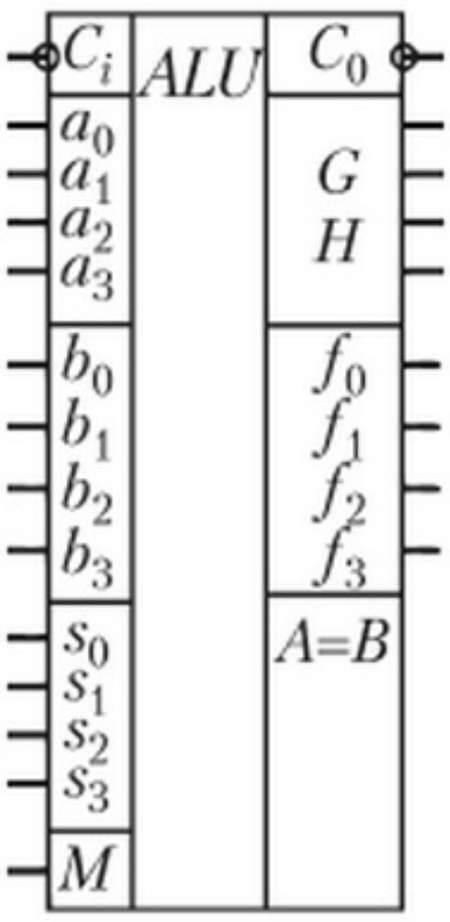
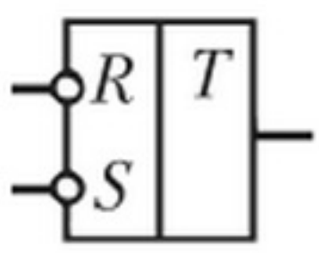
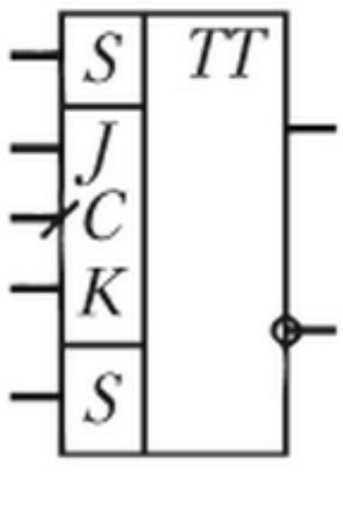
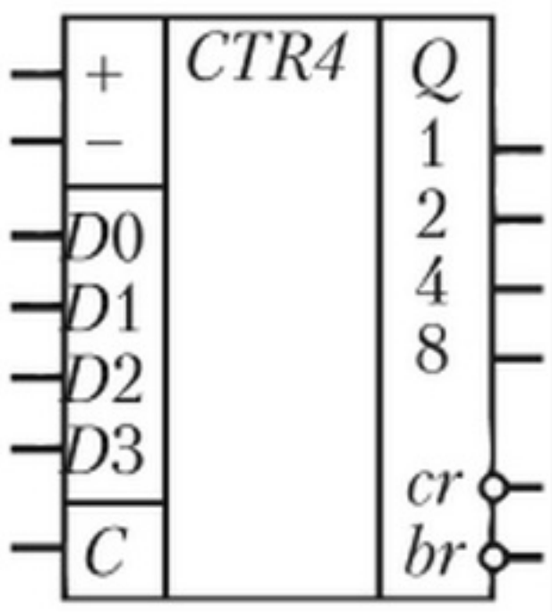
Содержание

Продолжение табл. П2

№ п/п	Наименование элемента	Условное обозначение	Примечание
6	Дешифратор		В интегральном исполнении дешифраторы часто выполняют с инверсными выходами; *C* — вход синхронизации; *EN* — вход разрешения работы (*enable*)
7	8-входовой мультиплексор		x_0, ..., x_7 — информационные входы; a_0, a_1, a_2 — адресные входы
8	Одноразрядный сумматор		Особенность сумматоров, выполненных в виде СИС, — наличие логической обработки входных сигналов. В данном примере для формирования единичных значений слагаемых необходимо наличие единичных сигналов на входах 0, 1 и нулевых на входах 2, 3. Сигнал суммы формируется в прямом *S* и инверсном $\overline{S}$ видах. C_i и $\overline{C}_0$ — сигналы переноса (*carry*)

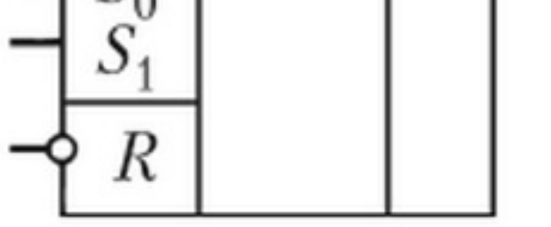
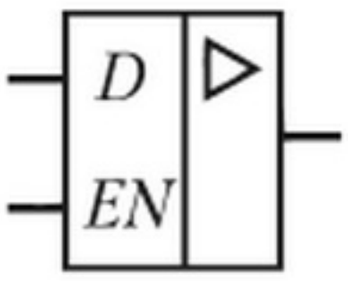
Содержание

Продолжение табл. П2

№ п/п	Наименование элемента	Условное обозначение	Примечание
9	4-разрядное арифметико-логическое устройство		a_0, ..., a_3, b_0, ..., b_3 — операнды; S_0, ..., S_1 — код выполняемой операции; M — признак арифметической или логической операции; f_0, ..., f_3 — результат операции; $A = B$ — признак равенства операндов; G и H — сигналы для организации параллельных переносов при наращивании разрядности ALU
10	Асинхронный *RS*-триггер		R и S — входы асинхронной установки в нуль и единицу
11	Универсальный двухтактный *JK*-триггер		Синхронизация осуществляется по переднему фронту сигнала C
12	4-разрядный двоичный счетчик		«+», «−» — входы прямого и обратного счета; $D0$, ..., $D3$ — входы начальной установки; *cr*, *br* — сигналы переноса и заема (*carry* и *borrow*)

Содержание

Окончание табл. П2

№ п/п	Наименование элемента	Условное обозначение	Примечание
13	Универсальный 8-разрядный реверсивный регистр		$D(0\div7)$ — входы приема информации; *DSL*, *DSR* — сигналы сдвига влево и вправо; *R* — сигнал асинхронной установки в нуль всех разрядов регистра. S_0, S_1 — сигналы управления: $S_0S_1 = 11$ — запись в регистр информации по входам *D*; $S_0S_1 = 01$ — сдвиг влево; $S_0S_1 = 10$ — сдвиг вправо; $S_0S_1 = 00$ — входы регистра недоступны
14	Элемент с мощным цифровым выходом		*D* — информационный вход; ▷ — признак мощного выхода

В обозначениях входов и выходов используются только латинские буквы, соответствующие, как правило, первым буквам английских слов, передающих функциональное назначение: *D* — *data* (данные); *EN* — *enable* (разрешение работы); *R* — *reset* (сброс); *S* — *set* (установка) и т.д.

Приложение 3. Ряды предпочтительных номиналов резисторов и конденсаторов

Таблица П3

E24	E12	E6	E3
Допуск ±5%	Допуск ±10%	Допуск ±20%	Допуск >±20%
1,0	1,0	1,0	1,0
1,1			
1,2	1,2		
1,3			

Окончание табл. П3

E24	E12	E6	E3
Допуск ±5%	Допуск ±10%	Допуск ±20%	Допуск >±20%
1,5	1,5	1,5	
1,6			
1,8	1,8		
2,0			
2,2	2,2	2,2	2,2
2,4			
2,7	2,7		
3,0			
3,3	3,3	3,3	
3,6			
3,9	3,9		
4,3			
4,7	4,7	4,7	4,7
5,1			
5,6	5,6		
6,2			
6,8	6,8	6,8	
7,5			
8,2	8,2		
9,1			

Типовые значения номиналов резисторов и конденсаторов определяются в соответствии с рядами номиналов, рекомендованных Международной электротехнической комиссией. Ряды номиналов определяются на основе геометрических рядов чисел, получивших наименования E3, E6, E12 и E24.

Базовым рядом предпочтительных значений является ряд E24, состоящий из округленных значений теоретических чисел $\sqrt[24]{10^n}$, где показатель степени n — целое поло-

Содержание

жительное или отрицательное число (числа, приведенные в таблице, получаются при замене n числом от 0 до 23).

Ряд E12 образуют округленные значения чисел $\sqrt[12]{10^n}$, а получают его из ряда E24 путем исключения четных членов. Аналогично, ряд E6 получают путем исключения четных членов ряда E12, а ряд E3 — путем исключения четных членов ряда E6.

Ряды предпочтительных значений определяют не только номинальные значения резисторов и конденсаторов, но и соответствующие им рекомендуемые допускаемые отклонения. Ряду E24 соответствует допуск ±5%, ряду E12 — ±10%, ряду E6 — ±20%, а ряду E3 — более ±20%.

Для прецизионных измерительных и вычислительных аналоговых устройств выпускаются элементы повышенной точности: с допуском ±2%, ±1%, ±0,5%, ±0,25%, ±0,1%. Для особо точных устройств используются элементы с допусками ±0,05%, ±0,02%, ±0,01% и даже ±0,005%.

Содержание

Предметный указатель

FPU 343, 358
MMX 359
MacroFusion 372
Micro-ops fusion 373
ROM микрокодов 355, 360, 370
SIMD 343
SIMD FPU 343
SSE 343
SSE2 359
SSE3 359
SSE4 378
SSE4.2 379
Адаптер
— параллельный периферийный 291
— программируемый связной 293
Амплитудно-частотная характеристика 64
Аналого-цифровой преобразователь 103
Анод
— диода 21
— тиристора 50
Аппаратные ядра 325
Арифметико-логическое устройство (АЛУ) 186, 236
Архитектура с двойной независимой шиной 345
Асинхронный обмен 288
Ассемблер 253

База транзистора 31
Базовые матричные кристаллы (БМК) 311
Барьерная емкость *p-n*-перехода 19, 28
Бит четности 295
Блок
— ALU 342, 357, 375
— арифметики с плавающей точкой 343

Содержание

— арифметики с плавающей точкой над множественными данными 343
— выборки команд 341
— завершения 344
— интерфейса 235
— интерфейса памяти 344
— предварительной выборки команд 352
— ускоренного переноса 188
— шинного интерфейса 346
Блоки
— ввода/вывода FPGA 322
— обработки чисел с плавающей точкой 358
— потоковых векторных операций над множественными данными 377
Большая интегральная схема (БИС) 52
Булева
— алгебра 141
— функция 144
Буфер
— переупорядочивания микроопераций 342
— переупорядочивания памяти 345
— предсказания переходов 340, 350
— предсказания переходов для кэш МО 355
Быстродействие ЗУ 211

Валентная зона 14
Варикап 28
Вектор прерывания 299
Векторные операции 344
Виртуальная память 270
Вложение прерываний 298
Внешние устройства 232
Внутрисегментный адрес 248
Вольт-амперная характеристика
— диода Шоттки 26
— полупроводникового диода 22
— тиристора 50
Время рассасывания неосновных носителей 157
Входная подсистема 352
Входная характеристика транзистора 37—38
Входное сопротивление 34
Входной буфер предсказания переходов 354

Выпрямитель
— двухполупериодный 108—109
— однополупериодный 108
— тиристорный управляемый 117—119
— трехфазный 110—111
Выходная характеристика транзистора 37—38
Вычислитель адреса следующей команды 339—340

Гальваническая развязка
— оптронная 130
— релейная 131
— трансформаторная 132
Генератор пилообразного напряжения 98—99
Генераторы адресов 357
Гиперпоточность 364

Двигатель постоянного тока 122
Декремент 237, 257
Дешифратор
— команд 236
— неполный 167
— полный 167
Дизъюнкция 141
Динистор — см. тиристор диодный
Диод
— выпрямительный 23
— полупроводниковый 21
— Шоттки (металлополупроводниковый) 26
Директивы ассемблера 254
Диспетчер ресурсов 356
Дифференциальный коэффициент усиления 82

Закон
— поглощения 143
— склеивания 144
Запоминающие устройства (ЗУ)
— адресные 214
— асинхронные 215
— ассоциативные (с ассоциативным доступом) 217
— динамические 215
— конвейерные 215

— масочные 220—221
— оперативные (ОЗУ, RAM, ЗУ с произвольной выборкой) 214
— полупроводниковые 211
— постоянные (ПЗУ, ROM) 214
— постоянные программируемые (ППЗУ, PROM) 215, 221—222
— постоянные репрограммируемые (EPROM, EEPROM) 216, 222—223
— с последовательным доступом 216
— статические 215
— стековые (буферы LIFO) 216
— тактируемые (синхронные) 215
— файловые 216
— флэш 225—226
— циклические 216
Запрещенная зона 14
Запрещенный набор 144
Затвор полевого транзистора 43—44
Захват системных шин 241—242
Зернистость 320
Зона проводимости 14

Инверсия 141
Инвертирующий вход 82
Инвертор 106, 157
Инжекционная электролюминесценция 28
Инициализация 234, 316
Инкремент 237, 257
Интегральная микросхема (ИМС) 52
Интерфейс 286
— параллельный 287
— последовательный 287
Информационная емкость ЗУ 211
Исполнение с изменением последовательности 341
Исток полевого транзистора 43
Источники первичного электропитания 106

Канал полевого транзистора 43
Карта Карно 147
Катод
— диода 21
— тиристора 50

Квитирование 288
КМОП-транзисторы 159
КМОП-элементы 166
Коллектор транзистора 31
Команды перестановки 344
Конвейер 267
  Компаратор аналоговый 100
  Комплементарные транзисторы — см. КМОП-транзисторы
  Конвейерный принцип работы МП 267
  Конденсатор проходной 127
  Конституента единицы 145
  Контактная разность потенциалов 26—27
  Контроллер ПДП 301
  Контроллер прерываний 297
  Концентратор 288
Конъюнктор 151
Конъюнкция 141
Коэффициент
— передачи тока в схеме ОЭ 36
— передачи тока эмиттера 35
— стабилизации 114
— усиления по мощности 34
— усиления по напряжению 34
— усиления по току 34
Кусочно-линейная аппроксимация 96
Кэш 268
— второго уровня 345
— данных первого уровня 345
— команд первого уровня 345
— последовательностей МО 354

Логическая единица 141
Логические блоки FPGA 320
Логический нуль 141

Магистраль 233—234, 288
Магистрально-модульная структура 233—234
Макроячейка 319
Маскирование прерываний 299
Матрица R—2R 102
— запоминающих элементов 217

Матричный умножитель 188—189
Метка команды 254
Микроархитектура 336
— Intel Core 367
— Nehalem 367
— NetBurst 351
— P6 336
— процессора 336—337
Микроконтроллеры 283
Микропроцессор (МП) 232
Микропроцессорная система 232
Микропроцессорный комплект 233
МикроЭВМ 232
Микроэлектроника 52
Минимизация булевых функций 146
Мнемоника команды 254
Многоядерные процессоры 366
Модем 294
Модуль счета 197
Мультивибратор 97
Мультиплексор 175

Набор булевой функции 144
Набор исполнительных блоков 342
Нагрузочная прямая 39
Напряжение
— обратное допустимое 23
— отсечки 46
— смещения нуля операционного усилителя 83
Негативная логика 150
Неинвертирующий вход операционного усилителя 82
Нелинейные искажения усилителя 65

Область насыщения операционного усилителя 83
Обмен по инициативе ВУ с прерыванием работы МП 239—241
Обмотка возбуждения 122
Обратный ток диода 22
Однотранзисторный запоминающий элемент 229
Операнд 236, 254
Оперативное запоминающее устройство (ОЗУ) 214—215
Операционное устройство 243

Оптрон 30
Очередь
— микроопераций 356
— МО над целыми числами и числами с ПТ 356
— МО с памятью 356
Передаточная характеристика 76
— выходного каскада 76
— полевого транзистора 46
Переход
— безусловный 262—264
— условный 262—264
Плавающий затвор 223—224
Планарно-диффузионная технология 54
Планировщик последовательностей микроопераций 341
Планировщики микроопераций 356
ПЛИС 314
— типа CPLD 314, 317
— типа FPGA 314, 319
Подсистема
— исполнения с изменением последовательности 339, 341
— памяти 338—339
— упорядоченного завершения 338—339
— упорядоченной предварительной обработки 338—339
Позитивная логика 150
Полоса частот усилителя 65
Полузаказные БИС/СБИС 311
Полупроводник
— *n*-типа 16—17
— *p*-типа 16—17
Порт 264, 285
Постоянное запоминающее устройство (ПЗУ) 214
Потенциальный барьер 18
Преддешифрация 370
Предсказывание переходов 272, 340
Прерывания — см. обмен по инициативе ВУ с прерыванием работы МП
Примеси
— акцепторные 17
— донорные 16
Примесные полупроводники 16

Содержание

Принцип двойственности 142
Приоритетное кольцо 300
Пробой *p*-*n*-перехода 22
— тепловой 22
— электрический 22
Проводимость
— *n*-типа (электронная) 16
— *p*-типа (дырочная) 16
Программируемые логические интегральные схемы (ПЛИС) 314
Программируемые логические матрицы 304—305
Программируемые схемы с матричной структурой 304
Программно-управляемый обмен 239
Программный счетчик 236—238
Процедура out-of-order 348
Процессоры
— суперконвейерные 267
— суперскалярные 267
Прямой доступ к памяти 241
Прямой ток диода 22—23
Пул команд 342

Распределитель 174
Регистр 203
— команд 236
— однофазный 204
— параллельный 204
— парафазный 204
— сдвигающий 205
— универсальный 206
— флажков 236
Регистровые файлы 204
Регистры 236
— адреса 236
— буферные 290
— данных 236
— сегментные 245
Режим
— максимальной конфигурации 249
— минимальной конфигурации 249
— насыщения 39
— отсечки 40

— работы транзистора активный 40
— работы транзистора по переменному току 69
— работы транзистора по постоянному току 68
— работы транзистора статический 39
Режимы адресации 250
— базовый индексный 250—253
— внутрисегментный косвенный 250—253
— внутрисегментный прямой 250—253
— межсегментный косвенный 250—253
— межсегментный прямой 250—253
— непосредственный 250—253
— относительный базовый индексный 250—253
— прямой 250—253
— регистровый 250—253
— регистровый косвенный 250—253
— регистровый относительный 250—253
Ряд предпочтительных значений номиналов 390—391

СБИС «система в кристалле» 325
Сверхбольшая интегральная схема (СБИС) 52
Светодиод 28
Сглаживающие фильтры 111—113
Сдвиг
— арифметический 259—261
— логический 259—261
— циклический (ротация) 259—261
Сегментация памяти 246—248
Семисегментные светодиодные матрицы 29
Синфазный сигнал 75
Синхронный обмен 287—288
Синхросимвол 294
Система повторного исполнения микроопераций 362—363
Системы межсоединений FPGA 323—324
Собственный коэффициент усиления 83
Совершенная дизъюнктивная нормальная форма (СовДНФ) 146
Сопроцессор 267—268
Способ
— параллельного кодирования 105
— пилообразного напряжения 104
— последовательного приближения 103

Стабилизатор напряжения 113
— импульсный 115—117
— компенсационный 113
— линейный 113—115
— параметрический 113
Стабилитрон 24
Станция-резервуар 342
Старт-бит 295
Стек 240
Сток полевого транзистора 43
Стокозатворная характеристика 46
Стоп-бит 295
Структура БМК 312
— бесканальная 312
— блочная 313
— канальная 312
Структура ЗУ типа 2*D* 217—218
Структура ЗУ типа 2*DM* 218—219
Сумма по модулю 2 176
Сумматор 180
— групповой с цепным переносом 185
— одноразрядный 180—182
— параллельный 183
— последовательный 182
— с групповым переносом 185
— с параллельным переносом 184
— с последовательным переносом 183—184
Схема
— вычитания 89—90
— дифференцирования 90—91
— интегрирования 90—91
— суммирования 88—89
Схема включения транзистора
— с общей базой 33—35
— с общим коллектором 40—41
— с общим эмиттером 35—37
Счетчик 197
— реверсивный 200
— синхронный с параллельным переносом 201
— суммирующий асинхронный 199

Таблица истинности булевой функции 144
Таблица назначения регистров 342

Таймер программируемый интервальный 302—304
Теговый адрес 217
Температурный дрейф напряжения смещения нуля 83
Теорема де Моргана 144
Терм 306
Термостабилизация рабочей точки 70
Тетрада 259
Технология
— Hyper Threading 364
— SSE 343
— обнаружения циклов программ 372
— слияния команд 372
— слияния микроопераций 373
Тиристор 50
— диодный (динистор) 50
— управляемый (триодный) 51—52
Транзистор 31
— биполярный 31
— биполярный *n-p-n*-типа 31—32
— биполярный *p-n-p*-типа 31—32
— ЛИЗМОП 223
— МДП 46
— многоэмиттерный 160—161
— МНОП 223
— МОП 46
— полевой 43
— полевой с изолированным затвором 46
— полевой с индуцированным каналом 48
— полевой с управляющим *p-n*-переходом 44—45
— полевой со встроенным каналом 46—47
— Шоттки 158
Трансляция 254
Триггер 189
— асинхронный 190
— двухступенчатый 197
— задержки (*D*-триггер) 193—194
— одноступенчатый 196—197
— с динамическими входами 196
— с установочными входами (*RS*-триггер) 190—191
— синхронный 190
— со статическими входами 190
— со счетным входом (*T*-триггер) 194—195
— универсальный (*JK*-триггер) 195—196

Содержание

Триггерный запоминающий элемент 228
ТТЛ-логика 160
ТТЛШ-логика 162

Указатель стека 240
Унитарный код 167
Управляющий электрод тиристора 51
Усилитель 62
— избирательный (полосовой) 64
— логарифмический 94—95
— низкой частоты 64
— операционный 81
— постоянного тока 64
— широкополосный 64
Усилительные каскады
— выходные двухтактные 79—80
— выходные однотактные 79
— дифференциальные 74
— низких частот на биполярных транзисторах 66—67
— низких частот на полевых транзисторах 71—72
Устройство
— аналоговое 61
— цифровое 61—62
Устройство подавления выбросов 127—128
— провалов 127
Устройство управления 236
Устройство управления шинами 236

Файловая флэш-память 226—227
Флажки 237
Фотодиод 29
Фотоносители 30
Фототок 30
Функциональность 320
Функция переходов 152

Целочисленный регистровый файл и логика 357
Центральный процессорный элемент 284
Цифроаналоговый преобразователь 101
Цифровое устройство
— асинхронное 153
— комбинационное 151—152

— последовательностное 151—152
— синхронное 153

Чип 52
Чип-сет 369

Шина 235
— QPI 369
— адреса 235
— данных 235
— управления 235
Шинный интерфейс 243
Шинный формирователь 290
Широтно-импульсная модуляция (ШИМ) 115
Шифратор 172
— приоритетный 173

Экранирование 128
Экспоненциальный генератор 95
Электромагнитная совместимость 125
Эмиттер транзистора 31
Эмиттерный повторитель 41
Энергонезависимость ЗУ 213
Эпитаксиальное наращивание 20
Эпитаксиально-планарная технология 54

Ядро процессора 336
Якорь 122

Содержание

Литература

1. Haswell // Материал из Википедии — свободной энциклопедии. URL: http://ru.wikipedia.org/wiki/Haswell.
2. Nehalem // Материал из Википедии — свободной энциклопедии. URL: http://ru.wikipedia.org/wiki/Nehalem.
3. Sandy_Bridge // Материал из Википедии — свободной энциклопедии. URL: http://ru.wikipedia.org/wiki/Sandy_Bridge.
4. *Бессонов, О.* Pentium 4: Мистический и загадочный Trace-кэш / Сайт компании «Ф-Центр». URL: http://www.fcenter.ru/online.shtml?articles/hardware/processors/13736.
5. *Волков, Н. И.* Электромашинные устройства автоматики / Н. И. Волков, В. П. Миловзоров. — М. : Высшая школа, 1986.
6. *Волков, Ю. С.* Электротехника и электроника / Ю. С. Волков, И.В Гончаров. — М. : Изд-во ВЗПИ, 1992.
7. ГОСТ 28884—90 (МЭК 63—63) Ряды предпочтительных значений для резисторов и конденсаторов.
8. *Гусев, В. Г.* Электроника и микропроцессорная техника : учебник для вузов / В. Г. Гусев, Ю. М. Гусев. — М. : Высшая школа, 2005. — 4-е изд. — М. : Мир, 1993.
9. *Колесниченко, О. В.* Аппаратные средства PC / О. В. Колесниченко, И. В. Шишигин. — 4-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2000.
10. *Лукоев, О.* Многоядерные перспективы корпорации Intel / О. Лукоев // КомпьютерПресс. — 2005. — № 3.
11. *Лю, Ю-Ч.* Микропроцессоры семейства 8086/8088: пер. с англ. / Ю-Ч. Лю, Г. Гибсон. — М. : Радио и связь, 1987.
12. *Любомудров, А. А.* Электромагнитные помехи и помехозащищенность радиоэлектронной аппаратуры. — М. : Изд-во Военной академии имени Ф. Э. Дзержинского, 1996.
13. Микроархитектура процессоров поколения P6 // Информационная система по процессорам Intel Celeron и Pentium-3. URL: http://psv-cmos.narod.ru/p6.htm.

Содержание

14. Микропроцессоры и микропроцессорные комплекты интегральных микросхем. Справочник : в 2 т. / Б. Б. Абрайтис [и др.] / под ред. В. А. Шахнова. — М. : Радио и связь, 1988.

15. *Миловзоров, В. П.* Элементы информационных систем / В. П. Миловзоров. — М. : Высшая школа, 1989.

16. Обзор микроархитектур современных десктопных процессоров. Часть 1. Часть 2. Часть 3 // Сайт Cyberguru.ru. URL: http://www.cyberguru.ru/hardware/processors/cpu-architecture-1.html; http://www.cyberguru.ru/hardware/processors/cpu-architecture-2.html; http://www.cyberguru.ru/hardware/processors/cpu-architecture-3.html.

17. *Панков, И. Г.* Электротехника, электроника и микропроцессорная техника : конспект лекций / И. Г. Панков. — М. : Издательство МГОУ, 1995.

18. *Пахомов, С. О.* Процессорная микроархитектура Intel следующего поколения / С. О. Пахомов // КомпьютерПресс. — 2008. — № 8.

19. *Полувялов, А.* Willamette — как будет работать новоиспеченный флагман от Intel // Сайт IXBT.com. URL: http://www.ixbt.com/cpu/pentium4—2.html.

20. Система сквозного проектирования Altium Designer // Сайт ЗАО «НПП «Родник». URL: http://www.rodnik.ru/product/sapr/pp_i_plis/altium/altium_designer_soft_vhdl.

21. *Титце, У.* Полупроводниковая схемотехника : учебник-справочник-энциклопедия: пер. с нем. / У. Титце, К. Шенк. —12-е изд. — М. : ДМК Пресс, 2008.

22. *Угрюмов, Е. П.* Цифровая схемотехника. — 2-е изд., доп. — СПб. : БВХ-Петербург, 2007.

23. *Федорков, Б. Г.* Микросхемы ЦАП и АЦП: функционирование, параметры, применение / Б. Г. Федорков, В. А. Телец. — М. : Энергоатомиздат, 1990.

24. *Хоровиц, П.* Искусство схемотехники: в 3 т.: пер. с англ. / П. Хоровиц, У. Хилл. — 4-е изд. — М. : Мир, 1993.

25. *Чернов, Е. А.* Электропривод и электрооборудование автоматизированного производства / Е. А. Чернов. — М. : Машиностроение, 1992.

Содержание

Наши книги можно приобрести:

Учебным заведениям и библиотекам:
в отделе по работе с вузами
тел.: (495) 744-00-12, e-mail: vuz@urait.ru

Частным лицам:
список магазинов смотрите на сайте urait.ru
в разделе «Частным лицам»

Магазинам и корпоративным клиентам:
в отделе продаж
тел.: (495) 744-00-12, e-mail: sales@urait.ru

Отзывы об издании
присылайте в редакцию
e-mail: red@urait.ru

Новые издания и дополнительные материалы доступны
в электронной библиотечной системе «Юрайт»
biblio-online.ru

Учебное издание

Миловзоров Олег Владимирович,
Панков Иван Григорьевич

ОСНОВЫ ЭЛЕКТРОНИКИ

Учебник для СПО

Формат 84×108 $^{1}/_{32}$.
Гарнитура «PetersburgC». Печать цифровая.
Усл. печ. л. 21,37. Тираж 1000 экз. Заказ № 3387, 12 экз.

ООО «Издательство Юрайт»
111123, г. Москва, ул. Плеханова, д. 4а.
Тел.: (495) 744-00-12. E-mail: izdat@urait.ru, www.urait.ru