МИР электроники

К. БОЙТ

Цифровая электроника

Перевод с немецкого
М.М. Ташлицкого

ТЕХНОСФЕРА
Москва
2007

Бойт К.
Цифровая электроника
Москва:
Техносфера, 2007. – 472 с. ISBN 978-5-94836-124-6

В книге шаг за шагом раскрывается мир цифровой электроники, ее взаимосвязь с другими областями техники. При создании учебника автор использовал свой большой опыт преподавательской деятельности. Материал изложен последовательно и понятно, начиная с основ.

Контрольные тесты с вопросами и задачами в конце каждой главы позволяют проверить понимание пройденного материала, в конце учебника приведены решения.

Книга предназначена для студентов электротехнических и машиностроительных специальностей, инженеров-практиков, техников, а также всех, кто интересуется современной цифровой техникой.

© Vogel Industrie Medien GmbH & Co KG, Würzburg (Germany)
© 2007, ЗАО "РИЦ "Техносфера", перевод на русский язык, оригинал-макет, оформление

ISBN 978-5-94836-124-6
ISBN 3-8023-1958-3 (нем.)

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ 9

Глава 1. ОСНОВНЫЕ ПОНЯТИЯ 10

1.1. Аналоговое и цифровое представление величин 10
1.1.1. Аналоговое представление величин 10
1.1.2. Цифровое представление величин 12
1.2. Бинарные и логические состояния 13
Контрольный тест 15

Глава 2. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ 17

2.1. Основные законы и элементы алгебры логики 17
2.1.1. Логический элемент И и операция логического умножения (конъюнкции) 17
2.1.2. Логический элемент ИЛИ и операция логического сложения (дизъюнкции) 18
2.1.3. Логический элемент НЕ и операция инверсии (отрицания) 19
2.1.4. Основные логические элементы 20
2.2. Комбинированные элементы 20
2.2.1. Логический элемент И-НЕ 20
2.2.2. Логический элемент ИЛИ-НЕ 21
2.2.3. Логический элемент эквивалентности 22
2.2.4. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR) 23
2.2.5. Комбинации элементов с двумя входами 24
2.3. Логические элементы с тремя и более входами 25
Контрольный тест 26

Глава 3. АНАЛИЗ СХЕМ 28

3.1. Таблица истинности и цифровая схема 28
3.1.1. Таблица истинности цифровой схемы с двумя входами 28
3.1.2. Таблица истинности цифровой схемы с тремя входами 29
3.2. Логические функции и цифровые схемы 31
3.2.1. Определение логической функции цифровой схемы 31
3.2.2. Синтез схемы по заданной логической функции 33
3.3. Требуемая функция и реальная функция 33
3.3.1. Как определить реальную логическую функцию 34
3.3.2. Поиск дефекта схемы 35
Контрольный тест 36

Глава 4. АЛГЕБРА ЛОГИКИ 38

4.1. Переменные и постоянные величины (константы) 38
4.2. Законы алгебры логики 39
4.3. Аксиомы и тождества алгебры логики 40
4.3.1. Аксиомы 40
4.3.2. Законы коммутативности и ассоциативности 40
4.3.3. Дистрибутивный закон 42
4.3.4. Теоремы де Моргана 44
4.3.5. Приоритеты логических операций 45
4.4. Функции И-НЕ и ИЛИ-НЕ 46
4.5. Примеры 52
Контрольный тест 55

Глава 5. СИНТЕЗ СХЕМ 57

5.1. Синтез схем на логических элементах по заданным условиям 57
5.2. Нормальные формы записи 59
5.2.1. Нормальная форма операции логического сложения ИЛИ 59

5.2.2. Нормальная форма операции логического умножения И ... 62
5.3. Упрощение и преобразование нормальной формы ИЛИ с помощью алгебры логики ... 63
5.3.1. Упрощение нормальной формы ИЛИ ... 63
5.3.2. Преобразование нормальной формы ИЛИ ... 64
5.4. Метод карт Карно ... 65
5.4.1. Карта Карно для двух переменных ... 65
5.4.2. Карта Карно для трех переменных ... 70
5.4.3. Карта Карно для четырех переменных ... 73
5.4.4. Карта Карно для пяти переменных ... 76
5.4.5. Карта Карно для более чем пяти переменных ... 79
5.5. Расчет логических схем ... 80
5.5.1. Общие указания ... 80
5.5.2. Цифровая схема включения и выключения из нескольких мест ... 81
5.5.3. Переключатель два из трех ... 82
5.5.4. Схема контроля четности ... 83
5.5.5. Пороговая логическая схема ... 85
5.5.6. Схема сравнения (компаратор) ... 86
5.5.7. Схема сортировки транзисторов ... 87
5.6. Задания по схемотехническому проектированию ... 88
5.6.1. Схема управления ... 88
5.6.2. Схема контроля нечетности ... 89
5.6.3. Мажоритарная схема ... 89
5.6.4. Схема блокировки ... 89
5.6.5. Анализатор показаний радаров ... 90
Контрольный тест ... 90

Глава 6. СЕМЕЙСТВА СХЕМ ... 92

6.1. Общие сведения ... 92
6.2. Бинарные уровни напряжения ... 93
6.3. Положительная и отрицательная логика ... 95
6.4. Свойства схем ... 97
6.4.1. Потребляемая мощность ... 97
6.4.2. Диапазон уровней и передаточная характеристика ... 97
6.4.3. Время переключения ... 99
6.4.4. Нагрузочная способность ... 100
6.4.5. Помехоустойчивость ... 101
6.4.6. Проводные логические операции ... 102
6.5. ДТЛ-схемы ... 104
6.5.1. Введение ... 104
6.5.2. Основные ДТЛ-схемы ... 104
6.5.3. МПЛ-схемы ... 107
6.6. ТТЛ-схемы ... 111
6.6.1. Строение и принцип действия ТТЛ-элементов ... 111
6.6.2. Стандартные ТТЛ-схемы ... 116
6.6.3. ТТЛ с пониженным энергопотреблением (Low-Power-TTL, LTTL) ... 128
6.6.4. Высокоскоростные ТТЛ (High-Speed-TTL, HTTL) ... 128
6.6.5. Шотки-ТТЛ (ТТЛШ) ... 129
6.6.6. ТТЛШ с пониженным энергопотреблением (Low-Power-ТТЛШ) ... 130
6.6.7. Сравнительная оценка логических элементов ... 131
6.7. Эмиттерно-связанная логика (ЭСЛ) ... 133
6.8. Логические элементы на МОП-транзисторах ... 137
6.8.1. Опасность статического электричества ... 137
6.8.2. Логические элементы на *p*-канальных МОП-транзисторах (*p*МОП) ... 137
6.8.3. Логические элементы на *n*-канальных МОП-транзисторах (*n*МОП) ... 140
6.8.4. Логические элементы на КМОП-транзисторах ... 141
Контрольный тест ... 149

Глава 7. БИНАРНЫЕ СХЕМЫ С ВРЕМЕННОЙ ЗАВИСИМОСТЬЮ 151
7.1. Введение 151
7.2. Классификация триггеров 154
7.3. Нетактируемые триггеры 157
7.3.1. Триггер на элементах ИЛИ-НЕ 157
7.3.2. Триггер на элементах И-НЕ 158
7.4. Тактируемые триггеры 159
7.4.1. *RS*-триггеры 159
7.4.2. *RS*-триггеры с доминирующим *R*-входом 161
7.4.3. *E*-триггер 162
7.4.4. *D*-триггер 163
7.4.5. Таблицы данных 163
7.5. Триггеры, управляемые по фронту синхроимпульса 167
7.5.1. Импульсные элементы 168
7.5.2. *RS*-триггеры, управляемые по одному фронту 169
7.5.3. *T*-триггеры, управляемые по одному фронту 171
7.5.4. *JK*-триггеры, управляемые по одному фронту 173
7.5.5. *D*-триггеры, управляемые по одному фронту 176
7.5.6. *RS*-триггеры, управляемые по обоим фронтам 179
7.5.7. *JK*-триггеры, управляемые по обоим фронтам 180
7.5.8. Дополнительные триггерные схемы 182
7.6. Временные диаграммы 186
7.7. Характеристические уравнения 189
7.8. Моностабильные ячейки 193
7.9. Элементы задержки 197
Контрольный тест 200

Глава 8. ДВОИЧНЫЕ КОДЫ И СИСТЕМЫ СЧИСЛЕНИЯ 202
8.1. Введение 202
8.2. Двоичная система счисления 202
8.2.1. Структура двоичной системы счисления 202
8.2.2. Перевод двоичных чисел в десятичную систему счисления 203
8.2.3. Перевод десятичных чисел в двоичную систему счисления 204
8.2.4. Вещественные двоичные числа (правильные дроби) 205
8.2.5. Сложение двоичных чисел 206
8.2.6. Вычитание двоичных чисел 207
8.2.7. Отрицательные двоичные числа 211
8.3. Двоично-десятичный код (*BCD*-числа) 213
8.3.1. Представление чисел в двоично-десятичном коде 213
8.3.2. Сложение в *BCD*-формате 214
8.3.3. Вычитание в *BCD*-формате 216
8.4. Другие тетрадные системы счисления 217
8.4.1. Код с избытком три 218
8.4.2. Код Айкена 219
8.4.3. Код Грея 221
8.5. Шестнадцатеричная система счисления (Hexadecimal) 222
8.5.1. Структура шестнадцатеричной системы счисления 222
8.5.2. Перевод шестнадцатеричных чисел в десятичную систему счисления 223
8.5.3. Перевод десятичных чисел в шестнадцатеричную систему счисления 224
8.5.4. Перевод двоичных чисел в шестнадцатеричную систему счисления 225
8.5.5. Перевод шестнадцатеричных чисел в двоичную систему счисления 228
8.6. Восьмеричная система счисления (Octal) 228
8.6.1. Структура восьмеричной системы счисления 228
8.6.2. Преобразование восьмеричных чисел 229
8.7. Коды, распознающие ошибки 231
8.7.1. Понятие избыточности 231

8.7.2. Дополнительный двоичный код ... 232
8.7.3. Код 2 из 5 ... 233
8.7.4. Код 3 из 5 ... 234
8.7.5. Код 2 из 7 ... 235
8.8. Коды, исправляющие ошибки ... 235
8.8.1. Принцип действия ... 235
8.8.2. Код Хемминга ... 236
Контрольный тест ... 239

Глава 9. ПРЕОБРАЗОВАТЕЛИ КОДОВ И УРОВНЕЙ ... 243

9.1. Преобразователи кодов ... 243
9.1.1. Расчет преобразователей кода ... 243
9.1.2. Преобразователи десятичного кода в *BCD*-код ... 245
9.1.3. Преобразователи *BCD*-кода в десятичный код ... 247
9.1.4. Преобразователи десятичного кода в код с избытком 3 ... 250
9.1.5. Преобразователи кода с избытком 3 в десятичный код ... 250
9.1.6. Преобразователи десятичного кода в семисегментный ... 251
9.1.7. Преобразователи *BCD*-кода в семисегментный ... 252
9.2. Преобразователи уровней ... 258
9.2.1. Введение ... 258
9.2.2. Структура преобразователей уровня ... 259
9.2.3. Интегральные преобразователи уровня ... 260
Контрольный тест ... 262

Глава 10. СЧЕТЧИКИ И ДЕЛИТЕЛИ ЧАСТОТЫ ... 263

10.1. Счет и разновидности счетчиков ... 263
10.2. Асинхронные счетчики ... 264
10.2.1. Асинхронные двоичные счетчики ... 264
10.2.2. Асинхронные *BCD*-счетчики (в двоично-десятичном коде) ... 265
10.2.3. Асинхронный декадный счетчик ... 280
10.2.4. Асинхронные счетчики по модулю *n* ... 281
10.2.5. Асинхронные счетчики с произвольным коэффициентом пересчета ... 283
10.2.6. Асинхронные счетчики для кода Айкена ... 284
10.2.7. Асинхронные счетчики для кода с избытком 3 ... 284
10.3. Синхронные счетчики ... 285
10.3.1. Принцип действия ... 285
10.3.2. Синхронные двоичные счетчики ... 286
10.3.3. Расчет синхронных счетчиков ... 290
10.3.4. Синхронные *BCD*-счетчики ... 296
10.3.5. Синхронный счетчик для кода с избытком 3 ... 301
10.4. Делители частоты ... 305
10.4.1. Асинхронные делители частоты с фиксированным коэффициентом пересчета *K* ... 306
10.4.2. Синхронные делители частоты с фиксированным коэффициентом пересчета *K* ... 308
10.4.3. Делитель частоты с регулируемым коэффициентом пересчета ... 310
Контрольный тест ... 310

Глава 11. ЦИФРОВЫЕ СХЕМЫ ВЫБОРКИ И СВЯЗИ ... 314

11.1. Цифровой коммутатор, мультиплексор и демультиплексор ... 314
11.1.1. Мультиплексор 4 в 1 ... 314
11.1.2. Цифровой селектор 2 × 4 в 4 ... 315
11.1.3. Цифровой селектор 4 × 8 в 8 ... 316
11.1.4. Цифровой селектор-мультиплексор 16 в 1 ... 316
11.1.5. Демультиплексор 1 в 4 ... 316
11.2. Дешифраторы ... 319
11.2.1. 2-битовые дешифраторы ... 319
11.2.2. 4-битовые дешифраторы ... 319

11.3. Цифровые компараторы ... 321
11.3.1. 1-битовый компаратор ... 321
11.3.2. 3-битовый компаратор для *BCD*-кода (двоично-десятичного кода) ... 322
11.3.3. 4-битовый компаратор для двоичного кода ... 324
11.4. Шины данных ... 325
11.4.1. Структура и принцип действия ... 325
11.4.2. Стандарты шин ... 327
Контрольный тест ... 328

Глава 12. РЕГИСТРЫ И ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ... 329

12.1. Сдвигающие регистры ... 329
12.1.1. Сдвигающие последовательные регистры ... 329
12.1.2. Сдвигающие регистры с параллельным считыванием ... 332
12.1.3. Сдвигающие регистры с параллельным вводом—выводом данных ... 333
12.1.4. Кольцевой сдвигающий регистр ... 335
12.1.5. Сдвигающие реверсивные регистры ... 336
12.2. Регистры хранения ... 336
12.3. Оперативные запоминающие устройства (RAM) ... 338
12.3.1. Статические ОЗУ (SRAM) ... 339
12.3.2. Динамические ОЗУ (DRAM) ... 342
12.3.3. Организация элементов памяти и ее параметры ... 344
12.4. Постоянные запоминающие устройства (ПЗУ, ROM) ... 356
12.5. Программируемое постоянное запоминающее устройство (ППЗУ, PROM) ... 360
12.6. Перепрограммируемые постоянные запоминающие устройства ... 361
12.6.1. EPROM и REPROM ... 361
12.6.2. Постоянные запоминающие устройства EEROM (ЭСППЗУ — электрически стираемое программируемое постоянное запоминающее устройство) и EAROM ... 367
12.7. Магнитные запоминающие устройства ... 368
12.7.1. Магнитные кольца ... 368
12.7.2. Матрица магнитных колец памяти ... 369
12.7.3. Чтение и запись ... 369
12.8. Память на магнитных доменах ... 371
12.8.1. Цилиндрические магнитные домены ... 371
12.8.2. Магнитные дорожки ... 372
12.8.3. Запись информации ... 374
12.8.4. Чтение информации ... 374
12.8.5. Структура магнитной памяти ... 375
Контрольный тест ... 376

Глава 13. ЦИФРОАНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ, АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ ... 378

13.1. Цифроаналоговый преобразователь (ЦАП) ... 378
13.1.1. Принцип цифроаналогового преобразования ... 378
13.1.2. Цифроаналоговые преобразователи с весовыми резисторами ... 379
13.1.3. *R*/2*R*-ЦАП ... 381
13.2. Аналого-цифровой преобразователь (АЦП) ... 382
13.2.1. Принцип аналого-цифрового преобразования ... 382
13.2.2. АЦП последовательного счета ... 385
13.2.3. АЦП с двойным интегрированием (Dual Slope) ... 386
13.2.4. АЦП компенсационного типа ... 388
13.2.5. АЦП по принципу напряжение—частота ... 390
13.2.6. АЦП прямого преобразования ... 391
Контрольный тест ... 392

Глава 14. СЧЕТНЫЕ СХЕМЫ ... 393

14.1. Полусумматор ... 393
14.2. Полные сумматоры ... 394

14.3. Параллельный сумматор ... 398
14.4. Последовательный сумматор ... 398
14.5. Схемы вычитания ... 400
14.5.1. Полувычитатель ... 400
14.5.2. Полный вычитатель ... 402
14.5.3. 4-битовый вычитатель ... 403
14.5.4. Вычитатель на полных сумматорах ... 404
14.6. Универсальный сумматор-вычитатель ... 405
14.7. Умножители ... 408
14.7.1. Параллельные умножители ... 408
14.7.2. Последовательный умножитель ... 411
Контрольный тест ... 412

Глава 15. МИКРОПРОЦЕССОРЫ И МИКРОКОМПЬЮТЕРЫ ... 414

15.1. Микропроцессор как универсальная схема ... 414
15.2. Арифметико-логическое устройство (АЛУ) ... 414
15.3. Аккумулятор ... 417
15.4. Аккумулятор с памятью ... 419
15.5. Программно-управляемый упрощенный компьютер ... 421
15.6. Микропроцессоры ... 423
15.6.1. Виды микропроцессоров ... 423
15.6.2. Микропроцессор SAB 8080A ... 425
15.6.3. Дополнительные модули для микропроцессоров ... 427
15.7. Микрокомпьютер ... 430
Контрольный тест ... 432

Глава 16. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ СХЕМЫ ... 433

16.1. Логические схемы, программируемые изготовителем ... 433
16.2. Логические схемы, программируемые потребителем ... 435
16.2.1. Основы ... 435
16.2.2. PAL-схемы ... 438
16.2.3. GAL-схемы ... 440
16.2.4. FPLA-схемы ... 445
16.2.5. PROM-схемы ... 447
16.2.6. MACRO-схемы ... 447
16.3. Разновидности PLD ... 449
16.4. Программирование PLD ... 450
Контрольный тест ... 451

Глава 17. РЕШЕНИЯ ЗАДАНИЙ КОНТРОЛЬНЫХ ТЕСТОВ ... 453

Предисловие

Цифровая электроника является интереснейшей, стремительно развивающейся в последние годы областью современной электроники. Не только инженеры-электронщики, но и специалисты других дисциплин стремятся расширить познания в этой новой области техники. В «Электронике 3»* было изложено краткое введение в курс цифровой электроники. Данная книга, «Электроника 4» (в русском переводе «Цифровая электроника» – прим. ред.), позволяет приобрести обширные знания в этой области.

Предполагается, что читатель знаком с основами электротехники и электроники, которые необходимы для понимания некоторых глав этой книги. Большинство разделов доступно без специальных предварительных знаний. Читателю придется привыкнуть к принятому в данной области электроники «цифровому мышлению».

Особое внимание уделяется ясному и исчерпывающему изложению материала. Последовательно раскрывается мир цифровой электроники, ее взаимосвязь с другими областями техники. При этом использован опыт длительной преподавательской деятельности автора учебника.

Книга может использоваться как в качестве основного учебника для курсов по специальности, так и для самостоятельного обучения. Контрольные тесты с вопросами и задачами в конце каждой главы позволяют проверить понимание пройденного материала. Решения приведены в конце учебника.

Книга предназначена для студентов электротехнических и машиностроительных специальностей, инженеров-практиков, техников, а также всех, кто интересуется современной цифровой техникой.

Всем, кто принял участие в работе над книгой, выражаю свою сердечную признательность. Отдельная благодарность издательству Vogel Buchverlag. Всегда рад отзывам читателей и предложениям по улучшению текста.

Вальдкирх/Брайзгау

Клаус Бойт

---

* Elektronik 3.Grundschaltungen von Beuth, Klaus/Schmusch, Wolfgang. Vogel Buchverlag, 2003. На русский язык не переводилась. – Прим. ред.

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ

1.1. Аналоговое и цифровое представление величин

Понятия «аналоговый» и «цифровой» пришли из вычислительной техники и стали общепринятыми для всей электротехники, включая разделы измерительной техники (метрологии).

1.1.1. Аналоговое представление величин

Для представления величины в аналоговой форме требуется *аналогичная,* то есть «соответствующая» физическая величина. Для аналоговых вычислительных машин такой аналогичной величиной является электрическое напряжение. Так, например, число можно представить следующим образом:

> Числу 1 соответствует значение напряжения в 1 вольт (В).

$$1 \triangleq 1\ \text{В}$$

($\triangleq$ означает «соответствует»).

Далее 2 вольта соответствуют числу 2 и 3,6 вольта — числу 3,6. Для представления числа 4,365, таким образом, необходимо напряжение 4,365 В. Чтобы работать с большими числами, нужно выбрать другой порядок, т. е. другой масштаб, например 1 $\triangleq$ 1 мВ. Иначе мы будем иметь дело со слишком высоким напряжением.

> Аналоговые величины — это значения соответствующих физических величин, которые могут принимать любое значение в пределах допустимого диапазона.

Точность представления аналоговых величин зависит от точности, с которой могут быть измерены соответствующие им величины. Здесь мы быстро сталкиваемся с физическими ограничениями. Напряжение может быть измерено без применения специальных решений с точностью до $\pm 1\,\%$, с применением более дорогих решений — до $\pm 0{,}1\,‰$. При дальнейшем увеличении точности резко возрастает стоимость измерений. Другой физической границей является зависимость от температуры.

> Аналоговые величины могут обычно быть представлены с точностью до третьего десятичного знака.

Простой аналоговой вычислительной машиной является давно известная логарифмическая линейка. В качестве соответствующей величины используется длина. Длина соответствует численным значениям в логариф-

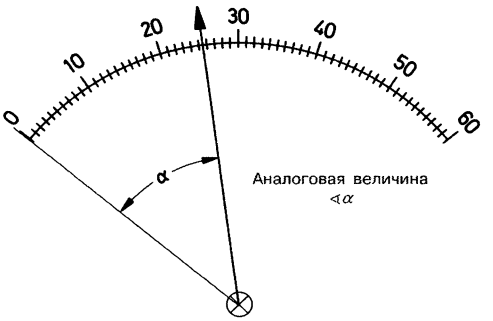


Рис. 1.1. Аналоговое представление измеряемой величины.

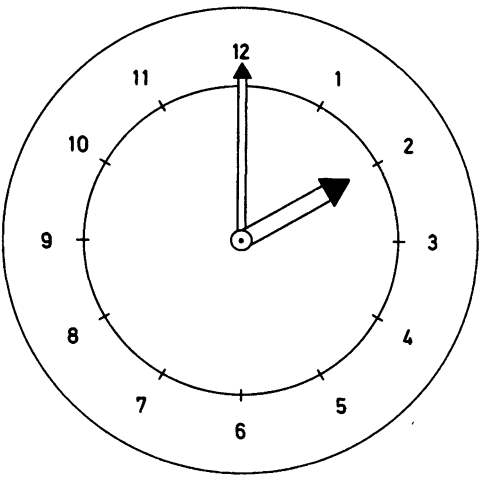


Рис. 1.2. Аналоговые часы.

мическом масштабе. Таким образом, зависимость между аналоговой величиной и соответствующими ей величинами также должна быть нелинейной. Точность логарифмической линейки зависит от возможности точно рассмотреть ее показания.

В измерительной технике аналоговое представление величин имеет особенно большое значение. Стрелочные измерительные приборы представляют измеряемые величины в аналоговом виде (рис. 1.1). Аналоговая величина — это угол, который стрелка образует со своей нулевой линией или соответствующий участок шкалы. Стрелка прибора может показать любое значение в пределах шкалы.

Стрелочные часы на рис. 1.2 показывают время в аналоговой форме. Аналоговой величиной является угол или дуга. Допустимым диапазоном в данном случае будут 360 градусов, т. е. полный оборот стрелки.

Диаграммы на рис. 1.3 также являются аналоговыми представлениями величин. Аналоговой величиной в данном случае является высота столбика.

Осциллограмма переменного напряжения на рис. 1.4 также является аналоговой величиной. Напряжение может принимать любые значения внутри допустимого диапазона.

При аналоговом представлении величин можно делать выводы о тренде (тенденции развития) процесса.

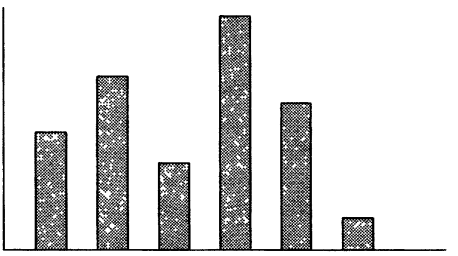


Рис. 1.3. Аналоговое представление, например, заработная плата различных профессий.

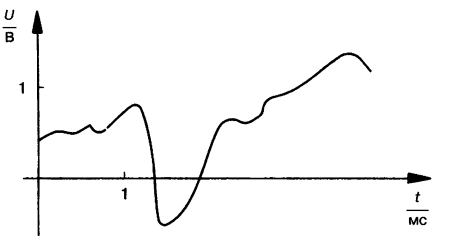


Рис. 1.4. Аналоговое представление напряжения.

1.1.2. Цифровое представление величин

При цифровом представлении применяют элементы, которые можно сосчитать. Понятие «цифровой» происходит от латинского слова digitus (палец). Число можно представить, например, количеством пальцев. Простым цифровым счетчиком являются всем известные счеты (рис. 1.5). Число представляется количеством костяшек.

> Цифровые величины состоят из элементов, которые можно сосчитать.

Преимущество цифрового представления очевидно. Точность представления не ограничивают физические эффекты. Если увеличить число костяшек на счетах, увеличится и точность.

> Цифровые величины можно представлять с любой точностью.

В электронных калькуляторах вместо костяшек применяют электрические импульсы. Число 3 можно представить тремя импульсами, а число 37 — соответственно 37. Однако такое представление нерационально и не применяется. Ведь для числа 100 000 было бы необходимо использовать 100 000 импульсов, а это почти нереально.

Для представления чисел в цифровом виде применяют так называемые коды. На рис. 1.6 показана диаграмма цифрового сигнала.

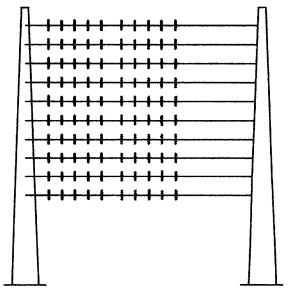


Рис. 1.5. Счеты как пример простого цифрового счетчика.

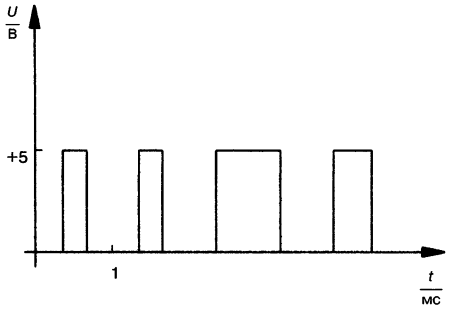
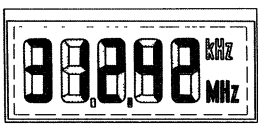


Рис. 1.6. Временная диаграмма цифрового сигнала.

Так как цифровые величины состоят из элементов, которые можно сосчитать, то для наглядности применяют представление величин в виде чисел.

> Отображение информации с помощью цифр называется цифровой индикацией.

Измерительные приборы с цифровой индикацией называются **цифровыми приборами** (рис. 1.7). Часы с цифровой индикацией называются **цифровыми часами**.



> Цифровые индикаторы однозначны.

Рис. 1.7. Цифровая индикация измерительного прибора.

1.2. Бинарные и логические состояния

Цифровая величина состоит, как было установлено в предыдущем разделе, из элементов, которые можно сосчитать. Эти элементы могут иметь два, три или больше состояний. На рис. 1.8 изображен сигнал с тремя возможными состояниями: 10 В, 5 В и 0 В.

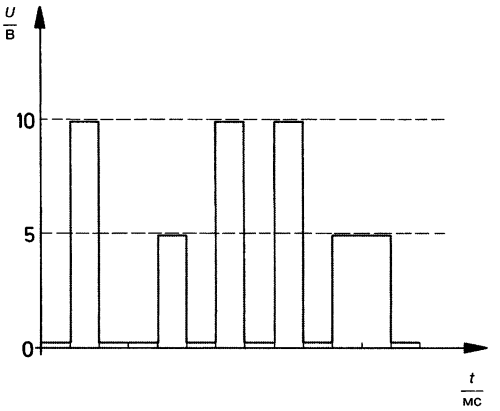


Рис. 1.8. Цифровой сигнал с тремя возможными состояниями.

> Цифровые сигналы могут иметь два, три или больше значений, т. е. два, три или больше фиксированных состояний.

Однако в цифровой электронике элементы почти всегда имеют только два состояния. Транзистор может быть либо закрыт, либо насыщен. Электрический импульс или существует, или нет. Есть только два возможных состояния цифрового элемента. Напряжение имеет согласованное верхнее значение или согласованное нижнее значение (с определенным допуском).

> Обычные цифровые элементы являются «двузначными», т. е. имеют два возможных состояния.

Свойство двузначности элементов выражают термином «бинарность» (от латинского слова bin — дважды). Применяемые в цифровой электронике элементы являются бинарными элементами.

> Так как цифровая электроника использует только бинарные элементы, более точно ее следовало бы называть «бинарная цифровая электроника».

Соответственно для цифровых микросхем также должно было бы использоваться обозначение «бинарные цифровые микросхемы». Но так как в настоящее время не имеется — по крайней мере в технической области — никакой другой цифровой техники, то дополнительное слово «бинарная» можно не использовать. Совсем недавно узнали, что существует четырехзначная «цифровая техника» в мире живых существ. Она используется прежде всего для кодирования, сохранения, выборки и передачи наследственной информации. Результаты дальнейших исследований покажут, превосходит ли цифровая техника природы придуманную людьми.

Бинарные состояния в цифровой технике также называют **цифровыми**. Примеры бинарных состояний:

Первое бинарное состояние	Второе бинарное состояние
Ключ разомкнут	Импульс есть
Транзистор открыт	Диод открыт
Напряжение высокое	Ток большой
Материал магнитный	Ключ замкнут
Импульса нет	Транзистор закрыт
Диод закрыт	Напряжение низкое
Ток малый	Материал немагнитный

Так как в цифровой электронике работают с микросхемами, то прежде всего бинарными являются уровни напряжения. Производители указывают для цифровых микросхем бинарные состояния напряжения в инструкциях по эксплуатации. Типичные бинарные состояния напряжения:

+2 В	0 В (заземление)
+5 В	0 В (заземление)
+5 В	−5 В
+12 В	0 В
0 В	−12 В

Для бинарных состояний напряжения имеются определенные допуски (рис. 1.9). Например, напряжение может иметь одно бинарное состояние в пределах от 4 до 5,5 В. Напряжение другого бинарного состояния может быть между 0 В и +0,8 В. НИЗКИЙ уровень напряжения обозначается символом *L* (от английского low — низкий), ВЫСОКИЙ уровень напряжения символом *H* (от английского high — высокий).

L = Low = НИЗКИЙ уровень.

Уровень, стремящийся к минус бесконечности ($-\infty$).

L = High = ВЫСОКИЙ уровень.

Уровень, стремящийся к плюс бесконечности ($+\infty$).

Бинарные состояния сами по себе ничего не выражают. Им должны быть поставлены в соответствие так называемые логические состояния.

Логическое состояние 1 в алгебре логики называют «верно» или «истинно». Логическое состояние 0 значит «неверно» или соответственно «ложно».

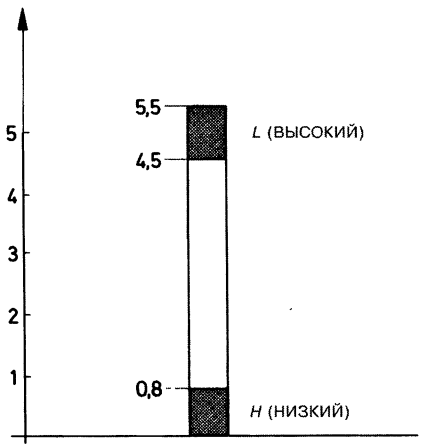


Рис. 1.9. Допуски бинарных уровней напряжения.

Присваивание бинарного состояния логическому производится произвольно.

Принятое соответствие бинарных и логических состояний должно соблюдаться для всего проекта. Типичный пример:

$0 \triangleq L = 0$ В (заземление)

$1 \triangleq H = +5$ В

или еще такой вариант:

$$0 \triangleq H = +5 \text{ В}$$

$$1 \triangleq L = 0 \text{ В (заземление).}$$

В системах, в которых логические состояния эквивалентны каким-либо сигналам или физическим величинам, например положительным или отрицательным импульсам, наличию или отсутствию импульсов, двум различным частотам и т. д. — для представления этих сигналов и величин можно применять термины ВЫСОКИЙ уровень (символ H) и НИЗКИЙ уровень (символ L). При этом требуется соблюдать однозначность присвоения.

Не следует путать между собой понятия бинарных состояний (например ВЫСОКИЙ и НИЗКИЙ уровни) и логических состояний. Логические состояния в алгебре логики называют также **величинами**. Подробнее о разнице в этих понятиях можно прочитать в DIN 40900, часть 12 (Обозначения условные графические на электрических схемах).

Контрольный тест

1. Чем отличается цифровая величина от аналоговой?
2. Назовите преимущества и недостатки аналогового представления данных.
3. Что понимают под бинарной величиной?

4. Какая точность возможна при цифровом представлении данных?

5. В инструкциях производители цифровых схем часто указывают обозначения *L* и *H*. Что означают эти символы?

6. Что такое логические состояния и какими символами они обозначаются?

7. Как представляются данные

а) аналоговым измерительным прибором;

б) цифровым измерительным прибором?

ГЛАВА 2

ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

2.1. Основные законы и элементы алгебры логики

2.1.1. Логический элемент И и операция логического умножения (конъюнкции)

Предложение «Если завтра будет хорошая погода и брат приедет, то мы пойдем на рыбалку» содержит в себе операцию логического *умножения* И. Условие *A* (хорошая погода) и условие *B* (брат приедет) должны одновременно выполниться, чтобы действие *X* (рыбалка) свершилось. Сказанное иллюстрирует таблица истинности (рис. 2.1). Состояние 1 значит «верно» или «истина». Состояние 0 значит «неверно» или «ложь». Возможны четыре комбинации. Последовательность комбинаций в принципе не имеет значения, однако, как будет показано позже, она должна соответствовать определенной схеме.

Электронную схему, в которой сигнал 1 на выходе появляется только тогда, когда на входе *A* и входе *B* совпадают сигналы 1, называется логическим элементом И (И-вентиль).

Простейший И-вентиль на последовательно включенных контакторах может быть реализован по схеме на рис. 2.2. Но в настоящее время почти всегда применяются интегрированные полупроводниковые микросхемы (см. раздел «Семейства схем»).

Любая схема, удовлетворяющая таблице истинности логического умножения, является логическим элементом И.

Для обозначения операции И в алгебре логики используется символ $\wedge$.

$$X = A \wedge B$$

В литературе встречаются другие символы для обозначения логического умножения, точка ($\cdot$) или &:

$$X = A \cdot B;\ X = A\ \&\ B.$$

Вар	*B*	*A*	*X*
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Рис. 2.1. Таблица истинности логического умножения и И-элемента.

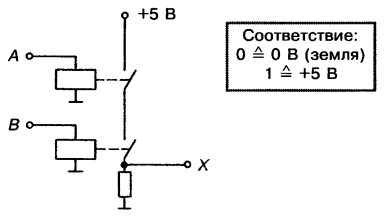


Рис. 2.2. И-элемент.

Scan Bookingolz

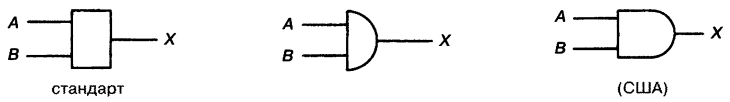


Рис. 2.3. Условные обозначения И-элементов с двумя входами.

Условное обозначение логического элемента И с двумя входами показано на рис. 2.3. Обозначения входов и выходов могут быть любыми. Часто входы обозначают *A* и *B*, а выход — *X* или *Q*.

На выходе логического элемента И сигнал 1 появится только тогда, когда на всех входах совпадут сигналы 1.

2.1.2. Логический элемент ИЛИ и операция логического сложения (дизъюнкции)

Предложение «Если я получу наследство или выиграю в лотерею, то поеду в кругосветное путешествие» содержит в себе операцию логического *сложения* ИЛИ. Путешествие становится возможным при истинности условия *A* (наследство) или условия *B* (лотерея), или при выполнении обоих условий одновременно. Сказанное иллюстрирует таблица истинности на рис. 2.4 (состояние 1 значит «истина», состояние 0 значит «ложно»).

Вар.	*B*	*A*	*X*
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Рис. 2.4. Таблица истинности логического сложения и ИЛИ-элемента.

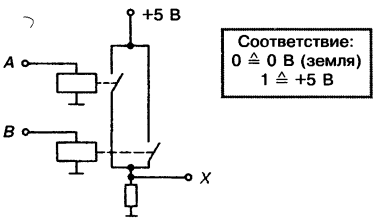


Рис. 2.5. ИЛИ-элемент.

Электронная схема, на выходе *X* которой появляется сигнал 1, если на входе *A* или входе *B* или на обоих входах присутствует сигнал 1, называется логическим элементом ИЛИ. Элемент ИЛИ может быть реализован по схеме на рис. 2.5.

Релейная схема приведена для наглядности. Сегодня элементы ИЛИ почти всегда используются в виде интегрированных полупроводниковых микросхем.

> Любая схема, удовлетворяющая таблице истинности логического сложения, является логическим элементом ИЛИ.

Для обозначения операции ИЛИ в алгебре логики используется символ ∨.

$$X = A \vee B$$

Также в литературе встречается знак + для обозначения логического сложения.

$$X = A + B$$

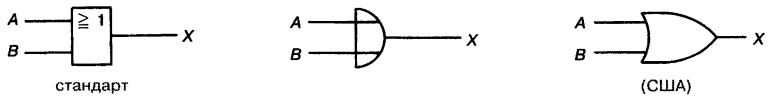


Рис. 2.6. Условные обозначения ИЛИ-элементов с двумя входами.

Условное обозначение логического элемента ИЛИ с двумя входами показано на рис. 2.6. Символ $\geqq 1$ означает, что хотя бы на одном из входов должен быть сигнал 1 для появления единицы на выходе.

На выходе логического элемента ИЛИ сигнал 1 появится только тогда, когда хотя бы на одном из его входов присутствует сигнал 1.

2.1.3. Логический элемент НЕ и операция инверсии (отрицания)

Предложение «Если приедет брат, то я не пойду сегодня вечером в театр» означает *отрицание*. Если высказывание *A* (приезд брата) верно, то действие *X* (посещение театра) не произойдет. Если неверно высказывание *A*, то высказывание *X* будет верным, и я иду в театр. Соответствующая таблица истинности (рис 2.7) имеет только два возможных варианта.

Электронную схему, состояние на выходе *X* которой всегда противоположно состоянию на входе *A*, называют логическим элементом НЕ или инвертором.

На рис. 2.8 приведена схема логического элемента НЕ. Как и ранее рассмотренные логические элементы, вентили НЕ почти всегда используются в виде интегрированных полупроводниковых микросхем.

Любая схема, удовлетворяющая таблице истинности логического инвертирования, является логическим элементом НЕ.

Вар.	*A*	*X*
1	0	1
2	1	0

Рис. 2.7. Таблица истинности логического отрицания и НЕ-элемента.

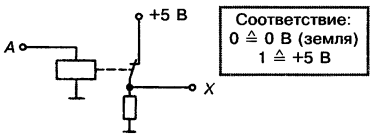


Рис. 2.8. НЕ-элемент.

Для обозначения операции НЕ в алгебре логики используется черта над символом или апостроф:

$$X = \bar{A}$$

Условное обозначение логического элемента НЕ показано на рис. 2.9.

Состояние выхода логического элемента НЕ всегда противоположно состоянию входа.

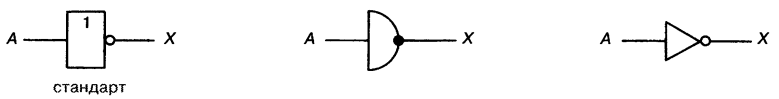


Рис. 2.9. Условные обозначения НЕ-элементов (USA).

2.1.4. Основные логические элементы

Логические элементы И, ИЛИ и НЕ предназначены для выполнения трех основных операций цифровой логики над дискретными сигналами. С помощью этих элементов можно реализовать логические операции любой сложности. Поэтому эти элементы называются **основными** (рис. 2.10). К основным логическим элементам относится также буфер (рис. 2.10а). Если на входе буфера 1, то и на выходе 1, иначе 0.

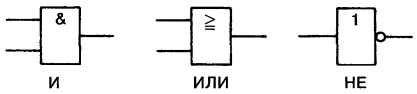


Рис. 2.10. Стандартизованные условные графические обозначения элементов.

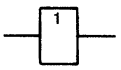


Рис. 2.10а. Усилитель уровня без отрицания (буфер).

2.2. Комбинированные элементы

2.2.1. Логический элемент И-НЕ

Если соединить последовательно элементы И и НЕ согласно схеме на рис. 2.11, то выход X логического элемента И инвертируется согласно таблице истинности (рис 2.12). В колонке X приведен выходной сигнал элемента И. X равен 1 только если $A = 1$ и $B = 1$ (вариант 4). В свою очередь X является входом для инвертора. Если на входе X элемента НЕ логическая 1, то выход $Z = 0$. Если на входе X элемента НЕ логический 0, то выход $Z = 1$.

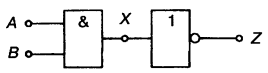


Рис. 2.11. Образование соединения И-НЕ.

Вар.	B	A	X	Z
1	0	0	0	1
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

Рис. 2.12. Таблица истинности схемы рис. 2.11.

В колонке Z приведен инвертированный выход X элемента И.

В англоязычной литературе такой комбинированный элемент обозначается как NAND (сокр. от NOT + AND).

Вентили И-НЕ используются очень часто. Для них придумали собственное условное обозначение (рис. 2.13). Оно получается из символа вентиля И с кружком на выходе. Этот кружок означает инвертирование выхода.

Для логического элемента И-НЕ действительно следующее утверждение:

> На выходе логического элемента И-НЕ логическая 1 будет только в том случае, если не на всех входах наступает состояние 1.

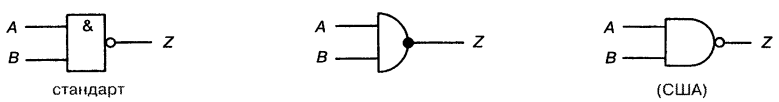


Рис. 2.13. Условные обозначения И-НЕ-элементов с двумя входами.

Логическая функция элемента И-НЕ отвечает выражению:

$$Z = \overline{A \wedge B}.$$

Длинная черта над $A \wedge B$ указывает, что инвертируется все выражение. Таблица истинности элемента И-НЕ приведена на рис. 2.14.

Вар.	B	A	Z
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

Рис. 2.14. Таблица истинности И-НЕ-элемента.

2.2.2. Логический элемент ИЛИ-НЕ

Таблица истинности соединенных последовательно элементов ИЛИ и НЕ согласно схеме на рис. 2.15 приведена на рис 2.16. Сначала входные сигналы A и B поступают на вентиль ИЛИ:

$$X = A \vee B.$$

X является одновременно входом элемента НЕ. Все состояния X инвертированы в столбце Z (из $X = 0$ будет $Z = 1$, из $X = 1$ будет $Z = 0$).

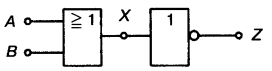


Рис. 2.15. Образование элемента ИЛИ-НЕ.

Вар.	B	A	X	Z
1	0	0	0	1
2	0	1	1	0
3	1	0	1	0
4	1	1	1	0

Рис. 2.16. Таблица истинности схемы рис. 2.15.

Выход Z является выходом элемента ИЛИ-НЕ. В англоязычной литературе такой комбинированный элемент обозначается как NOR (сокр. от NOT + OR).

Логические элементы ИЛИ-НЕ используются так же часто, как и элементы И-НЕ. Для них так же создано собственное условное обозначение (рис. 2.17). Оно получается из символа вентиля ИЛИ с кружком на выходе. Этот кружок означает инвертирование выхода.

Для логического элемента ИЛИ-НЕ действительно следующее утверждение:

> На выходе логического элемента ИЛИ-НЕ логическая 1 будет только в том случае, если ни на одном из входов нет состояния 1.

Логическая функция элемента ИЛИ-НЕ отвечает выражению:

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

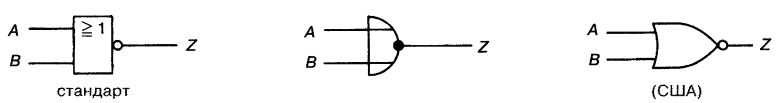


Рис. 2.17. Условные обозначения ИЛИ-НЕ-элементов с двумя входами.

2.2.3. Логический элемент эквивалентности

Часто возникает необходимость в схемах, в которых на выходе всегда логическая 1, когда на оба входа поданы одинаковые логические сигналы — либо оба 0, либо оба 1. Такая схема называется логическим элементом эквивалентности (эквивалентность — равноценность). Она строится из основных логических элементов соответственно (рис. 2.18).

Разберем подробно таблицу истинности эквивалентного элемента. Сначала для четырех возможных комбинаций записываются логические состояния входов A и B (рис. 2.19, столбцы ① и ②). Затем они инвертируются элементом НЕ, превращаясь в $\bar{A}$ и $\bar{B}$. Если $A = 0$, то соответственно $\bar{A} = 1$. Если, как в случае 4, $A = 1$, то соответственно $\bar{A} = 0$. Такое же правило действует и для B и $\bar{B}$. Так получают содержимое столбцов ③ и ④ на рис. 2.19. Состояние выхода Q получается из операции логического умножения A и B. В случае 1 $A = 0$, $B = 0$, следовательно, Q должен быть также равен 0 (столбец ⑤). В случаях 2 и 3 Q равен также 0, так как оба входа не являются логической 1. Только в случае 4, где $A = 1$ и $B = 1$, Q также равен 1.

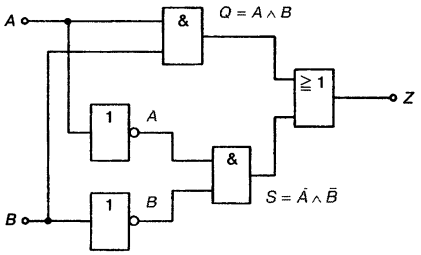


Рис. 2.18. Образование элемента эквивалентности из основных элементов.

Вар.	① B	② A	③ $\bar{B}$	④ $\bar{A}$	⑤ $Q = A \wedge B$	⑥ $S = \bar{A} \wedge \bar{B}$	⑦ $Z = Q \vee S$
1	0	0	1	1	0	1	1
2	0	1	1	0	0	0	0
3	1	0	0	1	0	0	0
4	1	1	0	0	1	0	1

Рис. 2.19. Образование таблицы истинности для элемента эквивалентности.

Символом S в колонке 6 обозначается результат логического умножения $\bar{A}$ и $\bar{B}$. $\bar{A}$ и $\bar{B}$ являются входами логического элемента ИЛИ с выходом S (рис. 2.18). В первом случае $\bar{A} = 1$ и $\bar{B} = 1$. Следовательно, для первого случая $S = 1$. В случаях 2 и 3 таблицы истинности $S = 0$, так как только один из входов имеет состояние 1. В случае 4 оба входа равны 0 и соответственно $S = 0$.

S и Q являются выходами обоих элементов И и одновременно входами элемента ИЛИ. Логический элемент ИЛИ производит операцию логического сложения состояний S и Q. В случае 1 $Q = 0$ и $S = 1$. Следовательно, на выходе Z (столбец ⑦) также 1. В случаях 2 и 3 оба входа 0, и таким образом, на выходе также 0. В случае 4 $Q = 1$ и $S = 0$, что при операции сложения дает результат 1.

Для элементов эквивалентности также создано собственное условное обозначение. Условное обозначение и таблица истинности приведены на рис. 2.20.

На выходе элемента эквивалентности состояние 1 будет только тогда, когда входы имеют равное состояние.

Логическая функция элемента эквивалентности отвечает выражению:

$$Z = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$$

Так как в нашем примере $Q = A \wedge B$ и $S = \bar{A} \wedge \bar{B}$, можно также записать $Z = Q \vee S$. Элемент эквивалентности можно построить и из других основных логических элементов (см. задания в конце гл. 2).

Вар.	B	A	Z
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	1

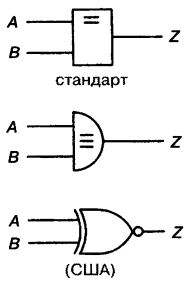


Рис. 2.20. Условное обозначение элемента эквивалентности и таблица истинности.

2.2.4. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR)

Если выход элемента эквивалентности инвертируется посредством последовательного подключения элемента НЕ, то возникает элемент, который на выходе всегда имеет 1, если его входы различны (рис. 2.21).

Такой элемент называется элементом ИСКЛЮЧАЮЩЕЕ ИЛИ. Речь идет при этом об элементе ИЛИ, в котором исключен случай, когда на выходе находится 1, если оба входа имеют 1 (случай 4). В англоязычной литературе такой элемент обозначается как XOR (сокр. от EXCLUSIVE + OR).

Логические элементы ИСКЛЮЧАЮЩЕЕ ИЛИ используются достаточно часто. Условное обозначение и таблица истинности приведены на рис. 2.22.

> На выходе элемента ИСКЛЮЧАЮЩЕЕ ИЛИ состояние 1 будет только тогда, когда оба входа имеют различное состояние.

Логическая функция элемента эквивалентности может быть получена из схемы на рис. 2.21:

$$X = (A \wedge B) \vee (\bar{A} \wedge \bar{B}).$$

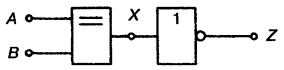
Тогда логическую функцию элемента ИСКЛЮЧАЮЩЕЕ ИЛИ можно представить как инверсию логической функции элемента эквивалентности:

$$Z = \overline{(A \wedge B) \vee (\bar{A} \wedge \bar{B})}$$

Это выражение можно преобразовать с помощью правил алгебры логики:

$$Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$$

Правила преобразования подробно рассматриваются в гл. 4.



Вар.	B	A	X	Z
1	0	0	1	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

Рис. 2.21. Образование XOR-элемента и его таблица истинности.

Вар.	B	A	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

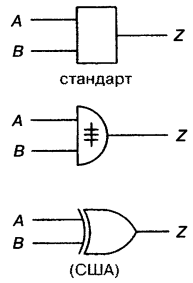


Рис. 2.22. Условное обозначение XOR-элемента и его таблица истинности.

2.2.5. Комбинации элементов с двумя входами

После рассмотрения элементов И, ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ, ЭКВИВАЛЕНТНОСТИ и ИСКЛЮЧАЮЩЕГО ИЛИ посмотрим варианты дальнейших возможных комбинаций и соответствующие им элементы. Существует еще много вариантов соединения, но они не имеют большого практического значения.

Для элементов с двумя входами (например *A* и *B*) возможны 4 различных варианта (комбинаций) входов, как мы видели в рассмотренных до сих пор таблицах истинности (см. рис. 2.22). Для этих 4 вариантов возможны 4 возможных варианта на выходе, например выход *Z* на рис. 2.23. В каждом из серых квадратиков может быть выходное состояние 0 или 1.

Вар.	*B*	*A*	*Z*
1	0	0	□
2	0	1	□
3	1	0	□
4	1	1	□

Рис. 2.23. Таблица истинности для элементов с двумя входами. Серые квадраты для возможных состояний выходов.

Можно составить 16 различных комбинаций выходных состояний. Они обозначены на рис. 2.24 от Z_1 до Z_{16}. Из рисунка сразу становится ясно, что некоторые из возможных комбинаций не имеют особого значения. Для «константы 0» и «константы 1» не нужно вводить никаких элементов. «Константа 0» означает, что выход всегда равен 0, абсолютно независимо от того, какие состояния на входах. При «константе 1» на выходе всегда 1, также независимо от состояния на входах.

Вар.	*B*	*A*	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}	Z_{16}
1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
2	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
4	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			Константа 0	ИЛИ-НЕ	Запрет *B*	Инверсия *B*	Запрет *A*	Инверсия *A*	Антиэквивалентность	И-НЕ	И	Эквивалентность	Тождественно *A*	Импликация *B*	Тождественно *B*	Импликация *A*	ИЛИ	Константа 1

Рис. 2.24. Общая таблица для 16 возможных состояний выходов элементов с двумя входами.

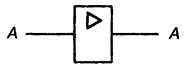


Рис. 2.25. Условное обозначение неинвертирующего усилителя.

«Инверсия *A*» и «инверсия *B*» соответственно реализуются логическим элементом НЕ. Для «тождественно *A*» и «тождественно *B*» можно использовать неинвертирующий усилитель (рис. 2.25).

> На выходе неинвертирующего усилителя только тогда логическая 1, когда на вход подана логическая 1.

Усилители такого рода предназначены для усиления слабых сигналов.

Запрещение является особой разновидностью элемента И. Состояние входа инвертируется перед элементом И. Если инвертируется вход *A*, то элемент называется **схема запрета *A*** (рис. 2.26). Если инвертируется вход *B*, то элемент называется **схема запрета *B*** (рис. 2.27).

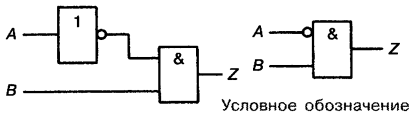


Рис. 2.26. Образование элемента «запрет *A*» и его условное обозначение.

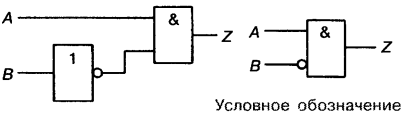


Рис. 2.27. Образование элемента «запрет *B*» и его условное обозначение.

Импликация является особой разновидностью элемента ИЛИ. Состояние входа инвертируется перед элементом ИЛИ. Если инвертируется вход *A*, то элемент называется **импликатор *A*** (рис. 2.28). Если инвертируется вход *B*, то элемент называется **импликатор *B*** (рис. 2.29).

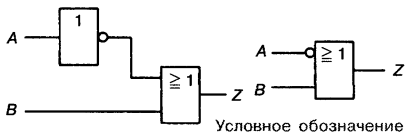


Рис. 2.28. Образование элемента «импликация *A*» и его условное обозначение.

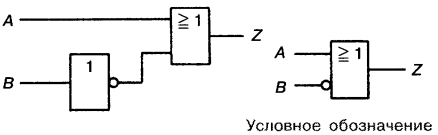


Рис. 2.29. Образование элемента «импликация *B*» и его условное обозначение.

Логические элементы запрещения и импликации имеют ограниченное практическое значение и почти не производятся. В случае необходимости их можно собрать из основных логических элементов.

2.3. Логические элементы с тремя и более входами

Если необходимы три входа или больше, то можно включить последовательно несколько двухвходовых элементов (рис. 2.30).

Каждый элемент с двумя входами имеет, как известно, 4 возможных комбинаций входов-выходов. Для входов *A* и *B* имеет силу обыкновенная таблица истинности. Если добавляется еще один вход, например *C*, то он может быть либо 0, либо 1.

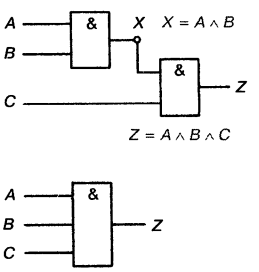


Рис. 2.30. Соединение двух И-элементов с двумя входами в один И-элемент с тремя входами.

Вар.	C	B	A	Z
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

Рис. 2.31. Таблица истинности И-схемы и И-элемента с тремя входами.

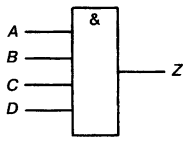


Рис. 2.32. И-элемент с четырьмя входами.

Рис. 2.33. Таблица истинности И-элемента с четырьмя входами.

Вар.	D	C	B	A
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Прежние 4 комбинации от *A* и *B* комбинируются один раз с $C = 0$ и второй раз с $C = 1$ (рис. 2.31). Таким образом получаются 8 комбинаций. Если теперь к трем входам, например *A*, *B*, *C* добавляют четвертый вход, например *D* (рис. 2.32), то прежние 8 комбинаций от *A*, *B* и *C* комбинируются один раз с $D = 0$ и второй раз с $D = 1$ (рис. 2.33). Член с 4 входами имеет, таким образом, 16 возможных комбинаций (рис. 2.33).

> С добавлением каждого нового входа число комбинаций (вариантов вход—выход) таблицы истинности удваивается.

При двух входах — 4 комбинации, при трех входах — 8 комбинаций, при четырех входах — 16 комбинаций и при пяти входах получаются 32 комбинации. При формировании таблиц истинности последовательность комбинаций выбирается произвольно. Нужно учитывать все варианты и не допускать повторов. Чтобы проще было составлять таблицы истинности, предлагаем следующую схему.

Первый вход (например *A*) меняет состояние каждый раз. Второй вход (например *B*) меняет состояние через раз. Третий вход (например *C*) меняет состояние через 4 варианта. Если продолжать по этой схеме, четвертый вход (например *D*) меняет состояние соответственно после 8 комбинаций, и так далее. Эта схема оправдала себя на практике. Указанные в данной книге таблицы истинности составлены согласно этой схеме.

Выпускаемые в настоящее время вентили И и ИЛИ имеют в основном от 2 до 4 входов. То же самое относится к вентилям И-НЕ и ИЛИ-НЕ. Изредка встречаются вентили с 8 и более входами.

Контрольный тест

1. Изобразите условное обозначение для вентилей И, ИЛИ, НЕ, И-НЕ и ИЛИ-НЕ. Все элементы, включая НЕ, должны иметь два входа.
2. Постройте таблицу истинности вентиля ИЛИ с тремя входами. Входы имеют обозначения *A*, *B*, *C*. Выход имеет обозначение *Z*.
3. Предложите вариант построения вентиля И-НЕ из основных логических элементов.
4. Изобразите таблицу истинности элемента НЕ с входом *A* и выходом *Y*.
5. Для элемента ИСКЛЮЧАЮЩЕЕ ИЛИ верно уравнение

$$Z = \left(A \wedge \overline{B}\right) \vee \left(\overline{A} \wedge B\right).$$

Синтезируйте его из логических элементов И, ИЛИ и НЕ и нарисуйте схему.

6. Опишите словами функции логических элементов И и ИЛИ.
7. Сколько возможных комбинаций имеет таблица истинности элемента ИЛИ с шестью входами?

8. Что понимают под логическим элементом ИСКЛЮЧАЮЩЕЕ ИЛИ? Изобразите для этого элемента таблицу истинности.

9. Как называется логический элемент, которому соответствует таблица истинности, изображенная на рис. 2.34?

10. Какую функцию выполняет элемент ЗАПРЕЩЕНИЕ? Как его можно построить из основных логических элементов? Нарисуйте возможную схему.

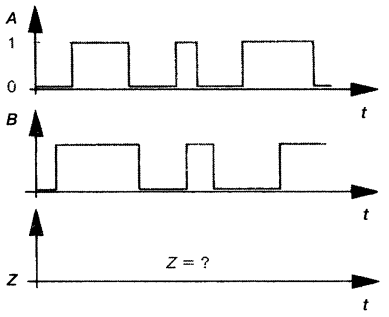
Вар.	*B*	*A*	*Z*
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	0

Рис. 2.34. Таблица истинности.

11. Временные диаграммы входов *A* и *B* представлены на рис. 2.35. Изобразите временную диаграмму выхода *Z*, если *A* и *B* входы

а) вентиля И,

б) вентиля ИЛИ.



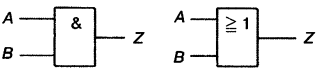
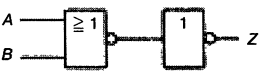


Рис. 2.35. Связь двух входных сигналов *A* и *B*.

12. Какую логическую операцию производят элементы на схеме рис. 2.36?

Рис. 2.36. Комбинация логических элементов.



13. Изобразите таблицу истинности элемента НЕ с пятью входами. Входы обозначить как E_1, E_2, E_3, E_4 и E_5. Выход обозначить как *X*.

14. На рис. 2.37 представлены входные сигналы *A* и *B* и выходной сигнал *Z* неизвестного элемента. Какую логическую операцию производит этот член?

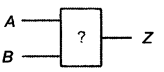
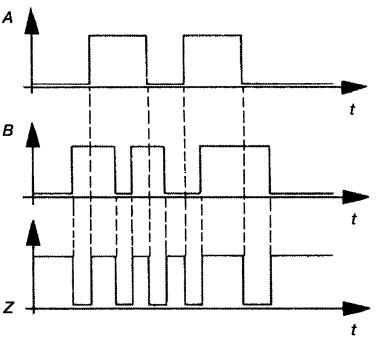


Рис. 2.37. Временные диаграммы входов и выхода.



ГЛАВА 3

АНАЛИЗ СХЕМ

Логические элементы, также называемые вентилями, редко применяются отдельно.

В подавляющем большинстве случаев схема в цифровой электронике состоит из достаточно большого количества последовательно соединенных логических элементов, которые совместно реализуют желаемую логическую функцию. Для практической работы очень важно уметь анализировать структуры соединений логических элементов. Иными словами, нужно уметь определять, какие логические операции производит каждый логический элемент в отдельности и какую функцию выполняет структура элементов схемы в целом. Процесс определения этих операций и функций называется **анализ схемы**.

Понятие «цифровая схема» в данной главе следует понимать как структуру, набор цифровых элементов без временной зависимости. Цифровые микросхемы с временной зависимостью будут рассмотрены в следующих разделах.

3.1. Таблица истинности и цифровая схема

В разделе 2.2 с помощью таблиц истинности были определены логические функции схем, составленных из нескольких основных логических элементов. И так же, как для нескольких последовательно включенных элементов можно определить таблицу истинности, ее можно определить для любой цифровой схемы, состоящей из большого числа логических элементов.

> Для любой цифровой схемы существует таблица истинности.

3.1.1. Таблица истинности цифровой схемы с двумя входами

Составим таблицу истинности для цифровой схемы на рис. 3.1. Таблица истинности позволяет определить, какие логические операции выполняет данная схема.

Так как схема имеет два входа (A, B), возможны только 4 варианта. Номера вариантов и комбинации состояний входов для A и B записываются по ранее описанной схеме (рис. 3.2).

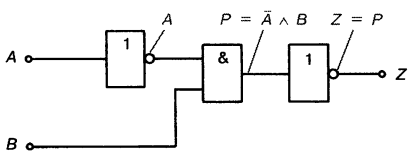


Рис. 3.1. Цифровая схема.

Вар	B	A
1	0	0
2	0	1
3	1	0
4	1	1

Рис. 3.2. Первый шаг при составлении таблицы истинности.

Вар.	B	A	$\bar{A}$	$P = A \wedge B$	$Z = \bar{P}$
1	0	0	1	0	1
2	0	1	0	0	1
3	(1)	0	(1)	(1)	0
4	1	1	0	0	1

Рис. 3.3. Следующие шаги при составлении таблицы истинности.

Первый элемент является элементом НЕ. Если состояние его входа обозначить как A, то на выходе будет инвертированное значение $\bar{A}$. Четвертый столбец таблицы истинности содержит инвертированные значения $\bar{A}$ (обозначено серым на рис. 3.3). Если в варианте $A = 0$, то $\bar{A} = 1$. Это варианты 1 и 3. Если в варианте $A = 1$, то $\bar{A} = 0$. Это варианты 2 и 4.

Вход элемента И (второго элемента) имеет обозначение $\bar{A}$, другой вход обозначен как B. Логическое умножение происходит между состояниями $\bar{A}$ и B. Соответствующие столбцы в таблице истинности (рис. 3.3) подчеркнуты серым. Выход элемента И обозначен как P, для которого верно равенство:

$$P = \bar{A} \wedge B.$$

P равно 1 только тогда, когда как $\bar{A} = 1$, так и $B = 1$. Это верно только для варианта 3. Значит $P = 1$ только в третьем случае, а для всех других $P = 0$ (рис. 3.3).

На входе третьего элемента (НЕ) находится состояние P, выход обозначим как Z. Так как этот логический элемент инвертирует состояние P, то $Z = \bar{P}$. Из 0 в P-столбце будет 1 в Z-столбце. Z-столбец является столбцом результата. В нем находится результат логической функции цифровой схемы.

Для Z можно записать следующие уравнения:

$$Z = \bar{P};$$

$$Z = \overline{\bar{A} \wedge B} \ \left(\text{так как } P = \bar{A} \wedge B\right).$$

Эти уравнения описывают принцип действия схемы или, другими словами, логическую функцию, которую она реализует.

3.1.2. Таблица истинности цифровой схемы с тремя входами

Составим теперь таблицу истинности для цифровой схемы с тремя входами (по рис. 3.4).

Цифровая микросхема с тремя входами имеет 8 вариантов. Номера вариантов и комбинации состояний входов для A, B и C записываются по ранее описанной в разд. 2.3 схеме (рис. 3.5). Для $\bar{A}$ и $\bar{B}$ предусмотрены два столбца.

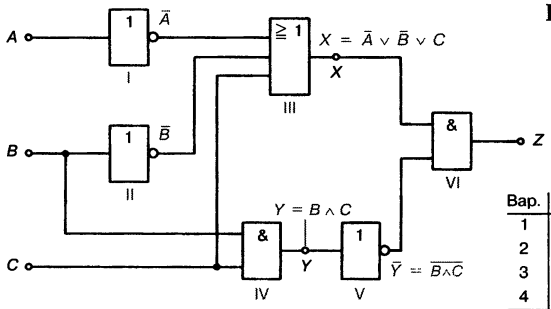


Рис. 3.4. Цифровая схема.

Вар.	C	B	A	$\bar{A}$	$\bar{B}$	$X = \bar{A} \vee \bar{B} \vee C$	$Y = B \wedge C$	$\bar{Y}$	$Z = X \wedge \bar{Y}$
1	0	0	0	1	1	1	0	1	1
2	0	0	1	0	1	1	0	1	1
3	0	1	0	1	0	1	0	1	1
4	0	1	1	0	0	0	0	1	0
5	1	0	0	1	1	1	0	1	1
6	1	0	1	0	1	1	0	1	1
7	1	1	0	1	0	1	1	0	0
8	1	1	1	0	0	1	1	0	0

Рис. 3.5. Таблица истинности схемы рис. 3.4.

В столбце $\bar{A}$ находятся состояния, инверсные состоянию A (из 0 будет 1, из 1 будет 0).

Входы элемента ИЛИ обозначены $\bar{A}$, $\bar{B}$, C. Содержимое этих трех столбцов логически складывается. На выходе X появляется состояние логической 1 в том случае, если по меньшей мере состояние одного входа равно 1.

Три рассматриваемых столбца подчеркнуты на рис. 3.5 серыми полосами. Для случая 1 $X = 1$, так как $\bar{A} = 1$ и $\bar{B} = 1$. Также $X = 1$ в случае 2, так как здесь $\bar{B} = 1$. Пройдя все варианты, мы устанавливаем, что только в случае 4 X имеет состояние 0. Во всех других случаях $X = 1$.

Следующий столбец в таблице истинности предназначен для Y. Y является результатом логического умножения B и C. Теперь рассматриваем только столбцы B и C. На рис 3.5 они подчеркнуты черным. Y будет равен 1 только в случаях, в которых как B, так и C имеют состояние 1. Это варианты 7 и 8.

Y в свою очередь является входом элемента V, и этот элемент является вентилем НЕ. Состояние Y должно инвертироваться. Выход элемента V обозначим как $\bar{Y}$. Для $\bar{Y}$ предусмотрен свой столбец в таблице истинности, заполненный соответствующими состояниями.

Элемент VI является вентилем И с входами X и $\bar{Y}$. То есть состояния X и $\bar{Y}$ подвергаются операции логического умножения. Соответствующие колонки на рис. 3.5 обозначены заштрихованными полосами. Выход $Z = 1$, только если $X = 1$ и $\bar{Y} = 1$. Это происходит в случаях 1, 2, 3, 5 и 6.

Для Z можно записать уравнение алгебры логики, которое образуется из:

$$Z = X \wedge \bar{Y},\ X = \bar{A} \vee B \vee C;$$

$$\bar{Y} = \overline{B \wedge C}\ (\text{так как } Y = B \wedge C);$$

$$Z = \left(\bar{A} \vee \bar{B} \vee C\right) \wedge \overline{B \wedge C}. \quad \bullet$$

Это уравнение отражает логическую функцию схемы.

Рекомендуется потренироваться в составлении таблиц истинности по заданиями из разд. 3.4.

3.2. Логические функции и цифровые схемы

3.2.1. Определение логической функции цифровой схемы

Логическая функция цифровой схемы может быть записана в виде таблицы истинности. Отдельные шаги при составлении таблицы истинности ведут к итоговому уравнению для выхода всей схемы, в котором присутствуют только состояния входов или их инвертированные состояния (см. подразд. 2.1.2). Такое уравнение выражает функцию всей схемы. Поэтому оно называется **логической функцией цифровой схемы**.

> Для каждой цифровой схемы существует логическая функция.

Уравнение функции может быть найдено из анализа цифровой схемы. Для его составления не требуется восстанавливать таблицу истинности.

Цифровая схема (рис. 3.6) состоит из элементов с *I* по *V*. Если обозначить вход элемента *I* как *A*, то на выходе получится $\bar{A}$. Входы элемента *II* обозначим *A* и *C*. Выход обозначим как *M*:

$$M = A \wedge C.$$

Входами элемента *III* являются $\bar{A}$, *B* и *C*. Выход обозначим как *K*:

$$K = \bar{A} \wedge B \wedge C.$$

M и *K* являются входами элемента *IV*, над которыми производится логическое сложение. Выход элемента *IV* обозначим *X*:

$$X = K \vee M.$$

В это уравнение могут быть подставлены уже известные выражения для *K* и *M*:

$$X = \overbrace{\quad K \quad} \vee \overbrace{\quad M \quad}$$

$$X = \overbrace{\left(\bar{A} \wedge B \wedge C\right)} \vee \overbrace{\left(A \wedge C\right)}.$$

X является также входом элемента *V*. Так как элемент *V* инвертирует состояние *X*, то

$$Z = \bar{X}$$

и

$$\boxed{Z = \overline{\left(\bar{A} \wedge B \wedge C\right) \vee \left(A \wedge C\right)}}$$

так как $X = \left(\bar{A} \wedge B \wedge C\right) \vee \left(A \wedge C\right)$.

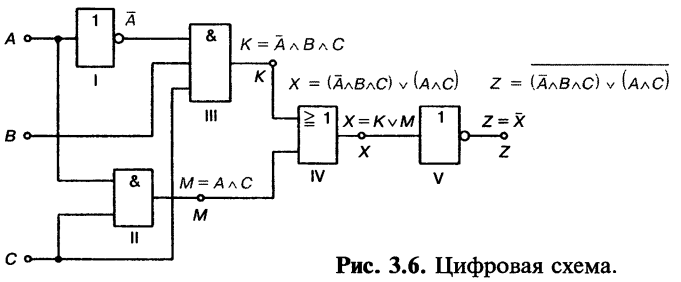


Рис. 3.6. Цифровая схема.

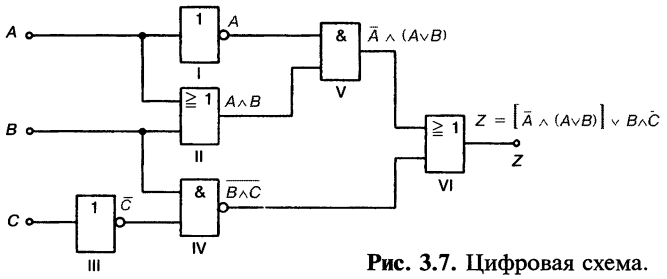


Рис. 3.7. Цифровая схема.

Уравнение для Z является искомой логической функцией схемы.

Если пройти еще раз по цепочке рассуждений, то можно не употреблять обозначения K, M и X в рассмотренном примере. Входы можно также обозначать, например как $(A \wedge C)$, то есть в виде логического выражения.

Найдем уравнение логической функции для схемы рис. 3.7.

На выходе элементов записываются логические выражения, то есть на выходе элемента $I - \bar{A}$, на выходе элемента $II - A \vee B$ и т. д. Эти выражения являются одновременно входами следующих элементов. Элемент IV имеет входы $\bar{C}$ и B. На выходе после операции логического сложения оказывается состояние $B \wedge \bar{C}$. Так как элемент IV является вентилем И-НЕ, то все выражение еще раз инвертируется, так что на выходе получается $\overline{B \wedge \bar{C}}$.

На выходе элемента V будет состояние $\bar{A} \wedge (A \vee B)$, так как его входами являются $\bar{A}$ и $A \vee B$.

Выражения, относящиеся друг к другу, должны всегда ставиться в скобки. В гл. 4 («Алгебра логики») говорится, что согласно правилам операция логического умножения И имеет более высокий приоритет, чем операция логического сложения ИЛИ, то есть выполняется первой. Поэтому на всякий случай нужно всегда ставить скобки — по крайней мере первое время. Линия инверсии рассматривается и действует так же, как скобки.

Вход элемента VI обозначен выражением $\bar{A} \wedge (A \vee B)$. Это выражение должно быть заключено в скобки. Второй вход элемента VI обозначен выражением $\overline{B \wedge \bar{C}}$. Здесь скобки не требуются, так как общая черта отрицания их заменяет.

Для Z получается следующее уравнение:

$$Z = \left[\bar{A} \wedge (A \vee B)\right] \vee \overline{B \wedge \bar{C}}.$$

Это уравнение является искомой логической функцией цифровой схемы.

Предположим, что элементом VI на рис. 3.7 будет элемент ИЛИ-НЕ. Как будет тогда выглядеть логическая функция схемы? Найденная для Z функция вся подвергается инверсии (отрицанию). Это указывается чертой отрицания над всем выражением:

$$Z = \overline{\left[\bar{A} \wedge (A \vee B)\right] \vee \overline{B \wedge \bar{C}}}.$$

Получившееся уравнение выглядит сложнее, чем является в действительности.

3.2.2. Синтез схемы по заданной логической функции

На практике часто бывает, что логическая функция получается в результате вычислений и требуется синтезировать схему, реализующую данную функцию.

Для уравнения

$$Z = \overline{A \vee \bar{B} \vee \bar{C}} \vee \left(A \vee \bar{C}\right)$$

требуется синтезировать цифровую схему.

Сначала нужно определить из уравнения число входов. Входами являются A, B и C. Для получения $\bar{B}$ и $\bar{C}$ необходимы два элемента НЕ (рис. 3.8). Выражение $A \vee \bar{B} \vee \bar{C}$ получают при помощи элемента ИЛИ с тремя входами. К этому элементу последовательно подключается элемент НЕ.

Для $(A \vee \bar{C})$ требуется элемент ИЛИ с двумя входами. Выходы с состояниями $\overline{A \vee \bar{B} \vee \bar{C}}$ и $(A \vee \bar{C})$ поступают на вход элемента ИЛИ.

Вместо элемента ИЛИ с тремя входами и последующим вентилем НЕ можно использовать элемент ИЛИ-НЕ с тремя входами (рис. 3.9).

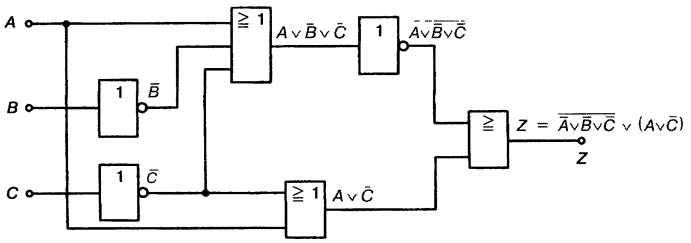


Рис. 3.8. Схема для заданного уравнения.

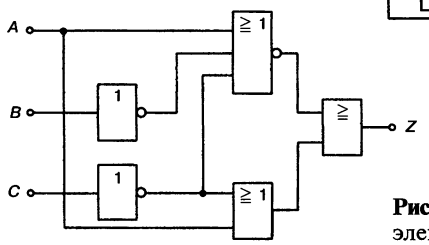


Рис. 3.9. Схема для заданного уравнения с применением элемента ИЛИ-НЕ.

3.3. Требуемая функция и реальная функция

Под **требуемой функцией** понимают логическую функцию, которую должна реализовывать схема на основе своей структуры. **Реальной функцией** является функция, которую на самом деле реализует схема в результате своей работы.

> При безупречно функционирующей схеме требуемая функция и реальная функция должны совпадать.

Если требуемая функция и реальная функция неодинаковы, то схема содержит одну или несколько ошибок, которые должны быть найдены и устранены.

3.3.1. Как определить реальную логическую функцию

Требуемая логическая функция определяется из заданной для схемы таблицы истинности. Реальная логическая функция определяется путем тестовых измерений.

> Перед началом измерений нужно установить соответствие уровней напряжения логическим состояниям 0 и 1.

Для определения логического состояния, или уровня напряжения, что является одним и тем же, требуется так называемый логический тестер. Он представляет собой маленький транзисторный усилитель и один или два диода для индикации показаний.

Прибор с одним индикаторным диодом показывает только логический уровень 1, или ВЫСОКИЙ уровень (для данного примера +5 В). Если диод не светится, значит, логический уровень равен 0. Такой прибор не позволяет обнаружить обрыв провода с логическим состоянием 0. Состояние 0 в большинстве случаев соответствует цифровой земле, а не неподключенному проводнику.

Лучше применять приборы с двумя светодиодами. Красный диод показывает, например, логическое состояние 1, а зеленый — логическое состояние 0. Если не горит ни один диод, значит в линии обрыв. Такой логический тестер на маленьком транзисторе можно собрать самому. Вход должен быть как можно более высокоомным.

Наряду с малыми логическими тестерами в продаже имеются более сложные приборы, позволяющие одновременно испытывать все входы и выходы. Вследствие этого значительно уменьшается время тестирования схемы. Последним звеном эволюции тестеров стали компьютеризированные тестовые стенды, которые автоматически тестируют всю схему и указывают дефекты.

Покажем на примере, как определяется реальная логическая функция.

Предположим, что имеется в наличии схема, изображенная на рис. 3.10. И у нас есть простой логический тестер, определяющий состояния 1 и 0 или соответственно ВЫСОКИЙ и НИЗКИЙ уровни.

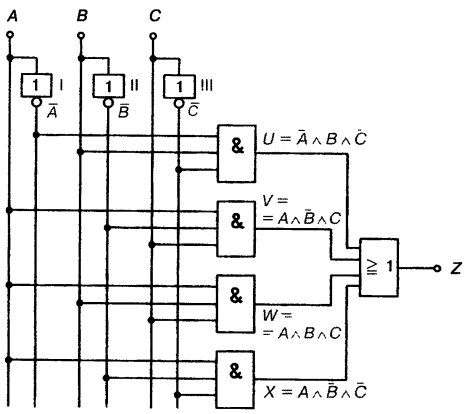


Рис. 3.10. Цифровая схема.

Присвоим логическим состояниям уровни напряжения:

> 0 ≙ 0 В (цифровая земля)
>
> 1 ≙ +5 В

Протокол измерений представим в виде таблицы (рис. 3.11).

Сначала измеряется вариант 1. Все три входа заземляются и получают состояние 0.

Затем тестируются выходы отдельных элементов — $\bar{C}$, $\bar{B}$, $\bar{A}$, U, V, W, X, Z — и заносятся в таблицу (обозначено серым на рис. 3.11).

Рис. 3.11. Таблица истинности.

Вар.	C	B	A	$\bar{C}$	$\bar{B}$	$\bar{A}$	$U = \bar{A} \wedge B \wedge \bar{C}$	$V = A \wedge \bar{B} \wedge C$	$W = A \wedge B \wedge C$	$X = A \wedge \bar{B} \wedge \bar{C}$	Z
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	1	1
3	0	1	0								
4	0	1	1								
5	1	0	0								
6	1	0	1								
7	1	1	0								
8	1	1	1								

После первого варианта измеряется вариант 2. На вход A подается 1, B и C остаются 0.

Снова измеряются выходы отдельных элементов, и их состояния заносятся в таблицу.

Это измерение производится для всех 8 возможных вариантов.

> Полученная таблица отражает реальную логическую функцию, реализуемую схемой.

Если предполагается, что схема исправна, то можно ограничиться определением состояния выхода Z. Если состояние Z отличается от требуемого, то проводится тест всех элементов.

3.3.2. Поиск дефекта схемы

Имея заданную таблицу истинности и таблицу протокола измерений, можно по их расхождению определить ошибку в схеме.

> Определение ошибки происходит путем сравнения требуемой логической функции и реальной.

Если требуемая логическая функция и реальная совпадают, то схема не содержит ошибок. Сначала сравнивают состояния выходов всей схемы. Если они совпадают, дальнейшее сравнение не требуется. Схема в порядке. Если состояния выходов различаются, то проводится пошаговый тест всех элементов, начиная от входных.

На рис. 3.12 изображена таблица истинности схемы по рис. 3.10 и таблица измерений.

Какие логические элементы работают неправильно?

Если проверять столбцы слева направо, то при $\bar{B}$ обнаруживается ошибка.

Это элемент НЕ, который должен инвертировать состояние входа B (элемент *II*), всегда имеет логическое состояние 1. Итак, он не работает.

Ошибка элемента *II* действует на выход V и X, так как только эти элементы используют сигнал $\bar{B}$. Для V и X логические операции выполнены верно при условии, что $\bar{B}$ всегда равен 1. Элементы V и X, следовательно, исправны.

Следующая ошибка обнаруживается на элементе W. Следовательно, элемент W неисправен. Ошибку при выполнении логической операции нельзя списать на счет элемента *II*, так как $W = A \wedge B \wedge C$ не включает в себя $\bar{B}$.

В результате проверки выяснилось, что замене подлежат логические элементы *II* и W.

Вар.	C	B	A	$\bar{C}$	$\bar{B}$	$\bar{A}$	$U = \bar{A} \wedge B \wedge \bar{C}$	$V = A \wedge \bar{B} \wedge C$	$W = A \wedge B \wedge C$	$X = A \wedge \bar{B} \wedge \bar{C}$	Z
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	**1**	**1**
3	0	1	0	1	0	1	**1**	0	0	0	**1**
4	0	1	1	1	0	0	0	0	0	0	0
5	1	0	0	0	1	1	0	0	0	0	0
6	1	0	1	0	1	0	0	**1**	0	0	**1**
7	1	1	0	0	0	1	0	0	0	0	0
8	1	1	1	0	0	0	0	0	**1**	0	**1**

Таблица истинности

Вар.	C	B	A	$\bar{C}$	$\bar{B}$	$\bar{A}$	$U = \bar{A} \wedge B \wedge \bar{C}$	$V = A \wedge \bar{B} \wedge C$	$W = A \wedge B \wedge C$	$X = A \wedge \bar{B} \wedge \bar{C}$	Z
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	**1**	**1**
3	0	1	0	1	[1]	1	**1**	0	0	0	**1**
4	0	1	1	1	[1]	0	0	0	[1]	[1]	[1]
5	1	0	0	0	1	1	0	0	0	0	0
6	1	0	1	0	1	0	0	**1**	0	0	**1**
7	1	1	0	0	[1]	1	0	0	0	0	0
8	1	1	1	0	[1]	0	0	[1]	**1**	0	**1**

Протокол измерений

Рис. 3.12. Таблица истинности и протокол измерений цифровой схемы.

Контрольный тест

1. Составьте таблицу истинности для схемы на рис. 3.13.

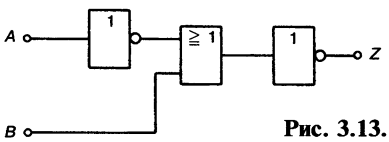


Рис. 3.13. Цифровая схема.

2. Составьте таблицу истинности для схемы рис. 3.14.

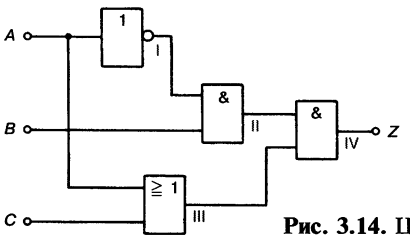


Рис. 3.14. Цифровая схема.

3. В схеме на рис. 3.14 элемент *II* дефектен. На его выходе детектируется всегда логическое состояние 1. Какую логическую функцию выполняет схема вследствие этого дефекта? Изобразите реальную функцию схемы в виде таблицы истинности.

4. Определите для схемы на рис. 3.15 логическую функцию и изобразите таблицу истинности.

5. Постройте схему, отвечающую следующей логической функции:

$$Z = \bar{A} \wedge B \wedge \overline{\overline{\bar{A} \wedge B \wedge C}}.$$

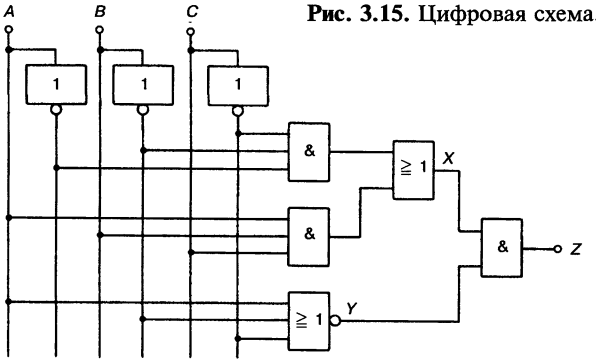


Рис. 3.15. Цифровая схема.

6. Постройте схему и таблицу истинности, отвечающую следующей логической функции:

$$Z = \overline{\overline{A} \vee B \vee C} \wedge \overline{\overline{A \vee \overline{B} \wedge \overline{C} \wedge D} \vee \overline{A \wedge D}}.$$

7. Схема на рис. 3.16 работает с ошибками. Таблица результатов тестирования (протокол измерений) изображена на рис. 3.17. Определите дефектные логические элементы.

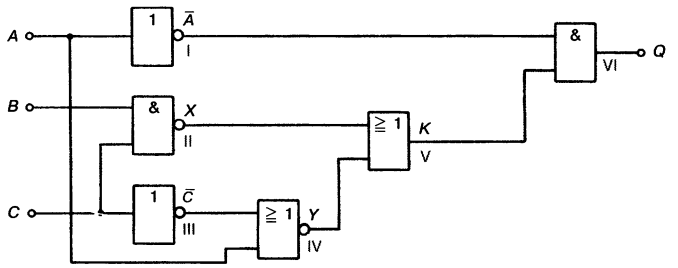


Рис. 3.16. Цифровая схема, работающая с ошибками.

Вар.	C	B	A	Ā	C̄	X	Y	K	Q
1	0	0	0	1	1	1	1	1	1
2	0	0	1	0	1	1	1	1	0
3	0	1	0	1	1	1	1	1	1
4	0	1	1	0	1	1	0	1	0
5	1	0	0	1	0	1	1	1	1
6	1	0	1	0	0	1	0	1	0
7	1	1	0	1	0	0	1	1	1
8	1	1	1	0	0	0	0	0	0

Рис. 3.17. Протокол измерений.

ГЛАВА 4

АЛГЕБРА ЛОГИКИ

Если с помощью цифровых схем хотят добиться выполнения определенных законов управления или вычислений, то необходимо найти схемы, которые «могут» реализовать задуманное. Схемы для простых заданий можно найти путем подбора (см. гл. 5). Однако чем выше требования к схеме, тем меньше вероятность ее нахождения методом подбора.

Даже если после длительных стараний удается найти подходящую схему, она обычно оказывается избыточно функциональной, и ее применение экономически нецелесообразно. Найти простую, но идеально подходящую схему методом подбора невозможно.

Для анализа и синтеза цифровых схем служит разработанный английским математиком Булем (1815—1864) математический аппарат, который изучается в средней школе в курсе «Теория множеств». Специальным разделом булевой алгебры является алгебра логики.

4.1. Переменные и постоянные величины (константы)

В алгебре логики, как и в обычной алгебре, есть понятия *переменных* и *постоянных* величин (констант). Но в алгебре логики константы могут иметь только два значения, а именно 0 или 1. Любая переменная в алгебре логики равна либо 0, либо 1.

> В алгебре логики есть только две константы: 0 и 1.

Эти константы соответствуют логическим состояниям 0 и 1.

Каждая величина, которая может принимать значение 0 или значение 1, представляет собой переменную величину. Входные величины схемы, например A, B, C, являются переменными величинами, так как они могут иметь логические состояния 1 или 0. Также выходные величины схемы являются переменными величинами. Выражения вида $(A \wedge B)$, состоящие из двух переменных величин, также являются величинами переменными, так как могут быть тоже равны только 0 или 1.

> Переменными алгебры логики являются величины, которые могут иметь состояния 0 или 1.

Следовательно, переменная алгебры логики является бинарной величиной. Ее можно наглядно изобразить в виде выключателя (рис. 4.1). Условимся, что

> Разомкнутому ключу соответствует логическое состояние 0.
> Замкнутому ключу соответствует логическое состояние 1.

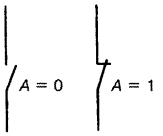


Рис. 4.1. Представление возможных состояний переменной *A*.

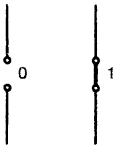


Рис. 4.2. Представление констант 0 и 1.

Это схемотехническое представление переменных величин очень просто для понимания. Можно ли так же просто изобразить графически постоянные величины? Можно понимать постоянные величины как «фиксированные переключатели». Если переключатель является фиксированным в разомкнутом состоянии, то он никогда не сможет замкнуться и всегда имеет значение 0. Если переключатель является фиксированным в замкнутом состоянии, то он никогда не сможет разомкнуться и всегда имеет значение 1.

Постоянно разомкнутый переключатель можно рассматривать как разрыв в линии. Постоянно замкнутый переключатель можно рассматривать как обычный провод (рис. 4.2).

Обрыв линии: 0 Неразрывное соединение: 1

4.2. Законы алгебры логики

Основные законы алгебры логики являются правилами, также называемыми еще аксиомами, действующими для логических операций над постоянными величинами.

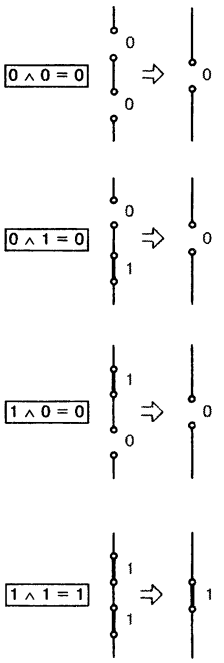


Рис. 4.3. Правило логического умножения И.

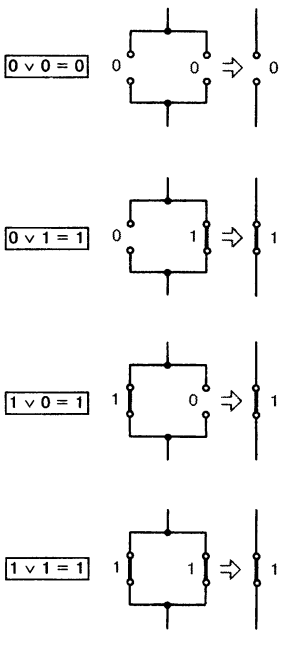


Рис. 4.4. Правило логического сложения ИЛИ.

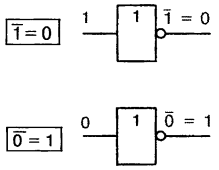


Рис. 4.5. Правило логического отрицания (инверсии) НЕ.

Закон для операции логического умножения И показан на рис. 4.3. Схематично логическое умножение И можно изобразить в виде последовательного соединения ключей. Логическая единица на выходе будет только в том случае, если оба ключа одновременно замкнуты (рис. 4.3). Иначе результат умножения равен нулю.

Закон для операции логического сложения ИЛИ изображен на рис. 4.4. Для логического сложения ключи в схеме рис. 4.4 включены параллельно. На выходе будет логическая 1 тогда, когда хотя бы один ключ находится в состоянии 1.

В логическом элементе НЕ единица на входе превращается в 0 на выходе, а 1 на входе — в 0 на выходе (рис. 4.5).

4.3. Аксиомы и тождества алгебры логики

4.3.1. Аксиомы

Правила для логической операции переменной величины с константой или переменной величины с самой собой или ее инвертированным значением называются **аксиомами**.

Обозначим некоторую переменную величину как A. Все, что верно для A, верно и для любой другой переменной величины. На рис. 4.6 изображены четыре возможные аксиомы логического умножения. Для представления инверсии $\bar{A}$ применен нормально-замкнутый контакт. Он замкнут, если главный выключатель разомкнут. И размыкается при замыкании главного выключателя. Таким образом, при $A \wedge \bar{A}$ один из последовательно включенных ключей всегда разомкнут и в линии имеет место разрыв (0).

Представление аксиом алгебры логики в виде простых схем очень наглядно. Также аксиомы можно изобразить в виде таблиц истинности (рис. 4.7).

Аксиомы для логической операции сложения ИЛИ следуют из рис. 4.8. Операцию ИЛИ можно изобразить в виде параллельного включения контактов.

Если переменная инвертируется и затем еще раз инвертируется, то она принимает первоначальное значение (рис. 4.9). Два штриха инверсии над переменной не меняют ее состояния.

Девять аксиом пронумерованы от 1 до 9. Под этими номерами они далее приводятся в сборнике формул.

4.3.2. Законы коммутативности и ассоциативности

Закон коммутативности еще называют **переместительным законом**. Он применяется для логического сложения и умножения и интуитивно понятен из схем на рис. 4.10 и 4.11.

> Результат операции логического умножения И не зависит от порядка обработки переменных.

> Результат операции логического сложения ИЛИ не зависит от порядка обработки переменных.

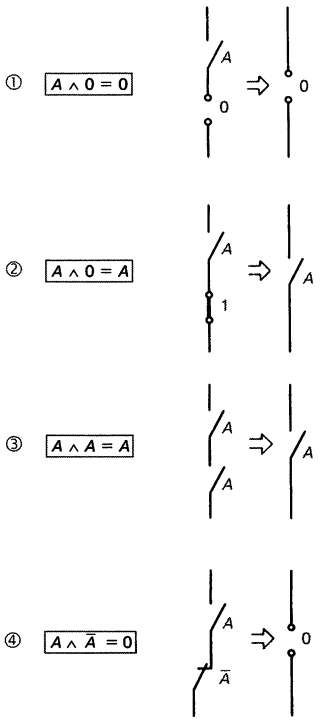


Рис. 4.6. Аксиомы логического умножения И.

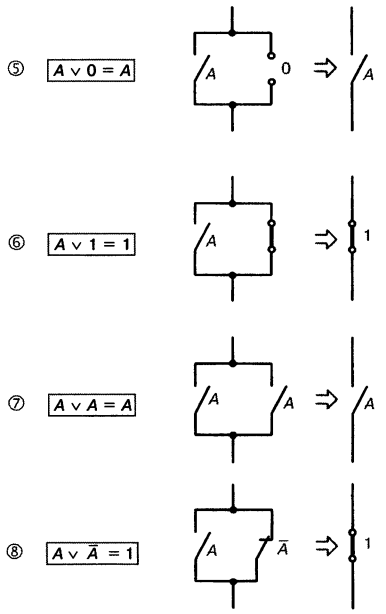


Рис. 4.8. Аксиомы логического умножения ИЛИ.

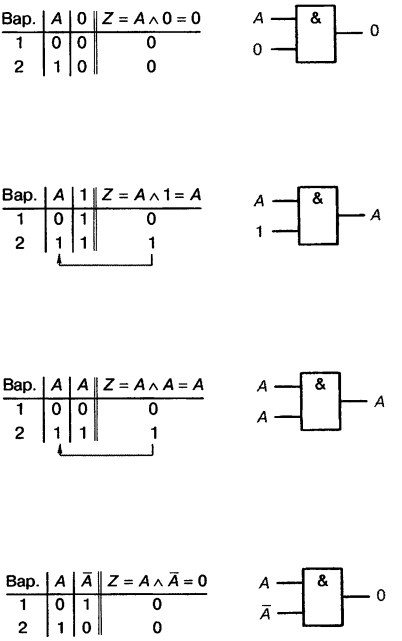


Рис. 4.7. Таблицы истинности для аксиом логического умножения И.

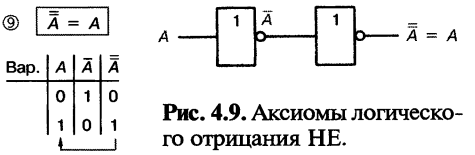


Рис. 4.9. Аксиомы логического отрицания НЕ.

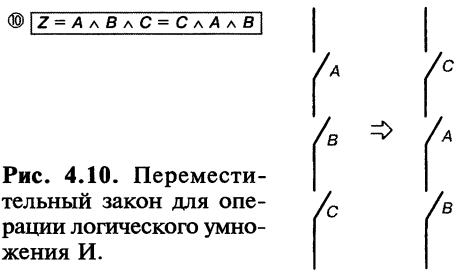


Рис. 4.10. Переместительный закон для операции логического умножения И.

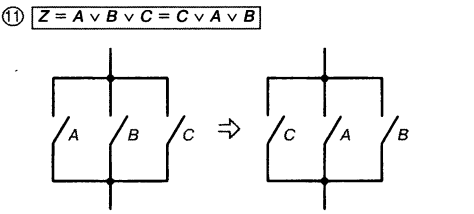
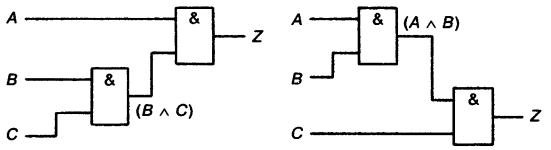
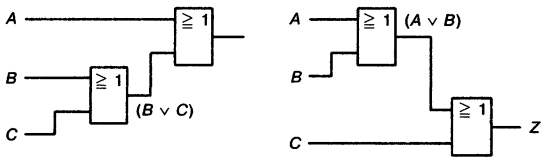


Рис. 4.11. Переместительный закон для операции логического сложения ИЛИ.



⑫ $Z = A \wedge (B \wedge C) = (A \wedge B) \wedge C$

Рис. 4.12. Сочетательный закон для операции логического умножения И.



⑬ $Z = A \vee (B \vee C) = (A \vee B) \vee C$

Рис. 4.13. Сочетательный закон для операции логического сложения ИЛИ.

Закон ассоциативности еще называют **сочетательным законом**. Он применяется для логического умножения (рис. 4.12) и сложения (рис. 4.13).

Результат операции логического умножения И не зависит от порядка обработки переменных.

Результат операции логического сложения ИЛИ не зависит от порядка обработки переменных.

4.3.3. Дистрибутивный закон

Дистрибутивный закон также называют **распределительным законом**. Распределительный закон логического умножения по отношению к сложению играет большую роль на практике при преобразовании логических выражений.

Различают конъюнктивный распределительный закон и дизъюнктивный распределительный закон. Конъюнктивный распределительный закон записывается как

⑭ $$Z = A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C).$$

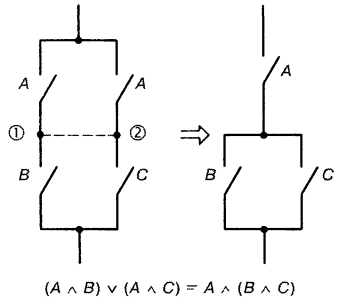
Переменная A в операции логического умножения И «распределяется» по переменным B и C. Схема на рис. 4.14 доказывает правильность этого тождества. Так как оба контакта A могут коммутироваться только одновременно, узлы 1 и 2 можно соединить без изменения действия схемы.

Чтобы еще лучше пояснить этот закон, тождество проверяется таблицей истинности (рис. 4.15). Состояния переменных в колонках X и Y одинаковы. Значит, конъюнктивный распределительный закон верен.

Дизъюнктивный распределительный закон записывается как

⑮ $$Z = A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C).$$

Переменная A в операции логического сложения ИЛИ «распределяется» по переменным B и C. Схема на рис. 4.16 доказывает правильность этого тождества. Так как оба контакта A могут коммутироваться только



$(A \wedge B) \vee (A \wedge C) = A \wedge (B \wedge C)$

Рис. 4.14. Конъюнктивный распределительный закон.

Вар.	C	B	A	A∧B	A∧C	(A∧B)∨(A∧C) (X)	B∨C	A∧(B∨C) (Y)
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	1	0	0	0	0	1	0
4	0	1	1	1	0	**1**	1	**1**
5	1	0	0	0	0	0	1	0
6	1	0	1	0	1	**1**	1	**1**
7	1	1	0	0	0	0	1	0
8	1	1	1	1	1	**1**	1	**1**

$(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C)$

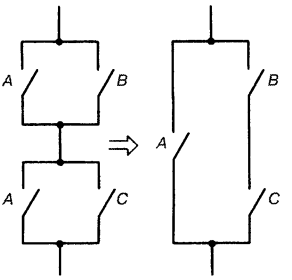
Рис. 4.15. Проверка правильности конъюнктивного распределительного закона при помощи таблицы истинности.

одновременно, схему можно преобразовать, как это изображено на рис. 4.16, без изменения действия схемы.

Советуем самостоятельно проверить последнее тождество таблицей истинности аналогично таблице на рис. 4.15.

Покажем применение дизъюнктивного распределительного закона на примере. Упростим выражение

$$Z = \left(K \vee \bar{M}\right) \wedge \left(K \vee M\right).$$



$(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C)$

Рис. 4.16. Дизъюнктивный распределительный закон.

Согласно дизъюнктивному распределительному закону оно преобразуется:

$$\begin{array}{c} (A \vee B) \wedge (A \vee C) = A \vee (B \wedge C) \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ Z = (K \vee \bar{M}) \wedge (K \vee M) = K \vee (\bar{M} \wedge M). \end{array}$$

Выражение $M \wedge \bar{M}$ является логическим сложением переменной и ее инвертированного значения. По аксиоме 4 на рис. 4.6 это выражение дает результат 0:

$$\begin{array}{l} A \wedge \bar{A} = 0 \\ \downarrow \quad \downarrow \\ M \wedge \bar{M} = 0. \end{array}$$

Тогда для Z:

$$\begin{array}{l} Z = K \vee \left(\bar{M} \wedge M\right) \\ Z = K \vee 0 \\ \quad \uparrow \quad \uparrow \\ A \vee 0 = A. \end{array}$$

Согласно аксиоме 5 (рис. 4.8) выражение (переменная ∨ 0) равно переменной

$$\underline{Z = K}.$$

4.3.4. Теоремы де Моргана

Английский математик де Морган (1806—1871) дополнил аксиомы алгебры логики теоремами, названными в его честь. Теоремы де Моргана имеют большое практическое значение при упрощении инвертируемых выражений для логических операций с элементами И-НЕ и ИЛИ-НЕ. Существуют две теоремы де Моргана.

Первая теорема де Моргана:

⑯ $$\boxed{Z = \overline{A \wedge B} = \bar{A} \vee \bar{B}.}$$

Эта теорема доказывается с помощью таблицы истинности (рис. 4.17).

Вторая теорема де Моргана:

⑰ $$\boxed{Z = \overline{A \vee B} = \bar{A} \wedge \bar{B}.}$$

Согласно теоремам взаимно меняется тип логической операции (И и ИЛИ). Вторая теорема доказывается с помощью таблицы истинности (рис. 4.18).

Вар.	B	A	$A \wedge B$	$\overline{A \wedge B}$	$\bar{A}$	$\bar{B}$	$\bar{A} \vee \bar{B}$
1	0	0	0	1	1	1	1
2	0	1	0	1	0	1	1
3	1	0	0	1	1	0	1
4	1	1	1	0	0	0	0

$\overline{A \wedge B} = \bar{A} \vee \bar{B}$

Рис. 4.17. Таблица истинности для доказательства первой теоремы Моргана.

Вар.	B	A	$A \vee B$	$\overline{A \vee B}$	$\bar{A}$	$\bar{B}$	$\bar{A} \wedge \bar{B}$
1	0	0	0	1	1	1	1
2	0	1	1	0	0	1	0
3	1	0	1	0	1	0	0
4	1	1	1	0	0	0	0

$\overline{A \vee B} = \bar{A} \wedge \bar{B}$

Рис. 4.18. Таблица истинности для доказательства второй теоремы Моргана.

Покажем важность теорем де Моргана на примере. С их помощью можно значительно упростить выражение:

$$P = \overline{R \wedge S} \vee \overline{\bar{R} \wedge S}.$$

Первая часть уравнения $\overline{R \wedge S}$ согласно первой теореме преобразуется в $\bar{R} \vee \bar{S}$. Вторая часть уравнения $\overline{\bar{R} \wedge S}$ согласно той же теореме преобразуется в $\overline{\bar{R}} \vee \bar{S}$. $\overline{\bar{R}}$ согласно аксиоме 9 равно R.

$$P = \overline{R \wedge S} \vee \overline{\bar{R} \wedge S};$$

$$P = \bar{R} \vee \bar{S} \vee \overline{\bar{R}} \vee \bar{S};$$

$$P = \bar{R} \vee \bar{S} \vee R \vee \bar{S}.$$

Последовательность переменных изменена. Проведем преобразования по аксиомам 8, 7 и 6:

$$P = \underbrace{\bar{R} \vee R}_{⑧\ \bar{A} \vee A = 1} \vee \underbrace{\bar{S} \vee \bar{S}}_{⑦\ A \vee A = A}$$

$$P = 1 \vee \bar{S}$$

⑥ $\boxed{1 \vee A = 1}$

$$P = 1$$

Теоремы де Моргана действуют также и для логических операций с большим количеством переменных:

$$\boxed{Z = \overline{A \wedge B \wedge C \wedge D \wedge \ldots} = \bar{A} \vee \bar{B} \vee \bar{C} \vee \bar{D} \vee \ldots}$$

$$\boxed{Z = \overline{A \vee B \vee C \vee D \vee \ldots} = \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge D \wedge \ldots}$$

Задание

Составьте таблицу истинности и проверьте истинность теоремы де Моргана для трех переменных.

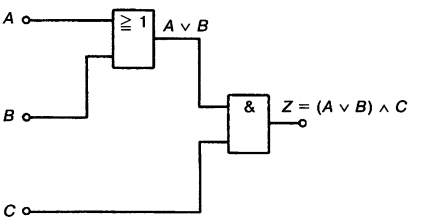
4.3.5. Приоритеты логических операций

Логическая операция И и ИЛИ с несколькими переменными может привести к неоднозначности. Уравнение

$$Z = A \vee B \wedge C$$

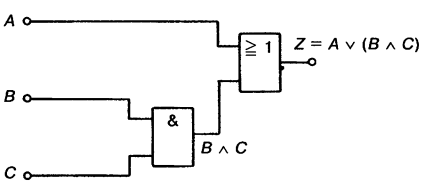
может быть решено двумя способами. Можно сначала сложить переменные A и B, затем умножить результат на C. Соответствующая схема и таблица истинности изображены на рис. 4.19.

С другой стороны, можно сначала перемножить переменные B и C, а результат сложить с A. Соответствующая схема и таблица истинности изображены на рис. 4.20. Z в этих двух вариантах является результатом абсолютно разных логических операций. Неоднозначность можно устранить с помощью скобок. В первом случае нужно писать $Z = (A \vee B) \wedge C$. Во втором случае — $Z = A \vee (B \wedge C)$.



Вар.	C	B	A	A ∨ B	Z
1	0	0	0	0	0
2	0	0	1	1	0
3	0	1	0	1	0
4	0	1	1	1	0
5	1	0	0	0	0
6	1	0	1	1	1
7	1	1	0	1	1
8	1	1	1	1	1

Рис. 4.19. Схема и таблица истинности для функции $Z = (A \vee B) \wedge C$.



Вар.	C	B	A	B ∧ C	Z
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	0
4	0	1	1	0	1
5	1	0	0	0	0
6	1	0	1	0	1
7	1	1	0	1	1
8	1	1	1	1	1

Рис. 4.20. Схема и таблица истинности для функции $Z = A \wedge (B \vee C)$.

От скобок можно отказаться, если ввести приоритеты логических операций.

Логическая операция с более высоким приоритетом выполняется перед другими логическими операциями. Приоритет существует и в обычной алгебре. Умножение и деление имеют там более высокий приоритет перед сложением и вычитанием.

В алгебре логики более высокий приоритет имеет операция логического умножения И.

> Операция логического умножения И выполняется перед логическим сложением ИЛИ.

$$Z = A \vee B \wedge C \Longrightarrow A \vee (B \wedge C).$$

Теперь рассмотренное выше уравнение становится однозначным.

Если в выражении алгебры логики присутствуют операции логического умножения и сложения, то переменные, связанные логическим умножением, должны читаться так, как будто взяты в скобки.

4.4. Функции И-НЕ и ИЛИ-НЕ

Алгебра логики построена на трех основных логических операциях — И, ИЛИ и НЕ. На элементах, выполняющих эти операции, можно реализовать любую логическую функцию. Поэтому элементы И, ИЛИ и НЕ называются **основными**.

Из первой теоремы де Моргана следует, что элемент логического умножения всегда может быть заменен элементом ИЛИ и несколькими элементами НЕ:

$\overline{A \wedge B} = \overline{A} \vee \overline{B}$ — первая теорема де Моргана

$$\overline{\overline{A \wedge B}} = \overline{\overline{A} \vee \overline{B}};$$

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

Это означает, что без элементов И можно обойтись. Отсюда следует правило:

> Любая логическая функция может быть реализована только на элементах ИЛИ и НЕ.

Вентили ИЛИ и НЕ можно реализовать на элементах ИЛИ-НЕ (рис. 4.21). Если соединить входы элемента ИЛИ-НЕ вместе, то получится элемент НЕ. Элемент ИЛИ получается путем инвертирования выхода ИЛИ-НЕ. Для этого к выходу элемента ИЛИ-НЕ последовательно подключается еще один элемент ИЛИ-НЕ, который действует как элемент НЕ (рис. 4.21).

Вентиль И может быть образован согласно уравнению, следующему из первой теоремы де Моргана:

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

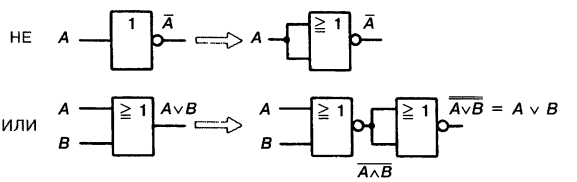


Рис. 4.21. Вентили ИЛИ и НЕ, реализованные на логических элементах ИЛИ-НЕ.

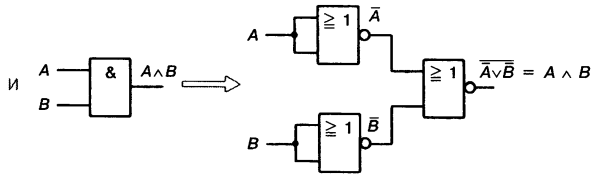


Рис. 4.22. Вентиль И, реализованный на логических элементах ИЛИ-НЕ.

Для получения $\bar{A}$ и $\bar{B}$ применяют два элемента ИЛИ-НЕ. Для логической операции сложения используется еще один элемент ИЛИ-НЕ (рис. 4.22).

Итак, если основные логические элементы ИЛИ, НЕ, И можно реализовать только на элементах ИЛИ-НЕ, то это значит, что любую возможную логическую функцию также можно реализовать только на элементах ИЛИ-НЕ.

> Любая логическая функция может быть реализована только на элементах ИЛИ-НЕ.

Элементы ИЛИ-НЕ можно использовать как универсальные логические элементы.

Из второй теоремы де Моргана следует, что логический элемент ИЛИ может быть заменен логическим элементом И и несколькими элементами НЕ:

$$\overline{A \vee B} = \bar{A} \wedge \bar{B}$$ — вторая теорема де Моргана

$$\overline{\overline{A \vee B}} = \overline{\bar{A} \wedge \bar{B}};$$

$$A \vee B = \overline{\bar{A} \wedge \bar{B}}.$$

Это означает, что без элементов ИЛИ можно обойтись. Отсюда следует правило:

> Любая логическая функция может быть реализована только на элементах И и НЕ.

Вентиль НЕ можно реализовать на элементах И-НЕ. Если соединить входы элемента И-НЕ вместе, то получится элемент НЕ. Элемент И получается путем последовательного включения элемента И-НЕ и еще одного элемента И-НЕ, который действует как элемент НЕ (рис. 4.23).

Вентили ИЛИ также можно реализовать на элементах И-НЕ. Из второй теоремы де Моргана следует:

$$A \vee B = \overline{\bar{A} \wedge \bar{B}}.$$

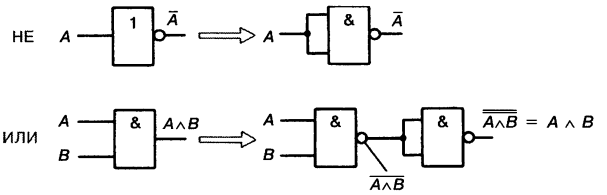


Рис. 4.23. Вентили НЕ и И, реализованные на логических элементах И-НЕ.

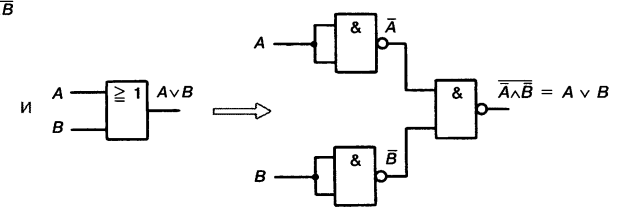


Рис. 4.24. Вентиль ИЛИ, реализованный на логических элементах И-НЕ.

Для реализации вентиля ИЛИ требуются два элемента И-НЕ, включенные как элементы НЕ. Еще один последовательно включенный элемент И-НЕ используется для логического умножения с инверсией (рис. 4.24).

Итак, если основные логические элементы ИЛИ, НЕ, И можно реализовать только на элементах И-НЕ, то любую возможную логическую функцию также можно реализовать только на элементах И-НЕ.

> Любая логическая функция может быть реализована только на элементах И-НЕ.

Элементы И-НЕ, так же как и элементы ИЛИ-НЕ, можно использовать как универсальные логические элементы.

Для синтеза цифровых схем только на элементах И-НЕ или ИЛИ-НЕ часто требуется многошаговое преобразование уравнений алгебры логики. Такие преобразования могут быть произведены разными способами. Путь, обычно ведущий к цели кратчайшей дорогой, начинается с операции двойного отрицания. Двойное отрицание не меняет результат выражения.

Пример

Преобразуйте уравнение $Z = (\bar{A} \wedge \bar{B} \wedge C) \vee (A \wedge \bar{B} \wedge \bar{C})$ так, чтобы соответствующая схема содержала только элементы И-НЕ.

$$Z = \overline{\overline{(\bar{A} \wedge \bar{B} \wedge C) \vee (A \wedge \bar{B} \wedge \bar{C})}};$$

$$Z = \overline{\overline{(\bar{A} \wedge \bar{B} \wedge C)} \wedge \overline{(A \wedge \bar{B} \wedge \bar{C})}}.$$

Синтезированная на основе преобразованного уравнения схема показана на рис. 4.25.

Пример

Преобразуйте уравнение $Z = (\bar{A} \wedge \bar{B} \wedge C) \vee (A \wedge \bar{B} \wedge \bar{C})$ так, чтобы соответствующая схема содержала только элементы ИЛИ-НЕ.

$$Z = \overline{\overline{(\bar{A} \wedge \bar{B} \wedge C)} \vee \overline{(A \wedge \bar{B} \wedge \bar{C})}};$$

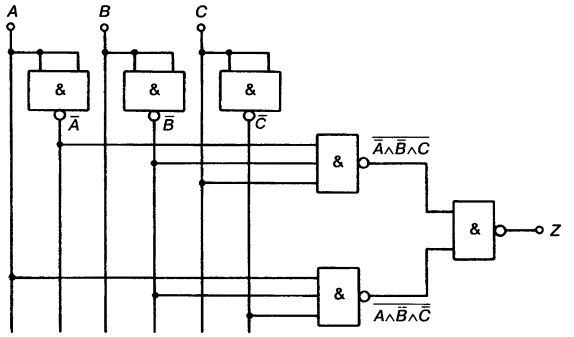


Рис. 4.25. Схема только на элементах И-НЕ.

$$Z = \overline{\left(\bar{\bar{A}} \wedge \bar{\bar{B}} \wedge \bar{C}\right)} \vee \overline{\left(\bar{A} \wedge \bar{\bar{B}} \wedge \bar{\bar{C}}\right)};$$

$$Z = \overline{\overline{\overline{(A \wedge B \wedge \bar{C})} \vee \overline{(\bar{A} \wedge B \wedge C)}}}.$$

Синтезированная на основе преобразованного уравнения схема показана на рис. 4.26.

На практике часто возникают трудности при преобразовании алгебраических уравнений в логические операции И-НЕ и ИЛИ-НЕ. Есть способ избежать этих преобразований. В схеме, построенной из основных элементов, отдельные основные элементы можно заменить элементами И-НЕ и ИЛИ-НЕ, как указано на рис. 4.21, 4.22, 4.23 и 4.24.

При этом получаются более сложные схемы, которые, тем не менее, легко упрощать. Так, например, часто встречаются два следующих друг за другом вентиля НЕ. Эти вентили можно вычеркнуть, так как их действия взаимно компенсируются (двойное инвертирование переменной не меняет ее). Вследствие этого схема значительно упрощается.

Пример такого упрощения показан на рис. 4.27. В верхней части рисунка представлена схема, соответствующая функции $Z = \left(\bar{A} \wedge \bar{B} \wedge C\right) \vee \left(A \wedge \bar{B} \wedge \bar{C}\right)$ и состоящая из основных элементов. Преобразуем эту схему в схему, состоящую только из элементов И-НЕ. Для этого каждый основной элемент

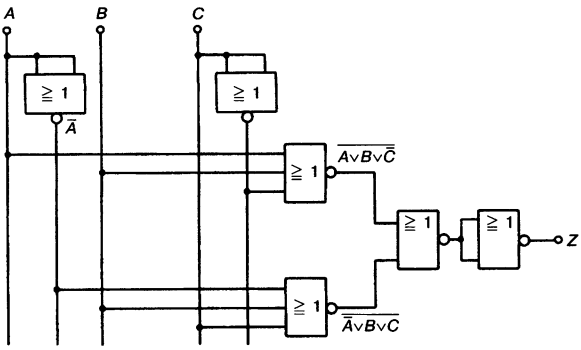


Рис. 4.26. Схема только на элементах ИЛИ-НЕ.

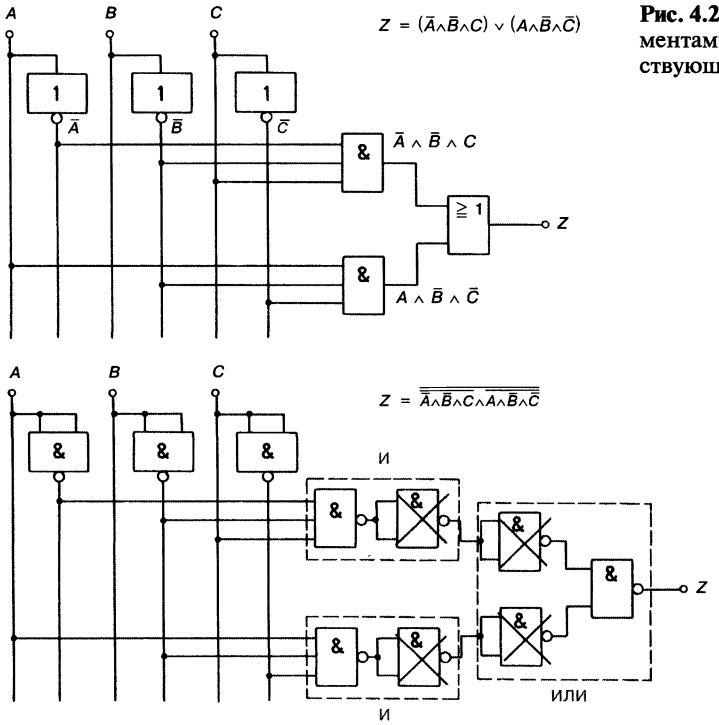


Рис. 4.27. Схема с основными элементами, замененными соответствующими элементами И-НЕ.

схемы заменяется соответствующим ему элементом И-НЕ. Последовательно включенные элементы НЕ можно вычеркнуть. В результате получается схема, функцию которой мы ранее определили расчетным путем (см. рис. 4.25). Этот способ всегда работает на практике, однако требует немного больше времени.

4.5. Примеры

Аксиомы и теоремы алгебры логики

Аксиомы

① $A \wedge 0 = 0$;

② $A \wedge 1 = A$;

③ $A \wedge A = A$;

④ $A \wedge \overline{A} = 0$;

⑤ $A \vee 0 = A$;

⑥ $A \vee 1 = 1$;

⑦ $A \vee A = A$;

⑧ $A \vee \overline{A} = 1$;

⑨ $\overline{\overline{A}} = A$.

Переместительный закон

⑩ $A \wedge B \wedge C = C \wedge A \wedge B$;

⑪ $A \vee B \vee C = C \vee A \vee B$.

Сочетательный закон

⑫ $A \wedge (B \wedge C) = (A \wedge B) \wedge C;$

⑬ $A \vee (B \vee C) = (A \vee B) \vee C.$

Распределительный закон

⑭ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C);$ ⑭а $A \wedge (A \vee B) = A;$

⑮ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C);$ ⑮а $A \vee (A \wedge B) = A.$

Теоремы де Моргана

⑯ $\overline{A \wedge B} = \bar{A} \vee \bar{B};$ ⑯а $\overline{A \wedge B \wedge C \wedge D \wedge \ldots} = \bar{A} \vee \bar{B} \vee \bar{C} \vee D \vee \ldots;$

⑰ $\overline{A \vee B} = \bar{A} \wedge \bar{B};$ ⑰а $\overline{A \vee B \vee C \vee D \vee \ldots} = \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \wedge \ldots.$

Уравнения 14а и 15а являются другой формой записи распределительного закона. Они получаются следующим образом:

$$A \wedge (A \vee B) = (A \vee 0) \wedge (A \vee B) = A \vee (0 \wedge B) = A \vee 0 = A$$

$$\uparrow \qquad\qquad \downarrow$$

$$A = A \vee 0 \qquad\qquad 0 \wedge B = 0$$

$$A \vee (A \wedge B) = (A \wedge 1) \vee (A \wedge B) = A \wedge (1 \vee B) = A \wedge 1 = A$$

$$\uparrow \qquad\qquad \downarrow$$

$$A = A \wedge 1 \qquad\qquad 1 \vee B = 1$$

Уравнения 16а и 17а являются расширением теорем де Моргана на любое число переменных.

В предыдущих уравнениях переменные A, B, C и D подразумевают, что на их месте может быть любая переменная. В качестве переменных могут также записываться выражения в скобках и функции многих переменных.

Примеры

$$Z = R \wedge 0 = 0; \qquad \text{(акс. 1)}$$

$$Z = \overline{S \wedge K} = \bar{S} \vee \bar{K}; \qquad \text{(уравн. 16)}$$

$$Z = (X \wedge Y) = \overline{X \wedge Y} = 1. \qquad \text{(акс. 8)}$$

В последнем уравнении речь идет об операции ИЛИ над переменной и ее инвертированным значением. Переменной является выражение $(X \wedge Y)$. Согласно уравнению 8 $\left(A \vee \bar{A} = 1\right)$ Z равно 1.

Примеры расчетов

Упрощение выражений

Пример 1

$$Z = \bar{A} \vee B \vee \bar{B} \vee C;$$

$$B \vee \bar{B} = 1; \qquad \text{(акс. 8)}$$

$$Z = \bar{A} \vee 1 \vee C;$$

$$Z = \left(\bar{A} \vee C\right) \vee 1.$$

Логическое сложение ИЛИ в результате дает 1 (акс. 1). Выражение в скобках $\left(\bar{A} \vee C\right)$ рассматривается как одна переменная:

$$\underline{Z = 1}.$$

Пример 2

$$X = \left(M \wedge \bar{N}\right) \vee \left(M \wedge N \wedge \bar{M}\right);$$

$$X = \left(M \wedge \bar{N}\right) \vee \left(M \wedge \bar{M} \wedge N\right);$$

$$M \wedge \bar{M} = 0; \qquad \text{(акс. 4)}$$

$$X = \left(M \wedge \bar{N}\right) \vee (0 \wedge N);$$

$$0 \wedge N = 0; \qquad \text{(акс. 1)}$$

$$X = \left(M \wedge \bar{N}\right) \vee 0.$$

Согласно аксиоме 5 логическое сложение переменной с нулем дает в результате саму переменную:

$$\underline{X = (M \wedge \bar{N})}. \qquad \text{(акс. 5)}$$

Пример 3

$$Z = B \vee \left(\bar{A} \wedge B \wedge C\right) \vee \bar{B}.$$

Порядок слагаемых при логическом сложении не имеет значения. Выражение $B \vee \bar{B}$ согласно аксиоме 8 равно 1:

$$Z = B \vee \bar{B} \vee \left(\bar{A} \wedge B \wedge C\right) = 1 \vee \left(\bar{A} \wedge B \wedge C\right).$$

Выражение в скобках рассматривается как одна переменная. Логическое сложение переменной с единицей дает в результате единицу (акс. 6):

$$\underline{Z = 1}.$$

Пример 4

$$Z = X \wedge \left(\bar{X} \vee S\right).$$

Согласно распределительному закону (уравн. 14), его можно «раскрыть»:

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C);$$
$$Z = X \wedge (\bar{X} \vee S) = (X \wedge \bar{X}) \vee (X \wedge S).$$

Выражение $X \wedge \bar{X}$ равно 0 (акс. 4):

$$Z = 0 \vee (X \wedge S).$$

Выражение в скобках $(X \wedge S)$ рассматривается как одна переменная. Логическое сложение переменной с нулем дает в результате переменную (акс. 5):

$$\underline{Z = X \wedge S}.$$

С помощью таблицы истинности можно проверить правильность расчетов (рис. 4.28).

Вар.	C	B	$\bar{B}$	$\bar{B} \vee C$	$Z = B \wedge (\bar{B} \vee C)$	$B \wedge C$
1	0	0	1	1	0	0
2	0	1	0	0	0	0
3	1	0	1	1	0	0
4	1	1	0	1	1	1

Рис. 4.28. Таблица истинности для проверки результата.

Пример 5

$$Z = A \vee \bar{B} \wedge \overline{A \vee \bar{B} \vee C}.$$

Сначала уберем длинную инвертирующую черту согласно второму закону де Моргана:

$$Z = A \vee \left(\bar{B} \wedge \bar{A} \wedge \bar{\bar{B}} \wedge \bar{C}\right);$$

$$Z = A \vee \left(\bar{B} \wedge \bar{A} \wedge B \wedge \bar{C}\right).$$

Поменяем переменные местами:

$$Z = A \vee \left(\bar{B} \wedge B \wedge \bar{A} \wedge \bar{A}\right);$$
$$\bar{B} \wedge B = 0; \qquad \text{(акс. 4)}$$

$$Z = A \vee \left(0 \wedge \bar{A} \wedge \bar{C}\right).$$

Выражение $\bar{A} \wedge \bar{C}$ рассматривается как одна переменная. Логическое умножение переменной с нулем дает в результате ноль:

$$Z = A \vee 0;$$

$$\underline{Z = A}. \qquad \text{(акс. 5)}$$

Пример 6

Сначала следует разбить длинную верхнюю черту отрицания на отрезки согласно уравнению 17. Двойные черты отрицания равной длины взаимно сокращаются и могут быть вычеркнуты (акс. 9). Черты отрицания над несколькими переменными равносильны скобкам:

$$Y = \overline{\overline{\overline{\overline{A \wedge X} \vee \overline{\overline{A} \wedge B \wedge X} \vee \overline{\overline{B} \wedge X}}}}.$$

Если они убираются, необходимо проверить, не надо ли вместо них поставить скобки. В этом примере скобки необходимы:

$$Y = \overline{\overline{\overline{\overline{A \wedge X} \vee \overline{\overline{A} \wedge B \wedge X} \vee \overline{\overline{B} \wedge X}}}};$$

$$Y = \left(\overline{\overline{A \wedge X} \vee \overline{\overline{A} \wedge B \wedge X}}\right) \wedge \overline{B} \wedge X.$$

Теперь можно разбить короткие черты отрицания и сортировать переменные:

$$Y = \left(\overline{A} \vee X \vee \overline{\overline{A}} \vee \overline{B} \vee \overline{X}\right) \wedge \overline{B} \wedge X;$$

$$Y = \left(\overline{A} \vee A \vee \overline{X} \vee \overline{X} \vee \overline{B}\right) \wedge \overline{B} \wedge X;$$

$$\overline{A} \vee A = 1; \qquad \text{(акс. 8)}$$

$$\overline{X} \vee \overline{X} = \overline{X}; \qquad \text{(акс. 7)}$$

$$Y = \left(1 \vee \overline{X} \vee \overline{B}\right) \wedge \overline{B} \wedge X;$$

$$1 \vee \left(\overline{X} \vee \overline{B}\right) = 1; \qquad \text{(акс. 6)}$$

$$Y = 1 \wedge \overline{B} \wedge X;$$

$$\underline{Y = \overline{B} \wedge X}.$$

Преобразование уравнений

Пример 7

Преобразуйте следующую функцию так, чтобы реализующая ее схема состояла только из элементов И-НЕ.

$$Z = \left[C \vee (N \wedge P \wedge S)\right] \wedge \left(\overline{A} \vee \overline{B}\right);$$

$$Z = \overline{\overline{\left[C \vee (N \wedge P \wedge S)\right]} \wedge \overline{\left(\overline{A} \vee \overline{B}\right)}};$$

$$Z = \overline{\overline{B} \wedge \overline{N \wedge P \wedge S}} \wedge \overline{\overline{\overline{A}} \wedge \overline{\overline{B}}};$$

$$Z = \overline{\overline{C} \wedge \overline{N \wedge P \wedge S}} \wedge \overline{A \wedge B};$$

$$Z = \overline{\overline{\overline{C} \wedge \overline{N \wedge P \wedge S}} \wedge \overline{A \wedge B}}.$$

Пример 8

Преобразуйте следующую функцию так, чтобы реализующая ее схема состояла только из элементов ИЛИ-НЕ.

$$X = \overline{A \wedge \bar{C}} \wedge \overline{\bar{B} \wedge \bar{R} \wedge \bar{S}};$$

$$X = \left(\bar{A} \vee \bar{\bar{C}}\right) \wedge \left(\bar{\bar{B}} \vee \bar{\bar{R}} \vee \bar{\bar{S}}\right);$$

$$X = \left(\bar{A} \vee C\right) \wedge \left(B \vee R \vee S\right);$$

$$X = \overline{\overline{\left(\bar{A} \vee C\right) \wedge \left(B \vee R \vee S\right)}};$$

$$X = \overline{\overline{\bar{A} \vee C} \vee \overline{B \vee R \vee S}};$$

$$X = \overline{\overline{\bar{A} \vee C} \vee \overline{B \vee R \vee S}}.$$

Контрольный тест

1. Назовите количество возможных постоянных величин в алгебре логики.
2. Как представляется переменная величина в алгебре логики? Какие логические операции можно производить с переменными?
3. Что такое переместительный закон?
4. Какое значение имеет распределительный закон?
5. Почему все возможные логические операции можно реализовать только на элементах И-НЕ?
6. Какая из двух логических операций имеет более высокий приоритет: И или ИЛИ?
7. Сформулируйте теоремы де Моргана.
8. Элементы И, ИЛИ или НЕ можно реализовать только на
   а) вентилях И-НЕ,
   б) вентилях ИЛИ-НЕ.

   Нарисуйте соответствующие схемы.
9. Максимально упростите следующие выражения:

   а) $Z = \bar{A} \wedge B \wedge A \wedge A \wedge B \wedge \bar{C}$;

   б) $Y = \overline{A \wedge B} \vee \bar{A} \vee \bar{C} \vee \overline{A \wedge \bar{B} \wedge C}$;

   в) $X = \left(\bar{A} \wedge B \wedge \bar{C}\right) \vee \left(\bar{A} \wedge B \wedge C\right)$;

   г) $Q = \bar{A} \vee \bar{B} \vee \bar{C} \vee \left(\bar{A} \wedge \bar{B} \wedge \bar{C}\right) \vee \left(A \wedge B\right) \vee \left(\bar{A} \wedge \bar{C}\right)$;

   д) $S = \overline{\overline{A \wedge B} \vee \overline{\bar{B} \wedge C} \vee \left(A \wedge B\right)}$.

10. Преобразуйте следующие функции так, чтобы реализующая их схема состояла

1) только из элементов И-НЕ;
2) только из элементов ИЛИ-НЕ:

а) $Z = (A \wedge S \wedge R) \vee (Q \wedge \overline{C} \wedge \overline{B})$;

б) $Y = \overline{A \vee B} \wedge \overline{C \vee D}$;

в) $X = (A \vee B \vee C) \wedge (M \vee N \vee P) \wedge (R \vee S)$;

г) $Q = \overline{(\overline{A} \wedge B) \vee \overline{C} \vee D} \wedge \overline{S \vee R}$;

д) $Q = \overline{A \wedge \overline{B} \wedge \overline{C} \wedge D \vee P \wedge Q \wedge S}$.

ГЛАВА 5

СИНТЕЗ СХЕМ

5.1. Синтез схем на логических элементах по заданным условиям

Цифровые электронные схемы на логических элементах применяются в качестве схем управления для самых различных задач контроля и регулирования технологических объектов. Под синтезом схемы понимают ее проектирование (разработку).

Перед началом синтеза схемы должна быть четко и однозначно сформулирована задача, которую должна решать схема. Словесные формулировки часто можно трактовать неоднозначно. Поэтому техническому заданию на разработку схемы требуется уделить особое внимание.

В первую очередь нужно назначить входные переменные. В качестве символов применяют заглавные буквы алфавита, начиная с первой, с индексом или без.

Входные переменные, например: A, B, C, D, E, F, G, E_1, E_2, E_3.

Затем назначают выходные переменные. В качестве символов для выходных сигналов применяют заглавные буквы алфавита, начиная с последней:

Выходные переменные, например: Z, Y, X, V_1, V_2, V_3.

Затем необходимо оговорить, при каких условиях переменные равны 1 и 0.

После этого можно приступать к составлению таблицы истинности. Сразу станет ясно, являлось ли словесное описание однозначным. Если в процессе составления таблицы встречается неясность, ее надо сразу устранять путем обсуждения с остальными разработчиками и заказчиками.

Таблица истинности однозначно определяет, как будет работать проектируемая схема.

После построения таблицы истинности подбирают логические элементы, на которых ее можно реализовать. Схема должна быть как можно проще и состоять из элементов, имеющихся в наличии.

Следует попытаться максимально упростить найденную схему. Если в наличии имеются, например, только элементы И-НЕ, схему надо преобразовать так, чтобы она состояла только из элементов И-НЕ.

Итак, для синтеза схемы можно выделить пять шагов:

1. Описание функции требуемой схемы.
2. Назначение входных и выходных переменных величин и присвоение значений 0 и 1.
3. Составление таблицы истинности.

4. Определение необходимых логических операций.
5. Упрощение и при необходимости преобразование схемы.

Рассмотрим этот пошаговый синтез на конкретном примере.

Пример

Требуется синтезировать схему, предотвращающую пуск лифта при определенных условиях.

Шаг 1. Описание функции требуемой схемы

Лифт не может трогаться при открытой двери. Он также не может трогаться при перегрузке. Для пуска необходимо нажать кнопку.

Шаг 2. Назначение входных и выходных переменных

Входная переменная A назначается для дверного контакта. $A = 1$ значит, что дверной контакт замкнут, $A = 0$ значит, что дверной контакт разомкнут.

Входная переменная B назначается для перегрузки ($B = 1$: перегрузка, $B = 0$: нет перегрузки).

Входная переменная C назначается для кнопки ($C = 1$: кнопка нажата, $C = 0$: не нажата).

Выходной переменной величиной будет Z. $Z = 1$ значит, что лифт может ехать. $Z = 0$ значит, что лифт ехать не может.

Шаг 3. Составление таблицы истинности

Мы имеем три переменных величины. Следовательно, таблица истинности имеет 8 возможных вариантов (рис. 5.1). Лифт может ехать тогда, когда дверь закрыта ($A = 1$), нет перегрузки ($B = 0$) и кнопка нажата ($C = 1$).

Все эти условия выполняются одновременно только в варианте 6 таблицы истинности (рис. 5.1). Для этого варианта $Z = 1$. Во всех остальных случаях $Z = 0$.

Шаг 4. Определение необходимых логических операций

После составления таблицы истинности можно рассчитать схему. Правила расчета будут даны позже. Для такой простой задачи можно также применять метод подбора.

Z равно 1 только тогда, когда $A = 1$, $B = 0$ и $C = 1$. Если подать вход B на инвертор НЕ, то на выходе этого элемента будет состояние 1. При $A = 1$, $\bar{B} = 1$ и $C = 1$ имеем три 1-состояния. Они далее поступают на вход

Вар.	C	B	A	Z
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	1
7	1	1	0	0
8	1	1	1	0

Рис. 5.1. Таблица истинности для схемы безопасности лифта.

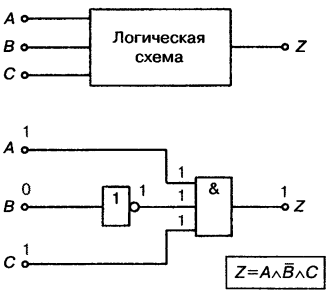


Рис. 5.2. Цифровая схема безопасности лифта.

трехвходового элемента И (рис. 5.2). На выходе элемента И только тогда действует 1, когда $A = 1$, $B = 0$ и $C = 1$. Этот выход является Z-выходом. На рис. 5.2 изображена требуемая схема безопасности. $Z = 1$ означает, например, что к выходу Z приложено напряжение +5 В. Это напряжение может коммутировать реле запуска лифта.

Способ нахождения схемы методом подбора можно описать как

> Возможность подобрать вариант схемы, выполняющей требуемые логические операции умножения или сложения входных переменных и их инвертированных значений.

Шаг 5. Упрощение и при необходимости преобразование схемы

Схему на рис. 5.2 упростить нельзя. Однако ее можно преобразовать. Предположим, что у нас есть под рукой только элементы ИЛИ-НЕ. Тогда функцию $Z = A \wedge \overline{B} \wedge C$ можно преобразовать:

$$Z = A \wedge \overline{B} \wedge C = \overline{\overline{A \wedge \overline{B} \wedge C}} = \overline{\overline{A} \vee B \vee \overline{C}}.$$

Схема, построенная на элементах ИЛИ-НЕ, изображена на рис. 5.3.

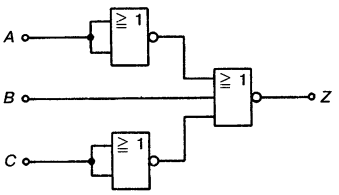


Рис. 5.3. Схема безопасности лифта на элементах ИЛИ-НЕ.

5.2. Нормальные формы записи

Стандартизованные формы записи выражений называются в математике **нормальными формами**. Для определенных целей необходимо логические функции приводить в нормальную форму.

5.2.1. Нормальная форма операции логического сложения ИЛИ

Нормальная форма записи логического сложения ИЛИ, также называемая нормальной дизъюнктивной формой записи (от дизъюнкция — сложение), является формой записи уравнения алгебры логики, в котором так называемые полные конъюнкции связаны друг с другом операцией логического сложения.

> Под полной конъюнкцией понимают операцию логического умножения, в которой участвуют все имеющиеся переменные или их инвертированные значения (от конъюнкция — умножение).

Если имеются переменные A и B, то получаются четыре возможные полные конъюнкции:

$A \wedge B$ | $\overline{A} \wedge B$ | $A \wedge \overline{B}$ | $\overline{A} \wedge \overline{B}$

Нормальная форма ИЛИ состоит из нескольких полных конъюнкций, которые логически складываются операцией ИЛИ. Она может состоять также из одной-единственной полной конъюнкции.

Все возможные логические функции могут быть записаны в виде нормальной формы ИЛИ.

Каждая нормальная форма ИЛИ имеет тесную связь с таблицей истинности. Покажем это на нескольких примерах.

Найдем таблицу истинности $Z_1 = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$. Она представлена на рис. 5.4. Видно, что Z_1 имеет два 1-состояния, а именно в случаях 1 и 4.

Вар.	B	A	$\bar{B}$	$\bar{A}$	$A \wedge B$	$\bar{A} \wedge \bar{B}$	$Z_1 = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$
1	0	0	1	1	0	1	1
2	0	1	1	0	0	0	0
3	1	0	0	1	0	0	0
4	1	1	0	0	1	0	1

$Z_1 = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$

Рис. 5.4. Таблица истинности для Z_1.

Теперь рассмотрим таблицу истинности для $Z_2 = (A \wedge B) \vee (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B})$ на рис. 5.5. Z_2 имеет три 1-состояния.

Вар.	B	A	$\bar{B}$	$\bar{A}$	$Z_3 = \bar{A} \wedge \bar{B}$
1	0	0	1	1	$1 \rightarrow \bar{A} \wedge \bar{B}$
2	0	1	1	0	0
3	1	0	0	1	0
4	1	1	0	0	1

$Z_3 = \bar{A} \wedge \bar{B}$

Рис. 5.6. Таблица истинности для Z_3.

Вар.	B	A	$\bar{B}$	$\bar{A}$	$A \wedge B$	$\bar{A} \wedge B$	$\bar{A} \wedge \bar{B}$	$Z_2 = (A \wedge B) \vee (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B})$
1	0	0	1	1	0	0	1	1
2	0	1	1	0	0	0	0	0
3	1	0	0	1	0	1	0	1
4	1	1	0	0	1	0	0	1

$Z_2 = (A \wedge B) \vee (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B})$

Рис. 5.5. Таблица истинности для Z_2.

Таблица истинности для $Z_3 = (\bar{A} \wedge \bar{B})$ изображена на рис. 5.6. Z_3 имеет одно 1-состояние. Из этого можно сделать вывод:

Количество 1-состояний в выходном столбце таблицы истинности (в данном случае Z-столбец) равно количеству полных конъюнкций нормальной формы ИЛИ.

Итак, вероятно, каждой 1 в Z-столбце соответствует полная конъюнкция. Рис. 5.8 подтверждает это утверждение. Мы имеем четыре возможных 1-состояния и четыре полных конъюнкции. Какая полная конъюнкция относится к какому 1-состоянию?

1-состояние полной конъюнкции $\bar{A} \wedge \bar{B}$ получается из таблицы на рис. 5.6. Таблица истинности полной конъюнкции $A \wedge B$ показана на рис. 5.7. Можно сделать вывод:

Если в рассмотренном варианте таблицы истинности переменная принимает значение 0, то в соответствующей полной конъюнкции она инвертируется. Если в рассмотренном варианте таблицы истинности переменная принимает значение 1, то в соответствующей полной конъюнкции она не инвертируется.

Вар.	B	A	Z
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	$1 \Rightarrow A \wedge B$

Рис. 5.7. Таблица истинности полной конъюнкции $A \wedge B$.

Вар.	B	A	Z
1	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B}$
2	0	1	$1 \Rightarrow A \wedge \bar{B}$
3	1	0	$1 \Rightarrow \bar{A} \wedge B$
4	1	1	$1 \Rightarrow A \wedge B$

Рис. 5.8. Соответствие полных конъюнкций возможным 1-состояниям.

Соответствие полных конъюнкций возможным 1-состояниям показано на рис. 5.8.

Каждому 1-состоянию в выходном столбце (*Z*-столбце) таблицы истинности соответствует полная конъюнкция.

При нескольких полных конъюнкциях нормальная форма ИЛИ получается посредством логического сложения ИЛИ полных конъюнкций.

При этом становится ясной взаимосвязь между таблицей истинности и нормальной формой ИЛИ, и для любой таблицы истинности мы можем записать соответствующую нормальную форму ИЛИ.

Нормальная форма ИЛИ представляет информационное содержание таблицы истинности в виде уравнения алгебры логики.

Пример

Дана таблица истинности на рис. 5.8а. Определить соответствующую нормальную форму ИЛИ. Каждое 1-состояние в *Z*-столбце соответствует полной конъюнкции. 0-состояния в *Z*-столбце можно не рассматривать.

Вар.	C	B	A	Z
1	0	0	0	0
2	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C}$
3	0	1	0	0
4	0	1	1	0
5	1	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge C$
6	1	0	1	0
7	1	1	0	0
8	1	1	1	$1 \Rightarrow A \wedge B \wedge C$

Рис. 5.8а. Таблица истинности.

Рассмотрим вариант 2 на рис. 5.8а. Переменная A равна 1. Поэтому она не инвертируется в полной конъюнкции. Переменные величины B и C равны 0. Они инвертируются в полной конъюнкции. Таким образом, полная конъюнкция для варианта 2 равна:

$$A \wedge \bar{B} \wedge \bar{C}.$$

Соответственно для варианта 5 полная конъюнкция $\bar{A} \wedge \bar{B} \wedge C$, а для варианта 8 полная конъюнкция $A \wedge B \wedge C$.

Нормальная форма ИЛИ является суммой всех полных конъюнкций:

$$Z = \left(A \wedge \bar{B} \wedge \bar{C}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C\right) \vee \left(A \wedge B \wedge C\right).$$

Эта нормальная форма ИЛИ представляет содержание таблицы истинности, изображенной на рис. 5.8а. Как и любое другое уравнение алгебры логики, нормальную форму можно преобразовать в таблицу истинности. Для выведенной нормальной формы получается таблица истинности (рис. 5.8а). Это можно проверить. Результат показан на рис. 5.9.

С помощью нормальной формы ИЛИ возможно для любой заданной или составленной по описанию проблемы таблицы истинности записать

Вар.	C	B	A	$\bar{B}$	$\bar{A}$	$A \wedge \bar{B} \wedge \bar{C}$	$\bar{A} \wedge \bar{B} \wedge C$	$A \wedge B \wedge C$	Z
1	0	0	0	1	1	0	0	0	0
2	0	0	1	1	0	1	0	0	1
3	0	1	0	0	1	0	0	0	0
4	0	1	1	0	0	0	0	0	0
5	1	0	0	1	1	0	1	0	1
6	1	0	1	1	0	0	0	0	0
7	1	1	0	0	1	0	0	0	0
8	1	1	1	0	0	0	0	1	1

Рис. 5.9. Обратное преобразование нормальной формы ИЛИ в таблицу истинности.

соответствующее уравнение алгебры логики. Метод подбора уже не нужен. Теперь можно без особых трудностей осуществлять синтез достаточно сложных логических схем.

5.2.2. Нормальная форма операции логического умножения И

Нормальная форма записи логического умножения И, также называемая нормальной конъюнктивной формой записи (от конъюнкция — умножение), является формой записи уравнения алгебры логики, в котором так называемые полные дизъюнкции связаны друг с другом операцией логического умножения.

> Под полной дизъюнкцией понимают операцию логического сложения, в которой участвуют все имеющиеся переменные или их инвертированные значения (от дизъюнкция — сложение).

Если имеются две переменные, например A и B, то получаются четыре возможные полные дизъюнкции:

$$A \wedge B \qquad \bar{A} \wedge B \qquad A \wedge \bar{B} \qquad \bar{A} \wedge \bar{B}$$

> Нормальная форма И состоит из нескольких полных дизъюнкций, которые логически перемножаются операцией И. Она может состоять также из одной-единственной полной дизъюнкции.

Если работа с нормальной формой ИЛИ не вызывает затруднения, то нормальная форма И уже не особенно нужна. Нормальную форму И можно легко преобразовать в нормальную форму ИЛИ.

Пример

Переведите нормальную форму И $Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$ в нормальную форму ИЛИ:

$$Z = (A \vee \bar{B}) \wedge (\bar{A} \vee B);$$

$$Z = \overline{\overline{(A \vee \bar{B}) \wedge (\bar{A} \vee B)}};$$

$$Z = \overline{\overline{(A \vee \bar{B}) \vee (\bar{A} \vee B)}};$$

$$Z = \overline{\overline{(\bar{A} \wedge B) \vee (A \wedge \bar{B})}};$$

$$\underline{Z = (\bar{A} \wedge B) \vee (A \wedge \bar{B}).}$$

5.3. Упрощение и преобразование нормальной формы ИЛИ с помощью алгебры логики

5.3.1. Упрощение нормальной формы ИЛИ

Нормальная форма ИЛИ воспроизводит содержание таблицы истинности в виде логического уравнения. По этому уравнению может быть синтезирована нужная схема.

> По нормальной форме ИЛИ можно синтезировать схему, удовлетворяющую соответствующей таблице истинности.

Часто эта схема не является самым простым вариантом из возможных. Во многих случаях нормальные формы ИЛИ можно упростить. Это упрощение может быть выполнено с помощью алгебры логики.

Пример 1

Упростите нормальную форму ИЛИ $Z = (A \wedge B) \vee (A \wedge \bar{B})$. Так как обе полные конъюнкции содержат переменную A, то она с помощью распределительного закона может быть вынесена за скобки:

$$Z = (A \wedge B) \vee (A \wedge \bar{B});$$

$$Z = A \wedge (B \vee \bar{B}).$$

Выражение $B \vee \bar{B}$ всегда равно 1 (см. гл. 4).

$$Z = A \wedge 1.$$

Логическое сложение переменной с 1 дает в итоге переменную. Результат упрощения нормальной формы ИЛИ:

$$\underline{Z = A}.$$

Пример 2

Упростите следующую нормальную форму ИЛИ:

$$Z = \underset{①}{(\bar{A} \wedge B \wedge C)} \vee \underset{②}{(\bar{A} \wedge B \wedge \bar{C})} \vee \underset{③}{(\bar{A} \wedge \bar{B} \wedge C)} \vee \underset{④}{(\bar{A} \wedge \bar{B} \wedge \bar{C})}.$$

Сначала упрощают полные конъюнкции ① и ②. $\bar{A} \wedge B$ рассматривается как одна переменная и выносится за скобки:

$$((\bar{A} \wedge B) \wedge C) \vee ((\bar{A} \wedge B) \wedge \bar{C}) = (\bar{A} \wedge B) \wedge (C \vee \bar{C}) = (\bar{A} \wedge B) \wedge 1 = (\bar{A} \wedge B).$$

Также можно упростить полные конъюнкции ③ и ④. $\bar{A} \wedge \bar{B}$ рассматривается как одна переменная и выносится за скобки:

$$((\bar{A} \wedge \bar{B}) \wedge C) \vee ((\bar{A} \wedge \bar{B}) \wedge \bar{C}) = (\bar{A} \wedge \bar{B}) \wedge (C \vee \bar{C}) = (\bar{A} \wedge \bar{B}) \wedge 1 = (\bar{A} \wedge \bar{B}).$$

Для Z тогда:

$$Z = \left(\bar{A} \wedge B\right) \vee \left(\bar{A} \wedge \bar{B}\right).$$

В этом уравнении $\bar{A}$ может быть вынесена за скобки как совместная переменная:

$$Z = \bar{A} \wedge \left(B \vee \bar{B}\right);$$

$$Z = \bar{A} \wedge 1;$$

$$\underline{Z = \bar{A}}.$$

Достаточно сложная нормальная форма ИЛИ сильно упростилась в этом примере. Такое сильное упрощение во многих случаях невозможно. Существует много нормальных форм ИЛИ, которые не упрощаются.

Пример 3

Упростите следующую нормальную форму ИЛИ:

$$Z = \left(A \wedge B \wedge \bar{C}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge \bar{C}\right).$$

Так как обе полные конъюнкции содержат переменную, то она с помощью распределительного закона может быть вынесена за скобки:

$$Z = \bar{C} \wedge \left((A \wedge B) \vee \left(\bar{A} \wedge \bar{B}\right)\right).$$

Можно поспорить, является ли вынесение за скобку $\bar{C}$ упрощением исходного выражения. Ответ будет очевиден только при сборке схемы на реальных элементах. Существенного преимущества в любом случае не получится.

5.3.2. Преобразование нормальной формы ИЛИ

Схема, которая строится согласно нормальной форме ИЛИ, должна базироваться на основных логических элементах. Во многих случаях можно использовать другие элементы, например И-НЕ или ИЛИ-НЕ. Нормальная форма ИЛИ в этих случаях должна быть преобразована.

Перевести нормальную форму ИЛИ на элементы И-НЕ очень просто. Нормальная форма ИЛИ сначала подвергается двойному отрицанию. Двойное отрицание, как известно, не меняет содержание уравнения. Затем нижняя черта инверсии разделяется согласно второй теореме де Моргана.

Пример 1

$$Z = \left(\bar{A} \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge C\right);$$

$$Z = \overline{\overline{\left(\bar{A} \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge C\right)}};$$

$$Z = \overline{\overline{\bar{A} \wedge B \wedge \bar{C}} \wedge \overline{A \wedge \bar{B} \wedge C}}.$$

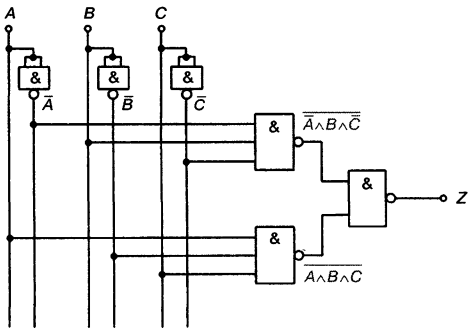


Рис. 5.10. Схема только с элементами И-НЕ.

Получающаяся из уравнения схема представлена на рис. 5.10.

Если требуется преобразовать нормальную форму ИЛИ так, чтобы схема состояла только из элементов ИЛИ-НЕ, то рекомендуется дважды инвертировать каждую полную конъюнкцию и каждую нижнюю черту инверсии преобразовать в соответствии с первой теоремой де Моргана. Затем все выражение еще раз подвергается двойному отрицанию.

Пример 2

$$Z = \left(\bar{B} \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge C\right);$$

$$Z = \overline{\overline{\left(\bar{A} \wedge B \wedge \bar{C}\right)} \vee \overline{\left(A \wedge \bar{B} \wedge C\right)}};$$

$$Z = \overline{A \vee \bar{B} \vee C} \vee \overline{\bar{A} \vee B \vee \bar{C}};$$

$$Z = \overline{\overline{A \vee \bar{B} \vee C} \vee \overline{\bar{A} \vee B \vee \bar{C}}}.$$

Схема к этому уравнению изображена на рис. 5.11.

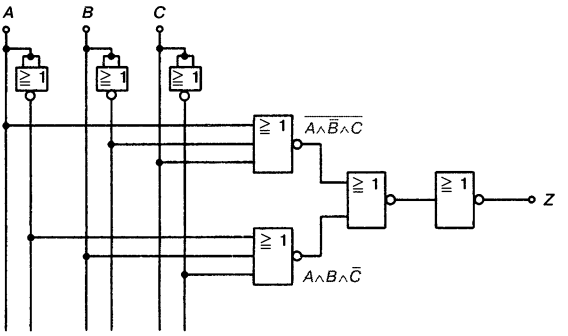


Рис. 5.11. Схема только с элементами ИЛИ-НЕ.

5.4. Метод карт Карно

Метод карт Карно служит для наглядного представления и упрощения нормальной формы ИЛИ. Он был придуман математиком Карно, и его еще называют методом диаграмм Карно.

5.4.1. Карта Карно для двух переменных

Карты Карно могут быть представлены в виде таблиц истинности для полных конъюнкций.

Карты Карно всегда имеют количество полей, равное количеству возможных полных конъюнкций.

При двух переменных возможны 4 полные конъюнкции. Таким образом, карта Карно для двух переменных должна иметь 4 поля (см. рис. 5.12). По краям карты записываются переменные. Каждая переменная величина должна быть представлена в инвертированной и в неинвертированной форме (рис. 5.12).

Переменные по краям являются координатной сеткой. Они определяют, какая полная конъюнкция какому полю принадлежит. На рис. 5.13 по своим полям расписаны 4 полных конъюнкции.

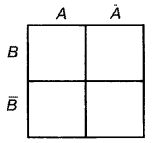


Рис. 5.12. Карта Карно для двух переменных (A, B).

Рис. 5.13. Карта Карно для двух переменных (A, B), заполненная полными конъюнкциями.

	A	$\bar{A}$
B	$A \wedge B$	$\bar{A} \wedge B$
$\bar{B}$	$A \wedge \bar{B}$	$\bar{A} \wedge B$

Поле полной конъюнкции $A \wedge B$ обозначено координатами A и B (рис. 5.13). Соответственно поле полной конъюнкции $\bar{A} \wedge B$ находится по координатам $\bar{A}$ и B. Так как полные конъюнкции определяются координатами, то нет необходимости записывать их в полной форме, как на рис. 5.13. Наличие полной конъюнкции может обозначаться 1 в соответствующем поле.

1 в поле карты Карно означает наличие полной конъюнкции.

На карте Карно (рис. 5.14) отмечены полные конъюнкции $A \wedge \bar{B}$ и $\bar{A} \wedge \bar{B}$. Карта Карно отражает следующую нормальную форму ИЛИ:

$$Z = \left(A \wedge \bar{B}\right) \vee \left(\bar{A} \wedge \bar{B}\right).$$

Символ Z в верхнем левом углу карты на рис. 5.14 показывает, что полные конъюнкции относятся к Z.

Отсутствующие полные конъюнкции обозначены нулем в соответствующем поле, или поле не заполняется.

Присваивание переменных координатам карты Карно производится произвольным образом.

Также возможно менять местами A и B на карте (рис. 5.15). Разумеется, переменные могут иметь совершенно другие обозначения, например E_1 и E_2. Прямое и инверсное значения переменной должны обязательно находиться на одной стороне карты.

Другое распределение координатных переменных ведет, естественно, к другому распределению полных конъюнкций по полям карты.

Желательно придерживаться определенной схемы распределения переменных и не менять ее без причины. Для облегчения работы рекомендуется первую переменную, например A, и ее инверсию все время ставить на верхнюю часть карты. Вторую переменную (например B) и ее инверсию ставить на левую часть карты.

Z	A	$\bar{A}$
B	0	0
$\bar{B}$	1	1

Рис. 5.14. Карта Карно для нормальной формы ИЛИ
$Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge \bar{B})$

	B	$\bar{B}$
A		
$\bar{A}$		

	E_1	$\bar{E}_1$
E_2		
$\bar{E}_2$		

Рис. 5.15. Карта Карно с измененными координатами.

Покажем на примере заполнение карты Карно нормальной формой ИЛИ и восстановление нормальной формы ИЛИ по карте Карно.

Пример 1

Занесите в карту Карно нормальную форму ИЛИ:

$$Z = (\bar{A} \wedge B) \vee (A \wedge B) \vee (\bar{A} \wedge \bar{B}).$$

Сначала нужно нарисовать карту Карно с данными координатами. Затем найти поля с полными конъюнкциями, присутствующими в нормальной форме и обозначить их 1. Результат показан на рис. 5.16.

Z	A	$\bar{A}$
B	1	1
$\bar{B}$		1

Рис. 5.16. Карта Карно для нормальной формы ИЛИ.

W	A	$\bar{A}$
B		1
$\bar{B}$	1	

Рис. 5.17. Карта Карно для нормальной формы ИЛИ.

Пример 2

Запишите нормальную форму ИЛИ, представленную на карте Карно (рис. 5.17).

Нормальная форма ИЛИ содержит 2 полные конъюнкции: одна $\bar{A} \wedge B$, вторая $A \wedge \bar{B}$. Следовательно, нормальная форма:

$$W = (\bar{A} \wedge B) \vee (A \wedge \bar{B}).$$

Представленная на карте Карно нормальная форма ИЛИ может быть упрощена при наличии определенных условий.

«Соседние» полные конъюнкции можно объединять в «группы».

Соседними считаются полные конъюнкции, клетки которых имеют общие стороны (рис. 5.18). Если клетки с полными конъюнкциями имеют только общий угол, то они не являются соседними.

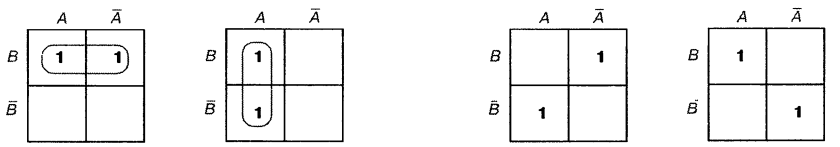


Рис. 5.18. Соседние и несоседние полные конъюнкции.

В одной группе могут быть объединены 2 или 4 соседние полные конъюнкции.

Каждая группа имеет определенные координаты. Группа слева наверху на карте Карно (рис. 5.18) имеет по одной стороне координату B, по другой — координату A и $\bar{A}$.

Содержание группы характеризуется ее координатами. Переменные, чьи координаты присутствуют в прямой и инверсной форме одновременно, исключаются.

Представленная на рис. 5.19 группа имеет координаты A, B и $\bar{B}$. Переменная B имеет как прямую, так и инверсную формы. Следовательно, она исключается. Значение группы будет A. Нормальная форма ИЛИ

$$Y = (A \wedge B) \vee (A \wedge \bar{B})$$

упростилась до $Y = A$.

$$Y = (A \wedge B) \vee (A \wedge \bar{B});$$

$$Y = A \wedge (B \vee \bar{B});$$

$$Y = A \wedge 1;$$

$$\underline{Y = A}.$$

Это упрощение может быть проверено с помощью алгебры логики.

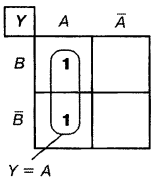


Рис. 5.19. Образование группы в карте Карно.

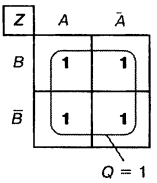


Рис. 5.20. Карта Карно с группой из четырех полей.

Особый случай представляет группа из 4 полных конъюнкций (рис. 5.20). Она имеет координаты A, $\bar{A}$, B, $\bar{B}$. Значит, переменные A и B исключаются. Значение группы равно 1. Справедливость этого можно доказать с помощью таблицы истинности. Алгебра логики также приведет к этому результату:

$$Z = (A \wedge B) \vee (A \wedge \bar{B}) \vee (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B});$$

$$Z = [A \wedge (B \vee \bar{B})] \vee [\bar{A} \wedge (B \vee \bar{B})];$$

$$Z = (A \wedge 1) \vee (\bar{A} \wedge 1);$$

$$Z = A \vee \bar{A};$$

$$\underline{Z = 1}.$$

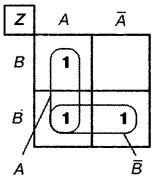


Рис. 5.21. Карта Карно с несколькими группами.

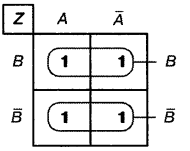


Рис. 5.22. Карта Карно с двумя группами из двух полей.

На одной карте можно образовать несколько групп (рис. 5.21). Одна полная конъюнкция может присутствовать в нескольких группах.

При наличии нескольких групп упрощенное уравнение получается в результате логического сложения значений отдельных групп.

Для карты (рис. 5.21) значения групп получаются равными A и $\bar{B}$. Упрощенное уравнение:

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

На карте Карно (рис. 5.20) можно образовать также две группы из двух полей. Тогда получится упрощенное уравнение, но не в самом простом виде. Покажем это. На рис. 5.22 показана карта Карно с такой группировкой. Значения групп равны B и $\bar{B}$. Упрощенное уравнение, следовательно:

$$Z = B \vee \bar{B}.$$

Сложение переменной величины с ее инверсией дает в итоге по правилам алгебры логики 1. Поэтому самой простой формой уравнения является $Z = 1$.

Для максимального упрощения уравнения необходимо образовывать группы как можно большего размера.

Чтобы закрепить полученные знания, решим следующий пример.

Пример 3

Максимально упростите при помощи карты Карно нормальную форму ИЛИ. Запишите упрощенное уравнение:

$$X = \left(\bar{A} \wedge \bar{B}\right) \vee \left(\bar{A} \wedge B\right) \vee \left(A \wedge \bar{B}\right).$$

Сначала полные конъюнкции заносятся в карту (рис. 5.23).

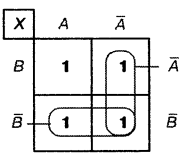


Рис. 5.23. Карта Карно к примеру 3.

Затем образуются две группы по два поля. Они имеют значения $\bar{A}$ и $\bar{B}$. Упрощенное уравнение:

$$X = \bar{A} \wedge \bar{B}.$$

5.4.2. Карта Карно для трех переменных

Для трех переменных возможны 8 различных полных конъюнкций (рис. 5.24). Следовательно, карта Карно для трех переменных должна иметь 8 клеток.

Распределение переменных по координатам может происходить, как и в карте для двух переменных любым образом. Однако целесообразно первые переменные поместить на верхнюю сторону диаграммы, а вторые величины — на левую сторону диаграммы. Третья переменная величина размещается на нижней стороне диаграммы. Для переменных величин A, B и C карта Карно изображена на рис. 5.25.

Вар.	C	B	A	Z
1	0	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge \bar{C}$
2	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C}$
3	0	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge \bar{C}$
4	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C}$
5	1	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge C$
6	1	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge C$
7	1	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge C$
8	1	1	1	$1 \Rightarrow A \wedge B \wedge C$

Рис. 5.24. Возможные полные конъюнкции для трех переменных.

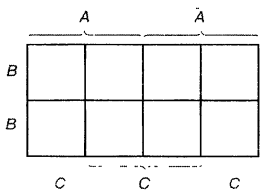


Рис. 5.25. Карта Карно для трех переменных.

Третья переменная C должна быть размещена, как указано на рис. 5.25. Если обозначить обе левые стороны клетки как C, а обе правые как $\bar{C}$, то для некоторых полных конъюнкций будет двойное место, а для некоторых — ни одного. На рис. 5.26 полные конъюнкции расписаны по ячейкам.

Для карт Карно с тремя переменными действуют правила, установленные для карт Карно с двумя переменными, со следующими дополнениями:

> В одной группе могут быть объединены 2, 4 или 8 соседних полных конъюнкций.

> Если быть совсем точными, карта Карно для трех переменных имеет цилиндрическую форму (рис. 5.27). Поэтому клетки, находящиеся в противоположных концах одной строки, являются соседними.

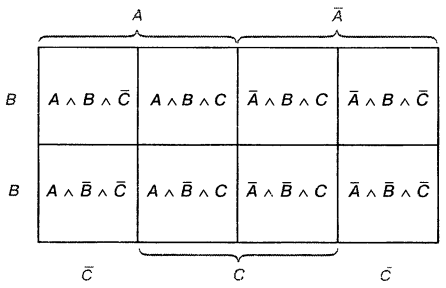


Рис. 5.26. Карта Карно для трех переменных с занесенными полными конъюнкциями.

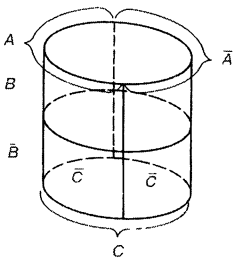


Рис. 5.27. Карта Карно для трех переменных в виде цилиндра.

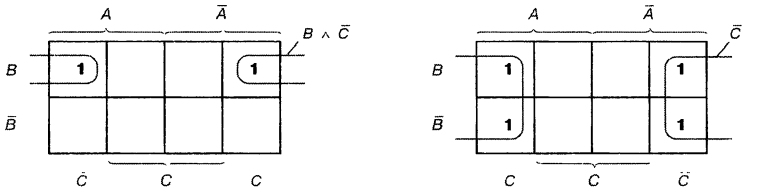


Рис. 5.28. Группировка по принципу расширенного соседства.

Карту Карно сложно представлять в виде цилиндра. Поэтому предпочитают форму на рис. 5.25, соблюдая принцип расширенного соседства. На рис. 5.28 представлены соседние полные конъюнкции и их группировки. Группа из двух клеток на верхней диаграмме имеет в итоге значение B и $\bar{C}$. Группа из четырех клеток на нижней диаграмме имеет значение $\bar{C}$. Группа должна быть прямоугольной или квадратной. Группа, изображенная на рис. 5.28а, недопустима.

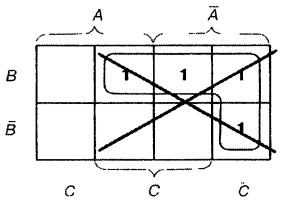


Рис. 5.28а. Недопустимая группа карты Карно.

Рассмотрим несколько примеров работы с картами Карно для трех переменных.

Пример 1

Заполните карту Карно полными конъюнкциями следующего уравнения:

$$Y = (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge B \wedge C).$$

Сначала надо правильно разместить полные конъюнкции по ячейкам (рис. 5.29). Запись лучше производить в алгебраической форме, тогда можно легко контролировать правильность выбора ячейки.

Каждую полную конъюнкцию обозначим 1 (рис. 5.30). При отсутствии затруднений можно сразу же рисовать обычную карту Карно.

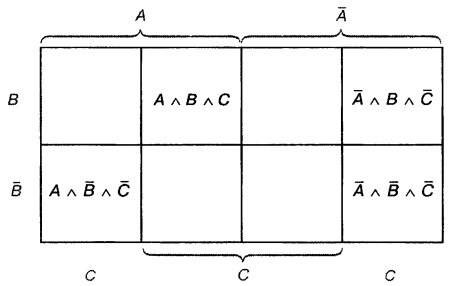


Рис. 5.29. Карта Карно с занесенными полными конъюнкциями к примеру 1.

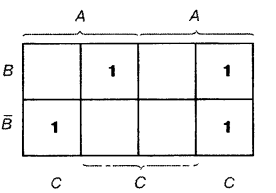


Рис. 5.30. Карта Карно к примеру 1.

Пример 2

Занесите данную нормальную форму ИЛИ в карту Карно и максимально упростите:

$$Z = (\bar{A} \wedge B \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge \bar{C}).$$

Имеющиеся полные конъюнкции обозначаются 1 (рис. 5.31). Затем производится группировка. Группу из четырех элементов образовать невозможно. Зато можно образовать 3 группы из двух клеток. Однако выделенная пунктиром группа является избыточной, так как двумя основными серыми группами все 1 уже охвачены. Если бы мы выбрали пунктирную группу в качестве основной, найденное уравнение не было бы максимально простым.

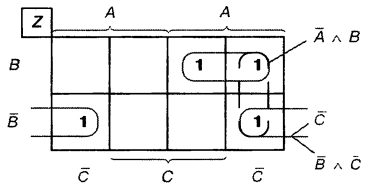


Рис. 5.31. Карта Карно к примеру 2.

Верхняя серая группа (рис. 5.31) имеет значение $\bar{A} \wedge B$. Значение нижней серой группы — $\bar{B} \wedge \bar{C}$. (Переменная A выпадает, так как встречается в координатах этой группы как в прямой, так и в инверсной формах.) Значения групп логически складываются. При этом получается упрощенное уравнение:

$$Z = (\bar{A} \wedge B) \vee (\bar{B} \wedge \bar{C}).$$

Пример 3

Запишите нормальную форму ИЛИ, заключенную в карте Карно (рис. 5.32).

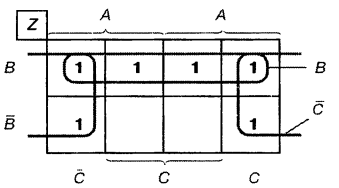


Рис. 5.32. Карта Карно к примеру 3.

Максимально упростите нормальную форму ИЛИ.

Нормальная форма ИЛИ по карте Карно:

$$Z = (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C) \vee (\bar{A} \wedge B \wedge C) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge \bar{C}) \vee A \wedge B \wedge C.$$

Могут быть образованы 2 группы из четырех клеток. Одна имеет значение B, другая $\bar{C}$. Упрощенное уравнение:

$$Z = B \vee \bar{C}.$$

Так как возможно образование двух больших групп из четырех клеток, то получается значительное упрощение нормальной формы.

5.4.3. Карта Карно для четырех переменных

Карта Карно для четырех переменных должна иметь 16 клеток, так как возможны 16 различных полных конъюнкций (рис. 5.33). Карта Карно для четырех переменных изображена на рис. 5.34.

Вар.	D	C	B	A	Z
1	0	0	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}$
2	0	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}$
3	0	0	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge \bar{C} \wedge \bar{D}$
4	0	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge \bar{D}$
5	0	1	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge C \wedge \bar{D}$
6	0	1	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge C \wedge \bar{D}$
7	0	1	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge C \wedge \bar{D}$
8	0	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge \bar{D}$
9	1	0	0	0	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge D$
10	1	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge D$
11	1	0	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge \bar{C} \wedge D$
12	1	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge D$
13	1	1	0	0	$1 \Rightarrow \dot{A} \wedge \bar{B} \wedge C \wedge D$
14	1	1	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge C \wedge D$
15	1	1	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge C \wedge D$
16	1	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge D$

Рис. 5.33. Возможные полные конъюнкции для четырех переменных.

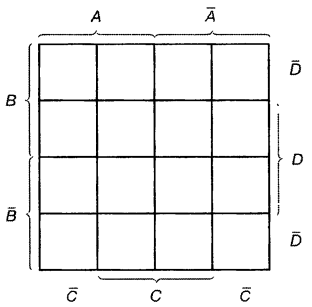


Рис. 5.34. Карта Карно для четырех переменных.

Переменные обозначены, как и раньше, A, B и C. Плюс добавлена переменная величина D. Разумеется, переменные могут быть обозначены иначе, например E_1, E_2, E_3, E_4. 16 полных конъюнкций показаны на рис. 5.35.

Для карт Карно с четырьмя переменными действуют правила, ранее установленные для карт Карно, со следующими дополнениями:

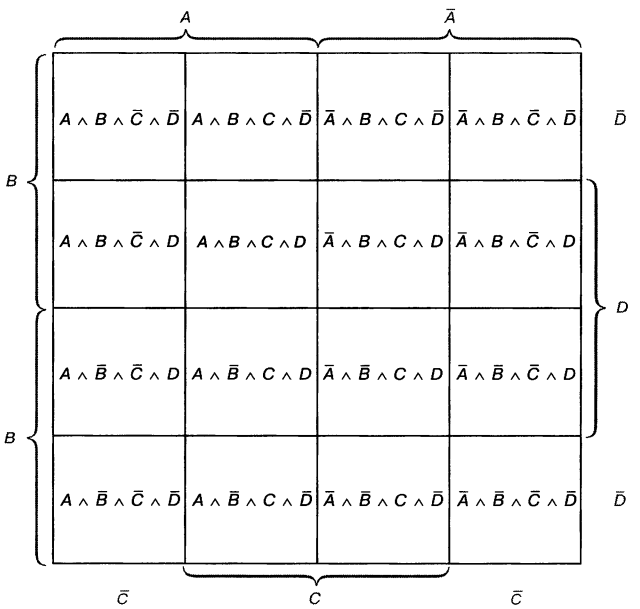


Рис. 5.35. Карта Карно для четырех переменных с занесенными полными конъюнкциями.

В одной группе могут быть объединены 2, 4, 8 или 16 соседних полных конъюнкций.

Карта Карно для четырех переменных имеет форму шара. Поэтому клетки, находящиеся в противоположных концах одной строки или столбца, являются соседними.

Разъясним подробнее принцип расширенного соседства. Рассмотрим рис. 5.36. Карта (*а*) показывает, что группы из двух клеток можно образовать не только из полных конъюнкций, которые находятся на концах одной строки, но и из полных конъюнкций, находящихся на концах одного столбца.

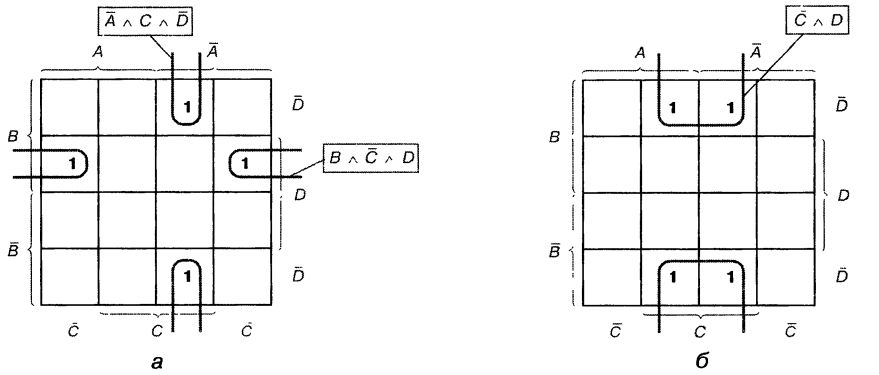


Рис. 5.36. Группировка по принципу расширенного соседства.

Диаграмма (*б*) показывает образование группы из четырех клеток.

Диаграмма (*в*) также показывает образование счетверенной группы. Единицы по углам являются соседними, так как при форме шара поля прилегают друг к другу смежными сторонами.

Другое дело карта на рис. 5.37. Только две единицы по углам не могут образовать сдвоенную группу, так как они не являются смежными — как показано на виде снизу. Рассмотрим ряд примеров с картами Карно для четырех переменных.

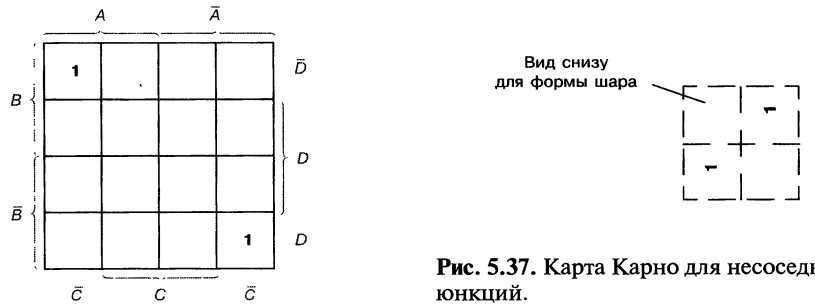


Рис. 5.37. Карта Карно для несоседних полных конъюнкций.

Пример 1

Составьте карту Карно по следующей нормальной форме ИЛИ:

$$Y = \underset{①}{(A \wedge B \wedge \bar{C} \wedge \bar{D})} \vee \underset{②}{(A \wedge B \wedge \bar{C} \wedge D)} \vee \underset{③}{(A \wedge B \wedge C \wedge D)} \vee$$
$$\vee \underset{④}{(\bar{A} \wedge B \wedge C \wedge D)} \vee \underset{⑤}{(\bar{A} \wedge B \wedge \bar{C} \wedge D)}.$$

Для большей наглядности полные конъюнкции отмечены серыми номерами. Они обозначают соответствующие клетки. На рис. 5.38 показана искомая диаграмма.

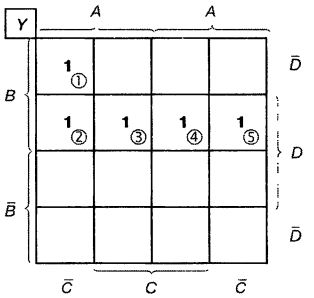


Рис. 5.38. Карта Карно к примеру 1.

Пример 2

Для задач управления требуется схема, удовлетворяющая таблице истинности (рис. 5.39). Эта схема должна быть максимально простой.

Вар.	D	C	B	A	Z
1	0	0	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}$
2	0	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}$
3	0	0	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge \bar{C} \wedge \bar{D}$
4	0	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge \bar{D}$
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge D$
11	1	0	1	0	0
12	1	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge D$
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

Рис. 5.39. Таблица истинности к примеру 2.

Из таблицы истинности может быть определена нормальная форма ИЛИ

$$Z = \underset{①}{(\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})} \vee \underset{②}{(A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})} \vee \underset{③}{(\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D})} \vee$$
$$\vee \underset{④}{(A \wedge B \wedge \bar{C} \wedge \bar{D})} \vee \underset{⑩}{(A \wedge \bar{B} \wedge \bar{C} \wedge D)} \vee \underset{⑫}{(A \wedge B \wedge \bar{C} \wedge D)}.$$

Номера полных конъюнкций совпадают с номерами вариантов таблицы истинности. Полные конъюнкции далее заносятся в диаграмму (рис. 5.40).

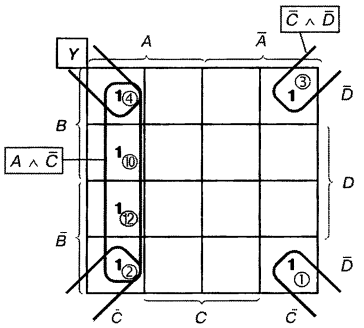


Рис. 5.40. Карта Карно к примеру 2.

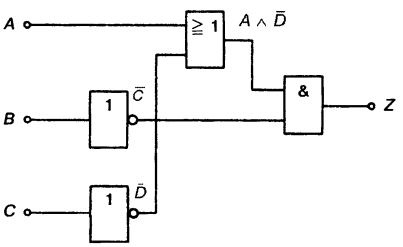


Рис. 5.41. Схема к примеру 2.

Следующим шагом является упрощение нормальной формы ИЛИ с помощью группировки соседних полных конъюнкций. Возможно образование двух групп из 4 клеток со значениями $\bar{C} \wedge \bar{D}$ и $A \wedge \bar{C}$. Упрощенное уравнение выглядит так:

$$Z = (\bar{C} \wedge \bar{D}) \vee (A \wedge \bar{C}).$$

Переменная $\bar{C}$ может быть вынесена за скобку:

$$Z = (\bar{C} \wedge \bar{D}) \vee (A \wedge \bar{C}) = \bar{C} \wedge (A \vee \bar{D}).$$

Получившаяся схема представлена на рис. 5.41.

5.4.4. Карта Карно для пяти переменных

Для пяти переменных возможны 32 различные полные конъюнкции. Следовательно, карта Карно для пяти переменных должна иметь 32 поля. Но на одном уровне в диаграмму для четырех переменных новые переменные уже добавить не получится.

Диаграмме надо добавить второй уровень. На рис. 5.42 показано, что имеется в виду. Переменные величины будут, как раньше, обозначены *A*, *B*, *C* и *D*. К ним добавляется переменная *E*.

К нижнему уровню диаграммы присоединяется координата *E*, к верхнему — координата $\bar{E}$. Нарисовать такую двухуровневую карту сложно, поэтому уровни рисуют рядом.

Диаграмма для пяти переменных состоит из двух таблиц, расположенных одна над другой (рис. 5.42а). Такая диаграмма имеет 32 ячейки для 32 полных конъюнкций.

Для карт Карно с пятью переменными действуют правила, ранее установленные для карт Карно, со следующими дополнениями:

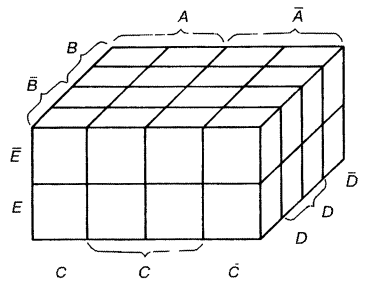


Рис. 5.42. Карта Карно для пяти переменных.

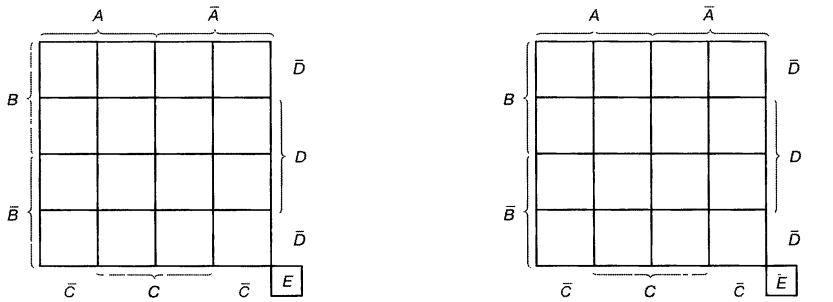


Рис. 5.42а. Карта Карно для пяти переменных, состоящая из двух таблиц.

В одной группе могут быть объединены 2, 4, 8, 16 или 32 соседних полных конъюнкций.

Сгруппированы могут быть также те полные конъюнкции, чьи поля находятся друг под другом в таблицах (рис. 5.42).

Рассмотрим эти правила на примерах.

Пример 1

Занесите данную нормальную форму ИЛИ в карту Карно и максимально упростите:

$$Z = \underset{①}{\left(\bar{A} \wedge B \wedge C \wedge \bar{D} \wedge E\right)} \vee \underset{②}{\left(\bar{A} \wedge B \wedge C \wedge D \wedge E\right)} \vee \underset{③}{\left(\bar{A} \wedge B \wedge C \wedge \bar{D} \wedge \bar{E}\right)} \vee$$

$$\vee \underset{④}{\left(\bar{A} \wedge B \wedge C \wedge D \wedge \bar{E}\right)} \vee \underset{⑤}{\left(\bar{A} \wedge \bar{B} \wedge C \wedge D \wedge \bar{E}\right)} \vee \underset{⑥}{\left(\bar{A} \wedge \bar{B} \wedge C \wedge \bar{D} \wedge \bar{E}\right)}.$$

Для большей наглядности полные конъюнкции отмечены серыми номерами. Они обозначают соответствующие клетки диаграммы. Возможно образование двух групп из 4 клеток. Серая группа на правой таб-

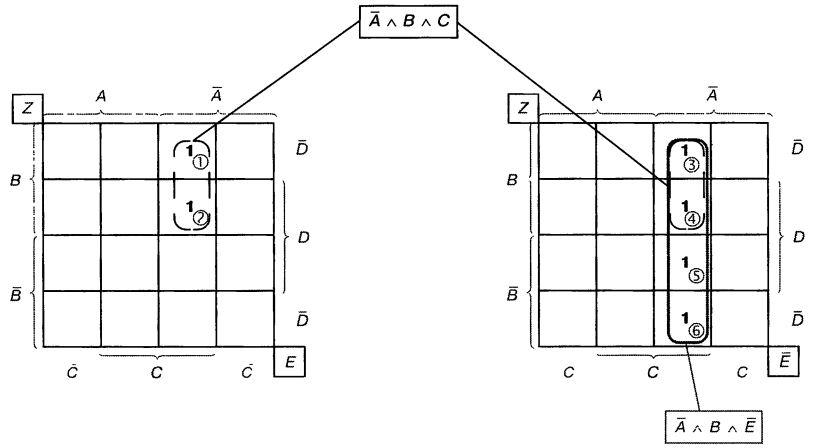


Рис. 5.43. Карта Карно к примеру 1.

лице имеет значение $\bar{A} \wedge C \wedge \bar{E}$. Переменные B и D в этой группе исключаются.

Выделенная пунктиром группа из 4 клеток проходит сквозь оба уровня. Для этого следует мысленно положить друг на друга уровни (рис. 5.43). Значение этой группы $\bar{A} \wedge B \wedge C$. Так как группа проходит сквозь два уровня, переменная E выпадает. Переменная D также выпадает. В итоге получается упрощенное уравнение:

$$Z = (\bar{A} \wedge B \wedge C) \vee (\bar{A} \wedge C \wedge \bar{E}).$$

Пример 2

В диаграмме на рис. 5.44 задана нормальная форма ИЛИ. Максимально упростите ее.

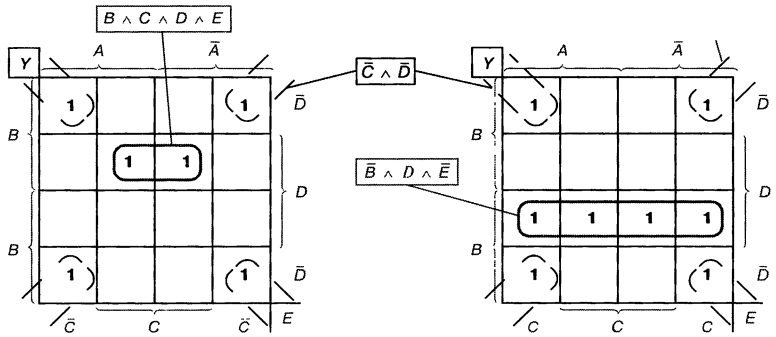


Рис. 5.44. Карта Карно к примеру 2.

Единицы по углам обоих уровней образуют восьмерную группу (группа из восьми ячеек). Эта группа проходит через два уровня. Ее значение $\bar{C} \wedge \bar{D}$.

Далее можно сформировать группы из 2 и 4 клеток. Группа из 4 клеток имеет значение $\bar{B} \wedge D \wedge \bar{E}$. Значение сдвоенной группы $B \wedge C \wedge D \wedge E$. Получается упрощенное уравнение:

$$Y = (\bar{C} \wedge \bar{D}) \vee (\bar{B} \wedge D \wedge \bar{E}) \vee (B \wedge C \wedge D \wedge E).$$

Упрощение значительно. Это видно, если записать содержащуюся в диаграмме на рис. 5.44 нормальную форму ИЛИ.

Пример 3

Запишите содержащуюся в диаграмме на рис. 5.44 нормальную форму ИЛИ.

Левая таблица диаграммы содержит 6 полных конъюнкций, правая таблица — 8. Таким образом, получается нормальная форма ИЛИ с 14 полными конъюнкциями:

$$Y = (A \wedge B \wedge \bar{C} \wedge \bar{D} \wedge E) \vee (\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D} \wedge E) \vee (A \wedge B \wedge C \wedge D \wedge E) \vee$$

$$\vee (\bar{A} \wedge B \wedge C \wedge D \wedge E) \vee (A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \wedge E) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \wedge E) \vee$$

$$\vee\left(A \wedge B \wedge \bar{C} \wedge \bar{D} \wedge \bar{E}\right) \vee \left(\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D} \wedge \bar{E}\right) \vee \left(A \wedge \bar{B} \wedge \bar{C} \wedge D \wedge \bar{E}\right) \vee$$
$$\vee\left(A \wedge \bar{B} \wedge C \wedge D \wedge \bar{E}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C \wedge D \wedge \bar{E}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge D \wedge \bar{E}\right) \vee$$
$$\vee\left(A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \wedge \bar{E}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D} \wedge \bar{E}\right).$$

5.4.5. Карта Карно для более чем пяти переменных

На практике нормальные формы ИЛИ с более чем пятью переменными встречаются редко. Поэтому редко возникает необходимость и в диаграммах Карно для более чем пяти переменных. Однако такие диаграммы реальны. Диаграммы для шести переменных еще можно наглядно представить. При семи и более переменных наглядное представление диаграммы затруднительно.

Для шести переменных возможны 64 различные полные конъюнкции. Следовательно, карта Карно для шести переменных должна иметь 64 поля. Если в качестве исходной брать диаграммы для пяти переменных величин, то к ней надо добавить еще третий и четвертый уровни-этажи (рис. 5.45).

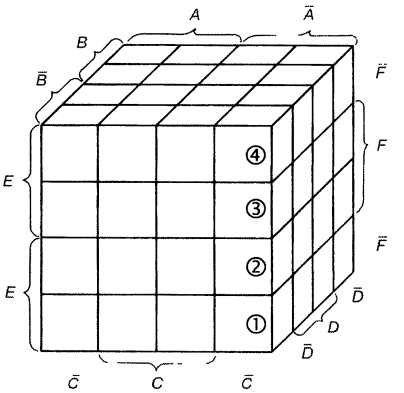


Рис. 5.45. Карта Карно для шести переменных.

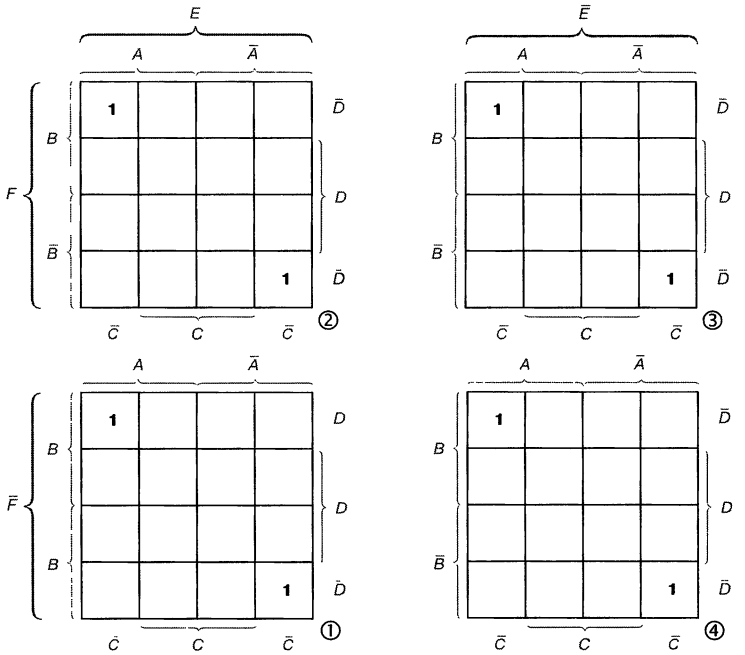


Рис. 5.46. Карта Карно для шести переменных, развернутая на одной плоскости.

Четыре уровня можно расположить в одной плоскости (рис. 5.46). При группировке нужно постоянно помнить, как реально расположены уровни относительно друг друга.

Для нормальной формы ИЛИ с шестью и более переменными целесообразно заменять две или три переменных новой переменной. Упрощение может происходить в несколько этапов:

$$Z = \left(A \wedge \bar{B} \wedge C \wedge D \wedge E \wedge \bar{F}\right) \vee \left(A \wedge \bar{B} \wedge \bar{C} \wedge D \wedge E \wedge \bar{F}\right) \vee$$

$$\vee \left(A \wedge B \wedge C \wedge \bar{D} \wedge E \wedge F\right) \vee \left(A \wedge B \wedge C \wedge D \wedge E \wedge F\right).$$

Пример

Все четыре полные конъюнкции содержат переменные A и E в одинаковом, в данном случае, неинвертируемом виде. Будем рассматривать $A \wedge E$ как одну переменную.

$$A \wedge E = P.$$

При таком условии получается нормальная форма ИЛИ только с пятью переменными:

$$Z = \left(P \wedge \bar{B} \wedge C \wedge D \wedge \bar{F}\right) \vee \left(P \wedge \bar{B} \wedge \bar{C} \wedge D \wedge \bar{F}\right) \vee$$

$$\vee \left(P \wedge B \wedge C \wedge \bar{D} \wedge F\right) \vee \left(P \wedge B \wedge C \wedge D \wedge F\right).$$

После упрощения снова заменим P на $A \wedge E$.

5.5. Расчет логических схем

5.5.1. Общие указания

Для синтеза схем в разд. 5.1 описаны следующие шаги:

1. Описание функции требуемой схемы.
2. Назначение входных и выходных переменных величин и присвоение значений 0 и 1.
3. Составление таблицы истинности.
4. Определение необходимых логических операций.
5. Упрощение и при необходимости преобразование схемы.

Если известна таблица истинности, то теперь целесообразно 4-й этап начинать с составления нормальной формы ИЛИ. Она будет максимально упрощена с помощью диаграммы Карно. В конце шага 4 получается упрощенная логическая функция, по которой можно собирать логическую цифровую схему.

В шаге 5 проверяется, является ли дальнейшее упрощение найденной с помощью алгебры логики функции возможным и рациональным. Если да, то упрощение необходимо провести.

Теперь надо узнать, какие логические элементы имеются в наличии.

Логическую функцию нужно преобразовать так, чтобы она содержала только имеющиеся логические элементы. Затем можно собирать схему.

5.5.2. Цифровая схема включения и выключения из нескольких мест

С помощью логических элементов требуется синтезировать схему, которая функционирует как схема включения и выключения из нескольких мест. Выходное состояние должно меняться только в случае, если меняется состояние одного из входов. Если оба входа меняют свое состояние, то выходное состояние измениться не должно. Схема должна быть построена на элементах ИЛИ-НЕ.

Искомая схема имеет два входа и один выход. Входные переменные называются A и B. Выходная переменная обозначается Z (рис. 5.47).

Таблица истинности схемы с двумя входными переменными имеет 4 варианта (рис. 5.48). Исходное состояние Z для первого варианта может устанавливаться любым образом. Выбрано $Z = 0$.

При переходе от варианта 1 к варианту 2 переменная A меняет свое состояние. Переменная B состояние не меняет. Если только один из входов меняет состояние, то согласно поставленному заданию выход Z должен поменять свое состояние. Z должен быть равен 1.

При переходе от варианта 2 к варианту 3 переменные A и B меняют свои состояния. Z не должно измениться. При переходе от варианта 3 к варианту 4 переменная A меняет свое состояние с 0 на 1. B остается равной 1. Таким образом, Z должно поменять состояние с 1 на 0. Таблица истинности готова. Она могла бы выглядеть иначе, если бы мы в варианте 1 выбрали $Z = 1$.

Для таблицы истинности (рис. 5.48) нужно записать нормальную форму ИЛИ. Она выглядит так:

$$Z = \left(\bar{A} \wedge B\right) \vee \left(A \wedge \bar{B}\right).$$

Если занести нормальную форму ИЛИ в диаграмму Карно, то видно, что дальнейшее упрощение невозможно (рис. 5.49).

Так как схема должна быть построена на элементах ИЛИ-НЕ, требуется преобразовать уравнения:

$$Z = \left(\bar{A} \wedge B\right) \vee \left(A \wedge \bar{B}\right);$$

$$Z = \overline{\overline{\left(\bar{A} \wedge B\right) \vee \left(A \wedge \bar{B}\right)}};$$

$$Z = \overline{\overline{\bar{A} \wedge B} \wedge \overline{A \wedge \bar{B}}}.$$

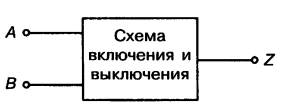


Рис. 5.47. Блок-схема схемы включения и выключения.

Вар.	B	A	Z
1	0	0	0
2	0	1	$1 \Rightarrow A \wedge \bar{B}$
3	1	0	$1 \Rightarrow \bar{A} \wedge B$
4	1	1	0

Рис. 5.48. Таблица истинности схемы включения и выключения.

	A	$\bar{A}$
B	1	
$\bar{B}$		1

Рис. 5.49. Карта Карно схемы включения и выключения.

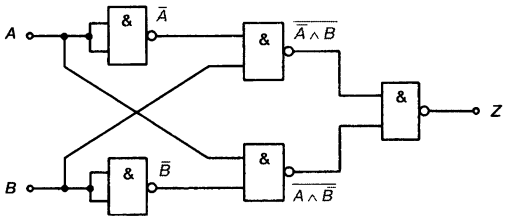


Рис. 5.50. Цифровая схема.

Схема, построенная согласно преобразованному уравнению, показана на рис. 5.50.

5.5.3. Переключатель «два из трех»

Рис. 5.51. Аварийные датчики и схема переключателя «два из трех».

Системы, связанные с повышенным риском, например атомная электростанция, должны быть в случае аварии сразу остановлены. Отключение происходит автоматически, с помощью цифровой схемы. В аварийных датчиках, ответственных за отключение, могут происходить ложные срабатывания. Поэтому в каждом критическом месте ставят три одинаковых аварийных датчика (рис. 5.51).

Отключение должно происходить только тогда, когда сработали по меньшей мере два аварийных датчика из трех. Такой подход предотвращает ненужные отключения системы, которые приносят финансовые потери. Аварийные датчики при срабатывании имеют состояние 1. Отключение системы должно происходить, если на выходе схемы действует состояние 1.

Итак, требуется схема, выход которой имеет состояние 1 тогда, когда, по меньшей мере 2 из 3 входов имеют состояние 1. Такая схема называется **переключатель «два из трех»**.

Входные переменные получают имена A, B и C. Выходная переменная — Z. Составим таблицу истинности по словесному описанию принципа действия схемы. Всегда, когда две входных переменных равны 1, $Z = 1$. Если все три входных переменных равны 1, Z также должна быть равна 1. Такая таблица истинности показана на рис. 5.52.

Вар.	C	B	A	Z	
1	0	0	0	0	
2	0	0	1	0	
3	0	1	0	0	
4	0	1	1	1	$\Rightarrow A \wedge B \wedge \bar{C}$
5	1	0	0	0	
6	1	0	1	1	$\Rightarrow A \wedge \bar{B} \wedge C$
7	1	1	0	1	$\Rightarrow \bar{A} \wedge B \wedge C$
8	1	1	1	1	$\Rightarrow A \wedge B \wedge C$

Рис. 5.52. Таблица истинности переключателя «два из трех».

Далее по составленной таблице истинности записывается нормальная форма ИЛИ:

$$Z = \left(A \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge C\right) \vee$$
$$\vee \left(\bar{A} \wedge B \wedge C\right) \vee \left(A \wedge B \wedge C\right).$$

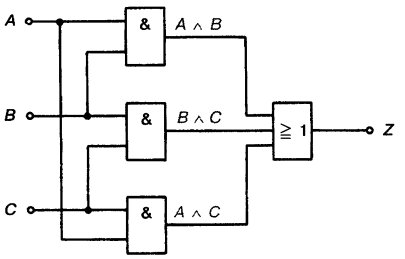


Рис. 5.54. Схема переключателя «два из трех».

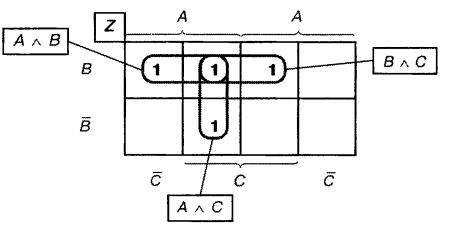


Рис. 5.53. Карта Карно переключателя «два из трех».

Нормальная форма ИЛИ упрощается с помощью диаграммы Карно (рис. 5.53). Можно образовывать три двойных группы. Упрощенное уравнение имеет вид:

$$Z = (A \wedge B) \vee (B \wedge C) \vee (A \wedge C).$$

По этому уравнению можно строить схему (рис. 5.54).

Часто под рукой имеются только элементы ИЛИ-НЕ. Чтобы построить схему только на элементах ИЛИ-НЕ, преобразуем уравнение:

$$Z = (A \wedge B) \vee (B \wedge C) \vee (A \wedge C);$$

$$Z = \overline{\overline{(A \wedge B) \vee (B \wedge C) \vee (A \wedge C)}};$$

$$Z = \overline{\overline{A \wedge B} \wedge \overline{B \wedge C} \wedge \overline{A \wedge C}}.$$

Соответствующая схема показана на рис. 5.55.

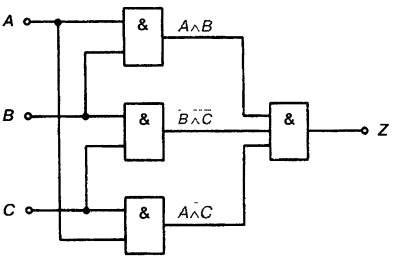


Рис. 5.55. Схема переключателя «два из трех» на элементах ИЛИ-НЕ.

5.5.4. Схема контроля четности

Для обнаружения ошибок в кодах (см. разд. 8.7 и 8.8), а также для задач контроля и наблюдения часто требуется схема, в которой выход равен 1 тогда, когда четное число входов имеют состояние 1.

Такая схема называется схемой контроля четности.

Требуется синтезировать схему с четырьмя входами. Входные переменные — *A*, *B*, *C* и *D*. Выходная переменная — *Y*.

Сначала нужно составить таблицу истинности. *Y* всегда будет равен 1, если 0, 2 или 4 входные переменные равны 1 (рис. 5.56).

Из таблицы истинности получается нормальная форма ИЛИ:

$$Y = \underset{①}{(\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})} \vee \underset{④}{(A \wedge B \wedge \bar{C} \wedge \bar{D})} \vee \underset{⑥}{(A \wedge \bar{B} \wedge C \wedge \bar{D})} \vee \underset{⑦}{(\bar{A} \wedge B \wedge C \wedge \bar{D})} \vee$$

$$\vee \underset{⑩}{(A \wedge \bar{B} \wedge \bar{C} \wedge D)} \vee \underset{⑪}{(\bar{A} \wedge B \wedge \bar{C} \wedge D)} \vee \underset{⑬}{(\bar{A} \wedge \bar{B} \wedge C \wedge D)} \vee \underset{⑯}{(A \wedge B \wedge C \wedge D)}.$$

Вар.	D	C	B	A	Y
1	0	0	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}$
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge \bar{D}$
5	0	1	0	0	0
6	0	1	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge C \wedge \bar{D}$
7	0	1	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge C \wedge \bar{D}$
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge \bar{C} \wedge D$
11	1	0	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge \bar{C} \wedge D$
12	1	0	1	1	0
13	1	1	0	0	$1 \Rightarrow \bar{A} \wedge \bar{B} \wedge C \wedge D$
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge D$

Рис. 5.56. Таблица истинности для схемы контроля четности.

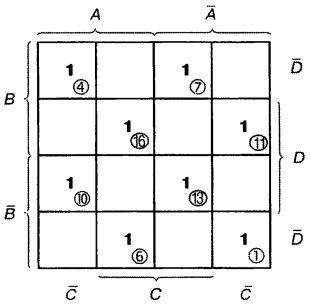


Рис. 5.57. Карта Карно схемы контроля четности.

Отдельные полные конъюнкции пронумерованы. Попробуем упростить нормальную форму ИЛИ с помощью карты Карно (рис. 5.57). Здесь мы столкнулись с редким случаем, когда образование групп невозможно. Значит, данная нормальная форма ИЛИ не упрощается, и ее схема приведена на рис. 5.58.

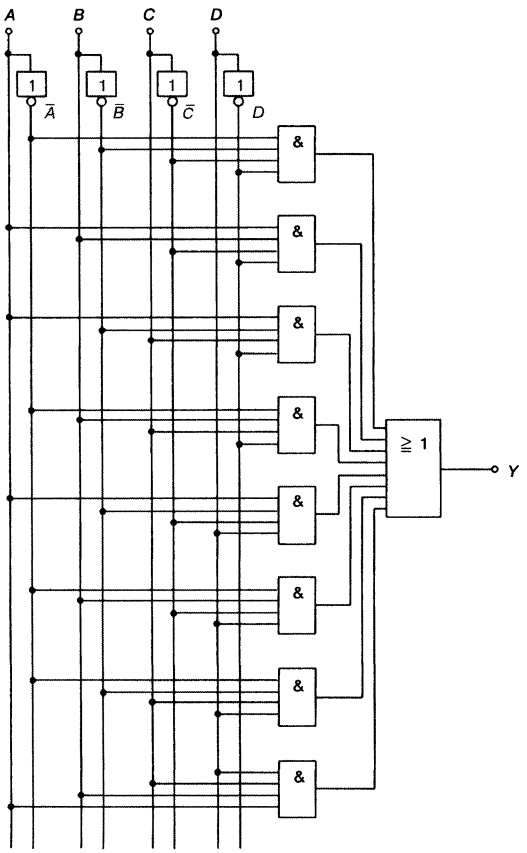


Рис. 5.58. Схема контроля четности.

Вар.	E	D	C	B	A	Z
1	0	0	0	0	0	0
2	0	0	0	0	1	0
3	0	0	0	1	0	0
4	0	0	0	1	1	0
5	0	0	1	0	0	0
6	0	0	1	0	1	0
7	0	0	1	1	0	0
8	0	0	1	1	1	0
9	0	1	0	0	0	0
10	0	1	0	0	1	0
11	0	1	0	1	0	0
12	0	1	0	1	1	0
13	0	1	1	0	0	0
14	0	1	1	0	1	0
15	0	1	1	1	0	0
16	0	1	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge D \wedge \bar{E}$
17	1	0	0	0	0	0
18	1	0	0	0	1	0
19	1	0	0	1	0	0
20	1	0	0	1	1	0
21	1	0	1	0	0	0
22	1	0	1	0	1	0
23	1	0	1	1	0	0
24	1	0	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge \bar{D} \wedge E$
25	1	1	0	0	0	0
26	1	1	0	0	1	0
27	1	1	0	1	0	0
28	1	1	0	1	1	$1 \Rightarrow A \wedge B \wedge \bar{C} \wedge D \wedge E$
29	1	1	1	0	0	0
30	1	1	1	0	1	$1 \Rightarrow A \wedge \bar{B} \wedge C \wedge D \wedge E$
31	1	1	1	1	0	$1 \Rightarrow \bar{A} \wedge B \wedge C \wedge D \wedge E$
32	1	1	1	1	1	$1 \Rightarrow A \wedge B \wedge C \wedge D \wedge E$

Рис. 5.59. Таблица истинности пороговой логической схемы.

5.5.5. Пороговая логическая схема

Пороговой логической схемой называется схема, в которой определенное минимальное количество входных переменных должно иметь состояние 1, чтобы на выходе появилась логическая 1.

Например, нужно рассчитать схему с пятью входными переменными. На выходе должна быть 1 только тогда, когда по меньшей мере на 4 входах присутствует 1.

Входные переменные имеют имена *A*, *B*, *C*, *D* и *E*. Выходная переменная — *Z*. Сначала нужно определить таблицу истинности. При пяти переменных величинах возможны 32 варианта (рис. 5.59):

$$Z = (A \wedge B \wedge C \wedge D \wedge \bar{E}) \vee (A \wedge B \wedge C \wedge \bar{D} \wedge E) \vee (A \wedge B \wedge \bar{C} \wedge D \wedge E) \vee$$
$$\vee (A \wedge \bar{B} \wedge C \wedge D \wedge E) \vee (\bar{A} \wedge B \wedge C \wedge D \wedge E) \vee (A \wedge B \wedge C \wedge D \wedge E).$$

Нормальная форма ИЛИ состоит из шести полных конъюнкций.

Нормальная форма ИЛИ упрощается с помощью диаграммы Карно (рис. 5.60). Можно образовать 5 двойных групп. Получается следующая упрощенная логическая функция:

$$Z = (A \wedge B \wedge C \wedge E) \vee (A \wedge B \wedge D \wedge E) \vee (A \wedge B \wedge C \wedge D) \vee$$
$$\vee (A \wedge C \wedge D \wedge E) \vee (B \wedge C \wedge D \wedge E).$$

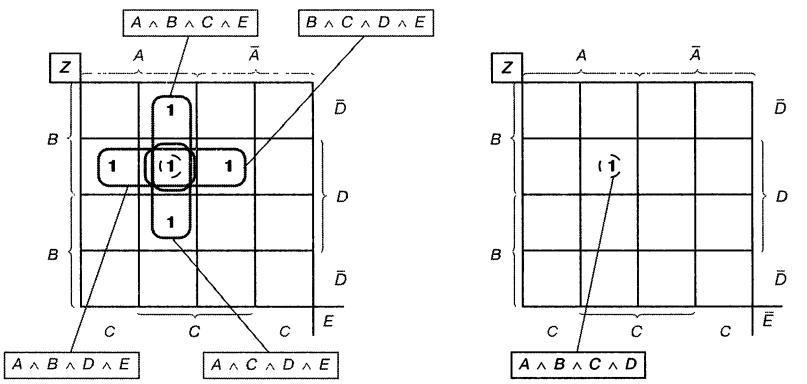


Рис. 5.60. Карта Карно пороговой схемы.

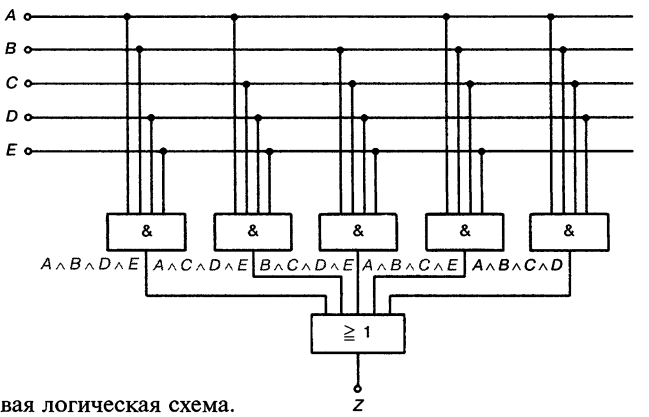


Рис. 5.61. Пороговая логическая схема.

Схема для упрощенной функции представлена на рис. 5.61. Данное уравнение можно еще упростить с помощью алгебры логики. Для первых трех полных конъюнкций можно вынести за скобки $(A \wedge B)$, для двух последних — $(C \wedge D)$. Получается функция:

$$Z = \left[(A \wedge B) \wedge \left((C \wedge E) \vee (D \wedge E) \wedge C \wedge D)\right)\right] \vee \left[(C \wedge D) \wedge \left((A \wedge E) \vee (B \wedge E)\right)\right].$$

Все-таки существенного упрощения добиться не удалось.

5.5.6. Схема сравнения (компаратор)

В цифровой технике часто нужно сравнить цифровые данные друг с другом. Самая простая схема сравнения, так называемый **компаратор**, сравнивает состояние двух переменных друг с другом.

Пусть переменные обозначены A и B. A и B могут быть равны. A может быть больше, чем B и наоборот. Компаратор имеет для этих трех возможных вариантов три выхода. Они обозначаются X, Y и Z и их состояния присваиваются следующим образом:

$$A = B \Rightarrow X = 1;$$

$$A > B \Rightarrow Y = 1;$$

$$A < B \Rightarrow Z = 1.$$

Итак, необходимо синтезировать схему с двумя входными переменными A и B и с выходными переменными X, Y и Z.

При формировании таблицы истинности следует соблюдать правила: A больше, чем B, если $A = 1$ и $B = 0$. Соответственно B больше, чем A, если $B = 1$ и $A = 0$. Таблица истинности показана на рис. 5.62.

Из таблицы истинности получаются логические функции:

$$X = \left(\overline{A} \wedge \overline{B}\right) \vee \left(\overline{A} \wedge B\right);$$

$$Y = A \wedge \overline{B};$$

$$Z = \overline{A} \wedge B.$$

Эти уравнения далее не упрощаются. Искомая схема показана на рис. 5.63.

			A = B	A > B	A < B
Вар	B	A	X	Y	Z
1	0	0	1	0	0
2	0	1	0	1	0
3	1	0	0	0	1
4	1	1	1	0	0

Рис. 5.62. Таблица истинности компаратора.

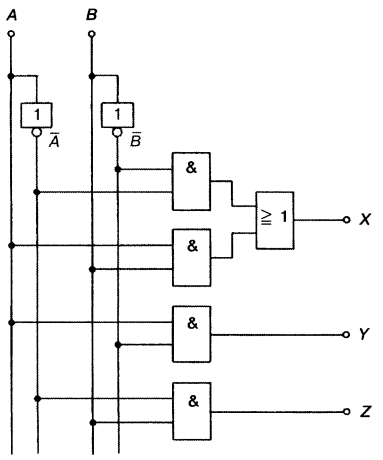


Рис. 5.63. Схема компаратора.

5.5.7. Схема сортировки транзисторов

Перед отправкой с завода транзисторы проверяются на соответствие четырех важных параметров A, B, C и D диапазону допустимых значений. Для измерения применяют четыре цифровых датчика. Датчик выдает 1, если измеряемая величина находится в пределах диапазона допустимых значений. Если измеряемая величина находится вне диапазона допустимых значений, то датчик выдает 0.

Сортировка транзисторов происходит с помощью цифровой схемы. Если все четыре величины находятся в пределах диапазона допустимых значений, на выходе переменная M получает состояние 1. Если только B находится вне диапазона допустимых значений, то выходная переменная N получает состояние 1. Если только B и D находятся вне диапазона допустимых значений, то выходная переменная U получает состояние 1. Во всех других случаях выход $Z = 1$, что означает, что транзистор является бракованным.

Требуется рассчитать схему и построить ее только на элементах И-НЕ (также говорят «в базисе И-НЕ»).

На вход поступают четыре переменные A, B, C и D. Выходными переменными являются M, N, U и Z. M становится равной 1, если $A = 1$, $B = 1$, $C = 1$ и $D = 1$. Это вариант 16 в таблице истинности (рис. 5.64). N будет 1, если $A = 1$, $B = 0$, $C = 1$ и $D = 1$ (вариант 14). U будет равна 1, если $A = 1$, $B = 0$, $C = 1$ и $D = 0$ (вариант 6). Во всех остальных случаях, кроме 6, 14 и 16, $Z = 1$.

Вар.	D	C	B	A	M	N	U	Z	Z
1	0	0	0	0				1	
2	0	0	0	1				1	
3	0	0	1	0				1	
4	0	0	1	1				1	
5	0	1	0	0				1	
6	0	1	0	1	–	–	–	+	–	1
7	0	1	1	0				1	
8	0	1	1	1				1	
9	1	0	0	0				1	
10	1	0	0	1				1	
11	1	0	1	0				1	
12	1	0	1	1				1	
13	1	1	0	0				1	
14	1	1	0	1	–	1	–	–	1
15	1	1	1	0				1	
16	1	1	1	1	1	–	–	–	1

Рис. 5.64. Таблица истинности для схемы сортировки транзисторов. Для большей наглядности нули для выходной переменной не записаны.

В результате получаются следующие логические функции:

$$M = A \wedge B \wedge C \wedge D;$$

$$N = A \wedge \bar{B} \wedge C \wedge D;$$

$$U = A \wedge \bar{B} \wedge C \wedge \bar{D}.$$

Функция Z содержит 13 полных конъюнкций. Z всегда тогда 1, если ни M, ни N, ни U не равняются 1. Лучше записать нормальную форму ИЛИ для $\bar{Z}$ (см. рис. 5.64):

$$\bar{Z} = (A \wedge B \wedge C \wedge D) \vee (A \wedge \bar{B} \wedge C \wedge D) \vee (A \wedge \bar{B} \wedge C \wedge \bar{D});$$

$$\bar{Z} = M \vee N \vee U.$$

Тогда для прямого значения Z:

$$Z = \overline{M \vee N \vee U}.$$

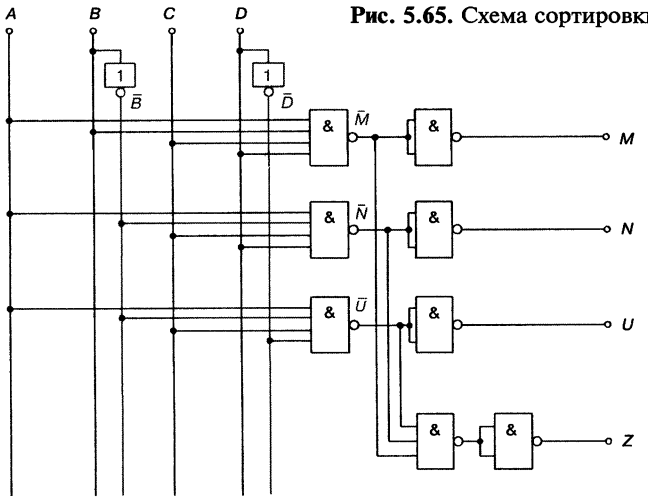


Рис. 5.65. Схема сортировки транзисторов.

Найденные функции для M, N и U упростить нельзя. Они должны быть пересчитаны вместе с уравнением для Z на базис И-НЕ:

$$M = \overline{\overline{A \wedge B \wedge C \wedge D}};$$

$$N = \overline{\overline{A \wedge \bar{B} \wedge C \wedge D}};$$

$$U = \overline{\overline{A \wedge \bar{B} \wedge C \wedge \bar{D}}};$$

$$Z = \overline{M \vee N \vee U} = \bar{M} \wedge \bar{N} \wedge \bar{U};$$

$$Z = \overline{\overline{\bar{M} \wedge \bar{N} \wedge \bar{U}}}.$$

Из этих уравнений получается схема, представленная на рис. 5.65.

Посредством выходов M, N, U и Z может управляться механическое устройство, которое распределяет транзисторы в 4 различные контейнера.

5.6. Задания по схемотехническому проектированию

5.6.1. Схема управления

Требуется составить схему управления, удовлетворяющую таблице истинности (рис. 5.66). Запишите нормальную форму ИЛИ для X, Y и Z и максимально упростите ее с помощью диаграммы Карно. Найденные уравнения для X, Y и Z нужно пересчитать в базисе ИЛИ-НЕ. Нарисуйте соответствующую схему.

Вар.	D	C	B	A	X	Y	Z
1	0	0	0	0	1	0	0
2	0	0	0	1	1	0	0
3	0	0	1	0	1	0	0
4	0	0	1	1	1	0	0
5	0	1	0	0	0	1	1
6	0	1	0	1	0	1	1
7	0	1	1	0	1	0	0
8	0	1	1	1	0	1	1
9	1	0	0	0	0	0	1
10	1	0	0	1	0	0	1
11	1	0	1	0	1	0	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	1	1
14	1	1	0	1	0	1	1
15	1	1	1	0	1	0	0
16	1	1	1	1	0	1	1

Рис. 5.66. Таблица истинности схемы управления.

5.6.2. Схема контроля нечетности

Схемой контроля нечетности называется схема, в которой выход равен 1 тогда, когда нечетное число входов имеют состояние 1. Схема должна иметь 3 входа. Требуется синтезировать максимально простую схему в базисе И-НЕ.

5.6.3. Мажоритарная схема

На выходе мажоритарной схемы состояние 1 тогда, когда большинство входов имеют состояние 1.

Составьте таблицу истинности для мажоритарной схемы с пятью входами. Запишите нормальную форму ИЛИ согласно таблице истинности и попробуйте максимально ее упростить. Изобразите как можно более простую схему на основных элементах.

Для реализации схемы в наличии имеются только элементы ИЛИ-НЕ. Найденные уравнения нужно преобразовать так, чтобы схему можно было построить только на элементах ИЛИ-НЕ (в базисе ИЛИ-НЕ).

5.6.4. Схема блокировки

Автомат по выдавливанию пластмассы может быть включен в производственный цикл только при условии, что режим выдавливания включен, датчик уровня показывает достаточное количество пластмассы в баке, аварийные датчики не показывают состояние аварии, датчики температуры показывают температуру плавления пластмассы и что так называемый цикл очистки не запущен.

Для запуска цикла очистки необходимо, чтобы режим цикла очистки был включен, датчик уровня показывал отсутствие пластмассы в баке, аварийные датчики показывали отсутствие состояния аварии и режим выдавливания не включен. Температура формы может быть любая.

Проверка названных условий должна происходить с помощью логической схемы. Оговорим для переменных величин состояния 0 и 1:

Режим выдавливания включен	$A = 1$
Датчик уровня показывает достаточное количество пластмассы	$F = 1$
Авария	$L = 0$
Датчики температуры показывают необходимую температуру	$B = 1$
Режим очистки включен	$C = 1$
Автомат может работать в режиме выдавливания	$Z = 1$
Автомат может работать в режиме очистки	$R = 1$

Искомая схема имеет входные переменные A, F, L, B, C и выходные переменные Z и R (рис. 5.67).

Она работает как так называемая схема блокировки, т. е. определенные рабочие режимы становятся допустимыми при выполнении определенных условий.

Рис. 5.67. Входы и выходы искомой цифровой схемы.

Требуется синтезировать максимально простую схему, удовлетворяющую описанному выше техническому заданию. Схема должна быть реализована в базисе И-НЕ.

5.6.5. Анализатор показаний радаров

Четыре радара передают в цифровую схему логические сигналы A, B, C и D, которые могут быть равны 1 или 0.

Сигнал A первого радара имеет состояние 1, если он фиксирует самолет. Второй радар выдает $B = 1$, если самолет летит по направлению к радарам на высоте не менее 2000 м. Третий радар фиксирует изменение курса. Он выдает $C = 0$, если курс самолета меняется в течение интервала времени Δt. Если одновременно еще один самолет влетает в воздушное пространство, четвертый радар выдает $D = 1$.

Схема должна работать следующим образом. На выходе Z появляется сигнал 1, если зафиксирован самолет, который летит к радарам на высоте до 2000 м с неизменным курсом и в зоне действия радаров не наблюдается одновременно еще один самолет. Или радар фиксирует самолет, летящий на высоте более 2000 м, с неизменным в течение времени Δt курсом, но в зоне действия радаров зафиксирован еще один самолет.

Требуется синтезировать максимально простую схему, удовлетворяющую описанному выше техническому заданию. Схема должна быть реализована в базисе ИЛИ-НЕ.

Контрольный тест

1. Что такое полная конъюнкция?
2. Как образуются нормальные формы ИЛИ? Приведите пример.
3. Чем отличается нормальная форма И от нормальной формы ИЛИ?
4. Таблица истинности для некоторой схемы изображена на рис. 5.68. Запишите нормальную форму ИЛИ, соответствующую этой таблице истинности.

Вар.	C	B	A	Z
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0

Рис. 5.68. Таблица истинности.

5. Изобразите карту Карно для переменных K, M, S и R.
6. Какие правила соседства действуют для карт Карно с четырьмя переменными?

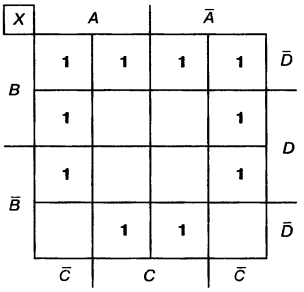


Рис. 5.69. Карта Карно с занесенной нормальной формой ИЛИ.

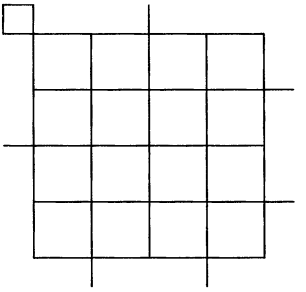


Рис. 5.70. Карта Карно.

7. Упростите с помощью алгебры логики следующую нормальную форму ИЛИ и проверьте полученное упрощение с помощью карты Карно:

$$Z = (A \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge C) \vee (\bar{A} \wedge \bar{B} \wedge C) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C}).$$

8. На карте Карно (рис. 5.69) представлена нормальная форма ИЛИ. Максимально упростите ее и запишите упрощенную логическую функцию.

9. Занесите в карту Карно функцию (рис. 5.70):

$$Z = (\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D}) \vee (A \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B}).$$

Выражения $(A \wedge \bar{C})$ и $(\bar{A} \wedge \bar{B})$ представьте как группы.

10. Как организована карта Карно для шести переменных?

ГЛАВА 6

СЕМЕЙСТВА СХЕМ

6.1. Общие сведения

Логические элементы, называемые также вентилями, строятся почти всегда на полупроводниковой базе. Представление логических элементов в виде переключателей-реле, примененное в разделе 2, служит только для лучшего понимания и на практике имеет небольшое значение. В наше время логические элементы в виде реле используются в основном для схем управления контакторами в силовых цепях с большими токами. **Контактор** — это мощное реле для коммутации в сетях с напряжением 220 В и выше.

Итак, современные логические элементы состоят из полупроводников. Известно, что транзистор может работать в качестве бесконтактного переключателя. Такой бесконтактный переключатель может быть построен как на биполярных, так и на полевых транзисторах ИЛИ на полупроводниковых диодах. Таким образом, существует много вариантов изготовления логических элементов из полупроводников.

> Логические элементы, построенные по одному определенному принципу, образуют семейство схем.

Логические элементы одного семейства совместимы друг с другом и без проблем собираются в схемы. Для сборки цифровой схемы применяют нужные элементы одного семейства. У таких элементов унифицированы напряжения питания и одинаковые уровни сигналов. Времена переключения отдельных элементов равны с некоторым допуском.

Логические элементы из различных семейств комбинируются друг с другом только при определенных условиях. Часто для их совместимости приходится применять специальные промежуточные элементы.

Первые полупроводниковые логические элементы строились на дискретных полупроводниковых элементах. Под **дискретным элементом** понимают единичную деталь в корпусе с выводами, то есть обыкновенные диоды, транзисторы и другие комплектующие изделия. Не дискретные элементы — это полупроводники, собранные вместе в интегральных микросхемах, — комбинации из многих транзисторов или диодов.

Значение семейства схем, построенных на дискретных элементах, в последнее время сильно уменьшилось. Такие схемы имеют относительно большие габариты, и их производство обходится дороже, чем производство интегральных микросхем. Их единственным преимуществом является легкость самостоятельной сборки.

Схемы семейства релейно-транзисторной логики (РТЛ) состоят из сопротивлений и биполярных транзисторов (рис. 6.1). Еще одним «дискретным» семейством схем является система DCTL (Direct Coupled Transistor Logic), состоящая из последовательно включенных биполярных транзисто-

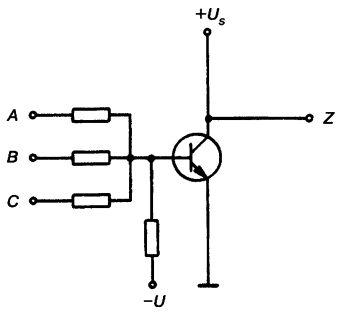


Рис. 6.1. Схема РТЛ.

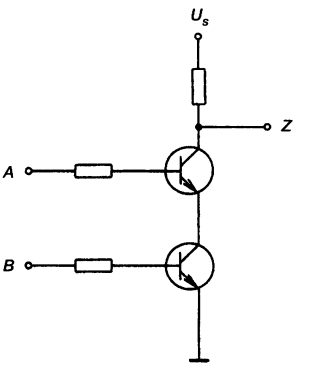


Рис. 6.2. Схема DCTL (Direct Coupled Transistor Logic).

ров (рис. 6.2). Оба этих семейства устарели и в настоящее время практически не применяются.

Большее значение сегодня имеет система ДТЛ. ДТЛ сокращенно означает диодно-транзисторная логика. Этому семейству посвящен разд. 6.5.

Очень большое значение имеют схемы семейства ТТЛ. ТТЛ сокращенно означает транзисторно-транзисторная логика. Элементы этих схем построены на интегральных микросхемах из биполярных транзисторов (разд. 6.6).

Следующее важное семейство называется ЭСЛ. ЭСЛ сокращенно означает эмиттерно-связанная логика (разд. 6.7). Транзисторы имеют общее эмиттерное сопротивление. Схемы этого семейства также построены на интегральных микросхемах из биполярных транзисторов.

В схемах семейства МОП транзисторной логики (МОП — металл, оксид, проводник) (разд. 6.8) применяются интегральные микросхемы из полевых транзисторов с каналами *N* или *P* типа (см. Бойт, Электроника, ч. 2). Если в одном элементе присутствуют МОП-транзисторы с каналами *N* и *P* типа, то говорят о комплементарной МОП-технологии. Такое семейство называется КМОП логика (подразд. 6.8.4).

6.2. Бинарные уровни напряжения

Логические элементы производятся в виде электронных схем. Но электронные схемы «не понимают» никакой цифровой логики. Они реагируют на напряжение на входе, на соответствующие токи и имеют на выходе определенные напряжения. То есть они работают «электрически». Эта мысль лежит в основе идеи объяснить принцип действия всех цифровых схем электрически — независимо от каких-либо логических взаимосвязей.

Можно составить таблицу, аналогичную таблице истинности, и в эту таблицу записать напряжения. Рассмотрим схему на рис. 6.3. Если ко входу *A* приложено напряжение +5 В, то диод D_1 начнет проводить в прямом направлении. На диоде падает напряжение 0,7 В (Si-диод). На выходе *Z* появляется напряжение 4,3 В. Напряжение 4,3 В также будет на выходе, если на *B* или на оба входа подать напряжение +5 В (рис. 6.4).

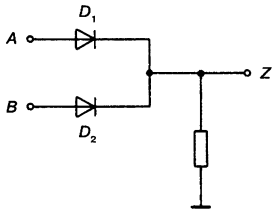


Рис. 6.3. Логическая цифровая схема.

Вар.	B	A	Z
1	0 В	0 В	0 В
2	0 В	+5 В	+4,3 В
3	+5 В	0 В	+4,3 В
4	+5 В	+5 В	+4,3 В

Рис. 6.4. Рабочая таблица с напряжениями.

Представленная на рис. 6.4 таблица называется рабочей согласно DIN 40700, часть 14. Она не может называться таблицей истинности, так как она не дает никакой информации о логических операциях.

Схема на рис. 6.3 может быть также запитана другим напряжением, например 4 В или 8 В. Тогда следует воспользоваться рабочими таблицами (рис. 6.5). Задание значений напряжения в рабочих таблицах является излишне громоздким. Также часто не всегда точно определено, каким напряжением питать схему. Можно выбрать любое напряжение в пределах допустимого диапазона. Поэтому целесообразно в рабочих таблицах различать только лишь ВЫСОКИЙ и НИЗКИЙ уровни напряжения. ВЫСОКИЙ уровень напряжения обозначается символом *H* (от англ. «High» — высокий), НИЗКИЙ уровень напряжения обозначается символом *L* (от англ. «Low» — низкий). *H* и *L* являются уровнями напряжения.

> *L* = Low = низкий уровень напряжения.

Вар.	B	A	Z
1	0 В	0 В	0 В
2	0 В	4 В	3,3 В
3	4 В	0 В	3,3 В
4	4 В	4 В	3,3 В

Вар.	B	A	Z
1	0 В	0 В	0 В
2	0 В	8 В	7,3 В
3	8 В	0 В	7,3 В
4	8 В	8 В	7,3 В

Рис. 6.5. Рабочая таблица с напряжениями.

Уровень, стремящийся к минус бесконечности ($-\infty$).

> *H* = High = высокий уровень напряжения.

Уровень, стремящийся к плюс бесконечности ($+\infty$).

Вар.	B	A	Z
1	*L*	*L*	*L*
2	*L*	*H*	*H*
3	*H*	*L*	*H*
4	*H*	*H*	*H*

Рис. 6.6. Рабочая таблица с уровнями.

Для схемы на рис. 6.3 получается представленная на рис. 6.6 рабочая таблица с указанием уровней. Цифровые схемы могут работать с разными напряжениями. Возможные уровни напряжений показаны на рис. 6.7.

H-уровень может колебаться в пределах определенного диапазона напряжения согласно данным производителя схемы. Также и *L*-уровень может колебаться в пределах некоторого диапазона. Эти диапазоны называются диапазонами напряжения высокого и низкого уровней (рис. 6.8).

> *L* и *H* не являются логическими состояниями, это бинарные уровни напряжения. Они являются частью технических данных схемы.

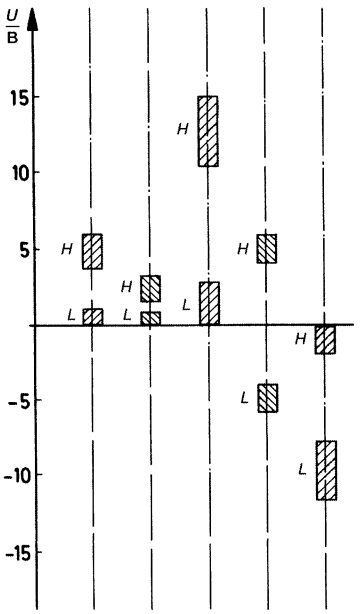


Рис. 6.7. Возможные диапазоны напряжений для уровней *L* и *H*.

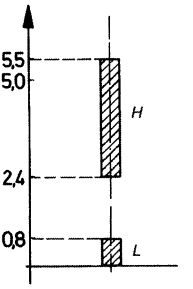


Рис. 6.8. Диапазоны напряжений для уровней *L* и *H*.

Определить, какую логическую операцию выполняет схема, можно только после приведения уровней *L* и *H* в соответствие с логическими состояниями 0 и 1.

6.3. Положительная и отрицательная логика

ВЫСОКИЙ и НИЗКИЙ уровни *L* и *H* могут соответствовать логическим состояниям 0 и 1 двумя различными способами:

L ≙ 0
H ≙ 1

(положительная логика)

L ≙ 1
H ≙ 0

(отрицательная логика)

Если состояние 1 определяет ВЫСОКИЙ уровень, а состояние 0 определяет НИЗКИЙ уровень, то такая логика называется положительной.

В современной цифровой технике работают преимущественно с положительной логикой. Если к схеме нет соответствующего примечания, считается, что она использует положительную логику.

Если состояние 1 определяет НИЗКИЙ уровень, а состояние 0 определяет ВЫСОКИЙ уровень, то такая логика называется отрицательной.

Отрицательная логика имела большое значение во времена существования только одних PNP-транзисторов. При отрицательном напряжении U_{CE} на выходе транзисторного каскада получалось отрицательное напряжение.

Пример

$$0 \triangleq -0{,}3\ \text{В} = H$$

$$1 \triangleq -6\ \text{В} = L$$

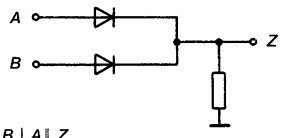
Отрицательная логика используется сегодня только в специальных схемах из соображений помехоустойчивости.

Какие логические операции производит схема на рис. 6.3 при положительной логике, а какие — при отрицательной?

Схема и соответствующая рабочая таблица представлены на рис. 6.9. Из рабочей таблицы определяется таблица истинности. При положительной логике ВЫСОКИЙ уровень H определяет логическое состояние 1, а НИЗКИЙ уровень L — логическое состояние 0 (рис. 6.10). При положительной логике схема производит логическое сложение ИЛИ.

При отрицательной логике ВЫСОКИЙ уровень H определяет логическое состояние 0, а НИЗКИЙ уровень L — логическое состояние 1 (рис. 6.11). Схема производит логическое умножение И. В таблице истинности меняется лишь последовательность вариантов.

При переходе от положительной логики к отрицательной и наоборот меняется тип логической операции.



Вар	B	A	Z
1	L	L	L
2	L	H	H
3	H	L	H
4	H	H	H

Рис. 6.9. Схема с рабочей таблицей.

Вар.	B	A	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

$L \triangleq 0$
$H \triangleq 1$

Рис. 6.10. Таблица истинности при положительной логике.

Вар.	B	A	Z
1	1	1	1
2	1	0	0
3	0	1	0
4	0	0	0

$L \triangleq 1$
$H \triangleq 0$

Рис. 6.11. Таблица истинности при отрицательной логике.

Логический элемент НЕ всегда работает как инвертор — и при положительной, и при отрицательной логике (рис. 6.12).

Пример

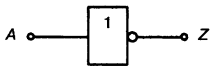
Схема работает при положительной логике как элемент И-НЕ. Какую логическую операцию производит схема при отрицательной логике?

Таблица истинности элемента И-НЕ представлена на рис. 6.13. Из нее можно определить рабочую таблицу с уровнями H и L. При положительной логике ВЫСОКИЙ уровень H определяет логическое состояние 1, а НИЗКИЙ уровень L — логическое состояние 0 (рис. 6.14).

Работа схемы при отрицательной логике показана в таблице истинности на рис. 6.15. Она получилась из рабочей таблицы, в которой теперь ВЫСОКИЙ уровень H определяет логическое состояние 0, а НИЗКИЙ уровень L — логическое состояние 1.

Получается логическая операция ИЛИ-НЕ.

Схема, которая при положительной логике выполняет операцию И-НЕ, при отрицательной логике выполняет операцию ИЛИ-НЕ.



Вар	A	Z
1	0	1
2	1	0

Положительная логика

$0 \triangleq L$

$1 \triangleq H$

Вар	A	Z
1	L	H
2	H	L

Отрицательная логика

$0 \triangleq H$

$1 \triangleq L$

Вар.	A	Z
1	H	L
2	L	H

Рис. 6.12. Работа логического элемента НЕ при положительной и отрицательной логике.

Вар.	B	A	Z
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

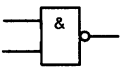


Рис. 6.13. Таблица истинности логического элемента И-НЕ.

Вар.	B	A	Z
1	L	L	H
2	L	H	H
3	H	L	H
4	H	H	L

Рис. 6.14. Рабочая таблица.

Вар.	B	A	Z
1	1	1	0
2	1	0	0
3	0	1	0
4	0	0	1

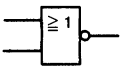


Рис. 6.15. Таблица истинности логического элемента ИЛИ-НЕ.

6.4. Свойства схем

Схемы каждого семейства имеют общие типичные свойства. На основе этих свойств для решения определенных задач выбираются схемы определенного семейства.

Важными свойствами, например, являются быстродействие и помехоустойчивость. Для схемы управления лифтом неважно, срабатывает схема в течение 0,1 мкс или 0,5 мкс. Важно, чтобы не происходило ложного срабатывания. Поэтому целесообразно выбрать в данном случае медленную, но более помехоустойчивую схему. Для компьютера желательны схемы одновременно и быстродействующие, и помехоустойчивые. Два этих качества являются взаимоисключающими, поэтому ищется компромисс между скоростью и помехоустойчивостью.

6.4.1. Потребляемая мощность

Большие цифровые схемы потребляют много энергии. Если отдельный логический элемент потребляет всего 10 мВт, то схема из 100 000 элементов — уже 1 кВт. Компьютерам же с их 10^6 элементами в таком случае требуется 10 кВт — об аккумуляторных батареях придется забыть.

Если ограничить мощность питания отдельных логических элементов, то это скажется на скорости работы схемы и ее помехоустойчивости. Время переключения будет больше, и вследствие вынужденного использования более низких уровней напряжения снизится помехоустойчивость.

Элементы различных семейств сильно различаются по своему энергопотреблению. При обсуждении семейств в разделах 6.5—6.8 мы еще вернемся к этому вопросу.

6.4.2. Диапазон уровней и передаточная характеристика

Для малого энергопотребления должно быть низкое напряжение питания. Напряжение питания определяет ВЫСОКИЙ уровень *H*. НИЗКИЙ уровень *L* определяется падением напряжения на открытых диодах и транзисторах.

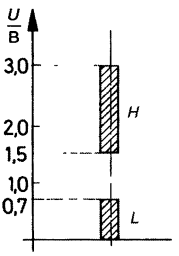


Рис. 6.16. Логические уровни для напряжения питания схемы 3 В.

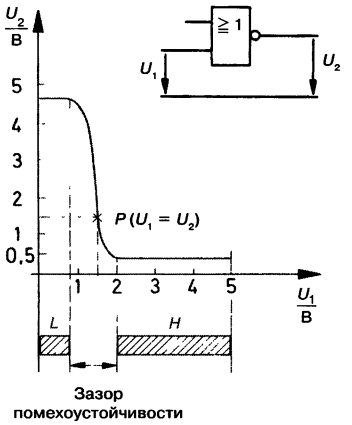


Рис. 6.17. Передаточная характеристика.

Если выбрать напряжение питания схемы 3 В, то ВЫСОКИЙ уровень *H* равен примерно 3 В. При подключении нагрузки на выход схемы и протекании тока по элементам ВЫСОКИЙ уровень *H* падает. Можно максимально допустить падение до 1,5 В, при дальнейшем падении напряжения зазор между ВЫСОКИМ и НИЗКИМ уровнями станет недопустимо мал (рис. 6.16). Область НИЗКОГО уровня *L* определяется падением напряжения на открытых диодах и транзисторах и находится в пределах от 0 В до +0,7 В.

Для основных семейств типовым является напряжение питания 5 В и выше. Для схем указывается так называемая передаточная характеристика, изображенная на рис. 6.17. На вертикальной оси отложено выходное напряжение U_2, на горизонтальной оси — входное напряжение U_1.

По передаточной характеристике можно определить *H*- и *L*-уровни.

НИЗКИЙ уровень *L* согласно характеристике рис. 6.17 мог бы занимать диапазон от 0 В до 1,5 В (точка *P*), а ВЫСОКИЙ уровень *H*- от 1,5 В до 5 В, если бы не необходимость в зазоре помехоустойчивости. Для лучшей помехоустойчивости этот зазор должен быть как можно большим. Без зазора помехи в виде импульсов напряжения могли бы вызывать самопроизвольные переключения уровней с *H* на *L* и наоборот. Кроме того, передаточная характеристика зависит от температуры и тока нагрузки. Точка *P* может перемещаться. Обычно в качестве ширины зазора помехоустойчивости выбирается длина участка падения U_1 характеристики. Таким образом, допустимый диапазон НИЗКОГО уровня *L* равен от 0 В до 0,8 В, а ВЫСОКОГО *H* — от 2 В до 5 В (рис. 6.18). Производители обычно немного уменьшают допустимые диапазоны уровней напряжений с целью повышения помехоустойчивости.

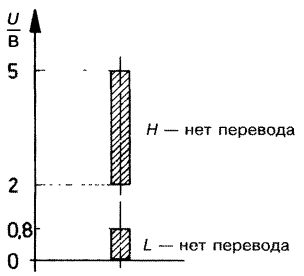


Рис. 6.18. Диапазоны уровней для передаточной характеристики рис. 6.17.

6.4.3. Время переключения

Скорость работы схемы определяется временем переключения ее логических элементов. Различают понятия *быстродействия* логического элемента, т. е. время реакции элемента на изменение сигнала на входах t_P, и **время нарастания сигнала** — t_T.

> Быстродействие t_{PLH} характеризует время задержки выходного сигнала по отношению к входному при изменении состояния выхода с *L* на *H*.

Соответственно t_{PHL} характеризует время задержки выходного сигнала по отношению к входному при изменении состояния выхода с *H* на *L*.

Для измерения быстродействия используют относительный уровень 1,5 В. Рис. 6.19 показывает, что быстродействие t_{PLH} характеризует время, за которое на выходе появляется напряжение входа 1,5 В. Характеристика для t_{PHL} представлена на рис. 6.20. Среднее быстродействие t_P определяется следующим образом:

$$t_P = \frac{t_{PLH} + t_{PHL}}{2}.$$

Вместо термина «быстродействие» также применяется термин «время задержки сигнала».

Переходное время сигнала относится только к выходу элемента. Оно характеризует крутизну фронтов выходного напряжения.

> Время нарастания сигнала t_{TLH} характеризует время, необходимое для возрастания выходного напряжения с 10 % до 90 % разницы между *L* и *H*.

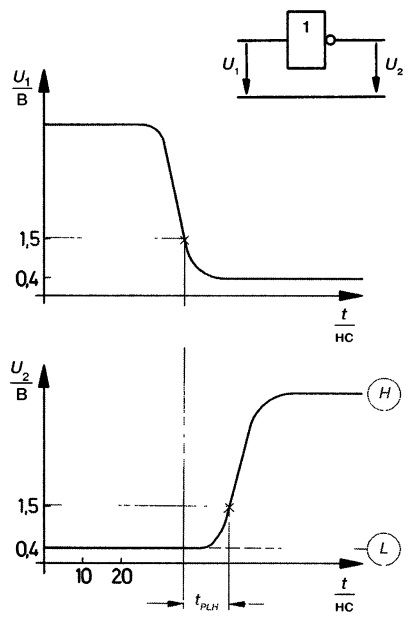


Рис. 6.19. Быстродействие t_{PLH}.

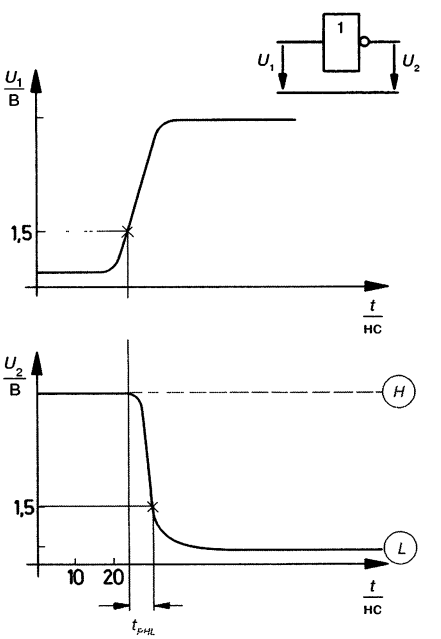


Рис. 6.20. Быстродействие t_{PHL}.

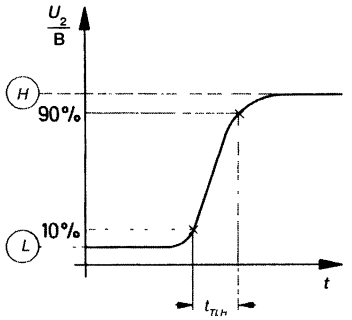


Рис. 6.21. Переходное время сигнала t_{TLH}.

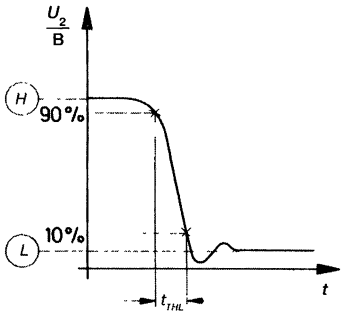


Рис. 6.22. Переходное время сигнала t_{THL}.

Время t_{TLH} показано на рис. 6.21. Время нарастания сигнала t_{THL} характеризует время, необходимое для уменьшения выходного напряжения с 90 до 10 % разницы между L и H (рис. 6.22).

6.4.4. Нагрузочная способность

Для управления логическими элементами требуются определенные напряжения и токи. К выходу элемента может присоединяться только определенное количество входов. Если подключить больше входов, то выходной уровень недопустимо снизится. Элементы будут перегружены, а схема работать не будет.

Различают два нагрузочных коэффициента: входной нагрузочный коэффициент и коэффициент разветвления по выходу. Для каждого семейства схем определена номинальная входная токовая нагрузка элементов. Для ТТЛ-элементов:

Уровень L на входе 0,4 В — 1,6 мА

Уровень H на входе 2,4 В — 40 мкА.

> Входной нагрузочный коэффициент логического элемента равен единице (F_I = 1), если вход элемента потребляет номинальную мощность.

Специальные входы могут потреблять двойную или тройную номинальную мощность. Тогда их нагрузочный коэффициент равен двум и соответственно трем. Такие нагрузочные коэффициенты встречаются прежде всего в интегральных схемах с высокой плотностью элементов.

> Под коэффициентом разветвления логического элемента по выходу F_O понимают количество нормальных входов других элементов, которое может быть подключено к его выходу.

Нормальным для стандартных логических элементов считается коэффициент разветвления не менее 10. Силовые логические элементы имеют коэффициент разветвления не менее 30.

Пример

Элемент И-НЕ на рис. 6.23 имеет коэффициент разветвления, равный 10. Сколько элементов дополнительно можно подключить к его выходу?

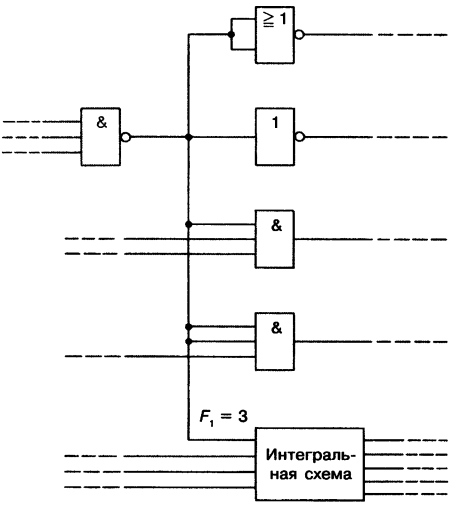


Рис. 6.23. Определение коэффициента разветвления логического элемента.

Верхний элемент ИЛИ-НЕ имеет два объединенных входа. Каждый вход имеет единичный нагрузочный коэффициент. Значит, этот элемент нагружает выход серого элемента двойной номинальной нагрузкой. Аналогичные рассуждения можно применить к нижнему элементу И. Суммарно к выходу элемента И-НЕ подключено шесть одинарных входов и один вход с $F_I = 3$. То есть девять номинальных входных нагрузок.

Можно присоединить еще один вход с $F_I = 1$.

6.4.5. Помехоустойчивость

Помехи в виде импульсов напряжения могут вызывать самопроизвольные ложные переключения выходов логических элементов с *H* на *L* и наоборот с *L* на *H*. Чтобы это не происходило, необходим зазор между допустимыми диапазонами ВЫСОКОГО и НИЗКОГО уровней. Чем больше зазор, тем выше помехоустойчивость схемы.

Различают статическую и динамическую (импульсную) помехоустойчивость.

Статическая помехоустойчивость элемента относится к импульсам напряжения, действующим дольше средней длительности переходных процессов в схеме t_P. К статическим помехам также относятся медленно возрастающие импульсы напряжения.

> Статическая помехоустойчивость характеризует максимально допустимое изменение напряжения на входах элемента, при котором не происходит изменение состояния выхода.

Она указывается для номинального режима работы и для самой неблагоприятной комбинации влияющих на помехоустойчивость факторов. Например, при неудачном соотношении напряжений управляющих и управляемых логических элементов, неблагоприятном сочетании входных сигналов, при неноминальной температуре окружающей среды и при задействованном максимальном коэффициенте разветвления. Такую комбинацию называют случаем худшей помехоустойчивости.

Динамическая помехоустойчивость элемента относится к импульсам напряжения, длящимся меньше средней длительности переходных процессов в схеме t_P.

Соответствующая импульсам энергия, характеризуемая амплитудой и длительностью импульса, не должна превышать определенное предельное

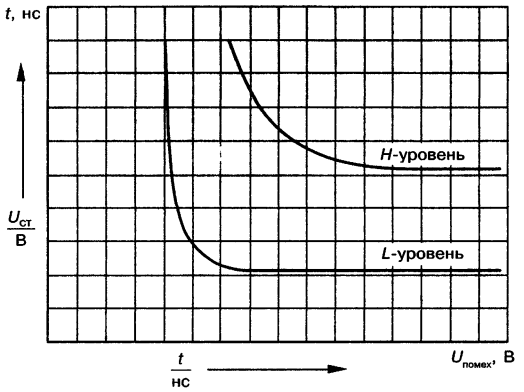


Рис. 6.24. Кривые динамической помехоустойчивости.

значение. Динамическая помехоустойчивость зависит в основном от чувствительности входа логического элемента. Она описывается граничной кривой (рис. 6.24).

> Динамическая помехоустойчивость характеризует длительность помехи по напряжению определенной величины на входах элемента, при которой не происходит изменение состояния выхода.

Точные данные для допустимой амплитуды напряжения помехи и допустимое время ее действия получают из анализа характеристик (рис. 6.24).

При этом существуют кривые для *H*-уровня и *L*-уровня. Если на входе действует состояние *H*, то следует пользоваться кривой для *H*-уровня. Соответственно если на входе действует состояние *L*, то следует пользоваться кривой для *L*-уровня. *H*-уровни более помехоустойчивы, поэтому кривая для *H*-уровня проходит выше.

6.4.6. Проводные логические операции

Если выходы двух элементов связаны гальванически, т. е. просто соединены проводником, то образуется логическая связь, которая в зависимости от внутреннего строения схемы представляет из себя операции И или ИЛИ.

Если выход одного элемента имеет состояние *H*, а другой выход — состояние *L* (рис. 6.25), то состояние точки *Q* не определено. Какое состояние примет эта точка — зависит от технических данных схемы.

Допустим, что уровень *L* соответствует заземлению, 0 В, а уровень *H*-напряжению питания (например +5 В).

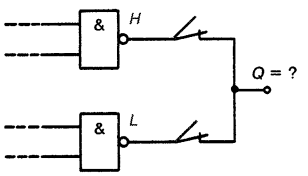


Рис. 6.25. Гальваническая связь выходов двух элементов И-НЕ.

Если выход, на котором действует состояние *L*, имеет малое сопротивление относительно земли, т. е. 0 В, то *H*-состояние другого выхода будет заземлено и точка *Q* примет состояние *L*. Говорят, при этих элементах доминирует уровень *L*.

В этом случае *Q* может иметь состояние *H* только тогда, когда оба выхода имеют состояние *H*. Такое соединение называется **проводное И** (при положительной логике).

Если в схеме доминирует уровень *L*, то при проводной связи на выходе выполняется логическое умножение И — при условии положительной логики.

Проводное И изображено на схемах (рис. 6.26). Для его обозначения возможны два варианта на выбор.

Если выход, на котором действует состояние *H*, имеет малое сопротивление относительно источника питания, то *L*-состояние другого выхода будет поднято до *H*-уровня и точка *Q* (см. рис. 6.25) примет состояние *H*. В этом случае говорят, что при этих элементах доминирует уровень *H*.

Такое соединение называется **проводное ИЛИ** (при положительной логике).

Если в схеме доминирует уровень *H*, то при проводной связи на выходе выполняется логическое сложение ИЛИ — при условии положительной логики.

Два варианта обозначения проводного ИЛИ представлены на рис. 6.27. Точка *Q* может принимать при определенной структуре схемы уровень, находящийся между диапазонами *H*-уровня и *L*-уровня. В таких схемах проводная связь выходов применяться не может. Проводное ИЛИ и проводное И в таких схемах запрещены.

При снижении выходного уровня с *H* на *L* на выходе с предыдущим состоянием *H* может течь недопустимо высокий ток. Также при повышении выходного уровня с *L* на *H* на выходе с предыдущим состоянием *L* может течь недопустимо высокий ток. Логические элементы вследствие этого будут перегружены по току.

Проводные соединения выходов, образующие проводные И и проводные ИЛИ, могут производиться только при разрешении производителя.

В каких случаях разрешаются проводные логические соединения, указывается в технических описаниях. Проводные логические операции ведут к упрощению схемы, к уменьшению времени коммутации и к сокращению производственных расходов.

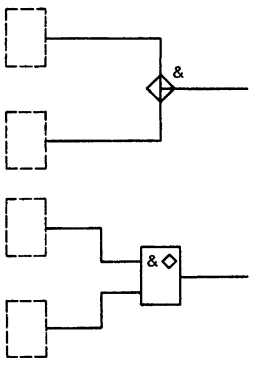


Рис. 6.26. Условное обозначение проводного И согласно DIN 40900, часть 12.

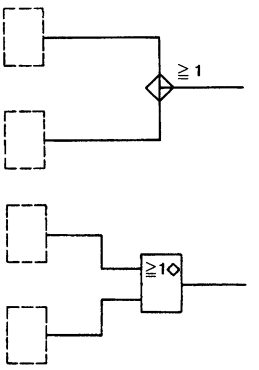


Рис. 6.27. Условное обозначение проводного ИЛИ согласно DIN 40900, часть 12.

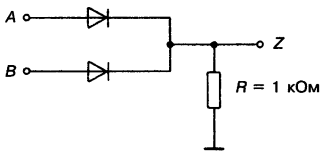
6.5. ДТЛ-схемы

6.5.1. Введение

ДТЛ-схемы реализуются на диодах и транзисторах. Конечно, дополнительно применяются и сопротивления. Термин ДТЛ произошел от английского «Diode Transistor Logic» и переводится как «диодно-транзисторная логика». Схемы этого семейства сначала строились на отдельных элементах, а затем в виде тонкопленочных и толстопленочных (интегральных) микросхем (см. Бойт, Электроника, часть 2). В настоящее время они производятся почти исключительно в виде монолитных микросхем.

6.5.2. Основные ДТЛ-схемы

Одна из трех основных схем семейства ДТЛ показана на рис. 6.28. С этой схемой мы уже познакомились в разд. 6.2. Если хотя бы на одном из входов присутствует ВЫСОКИЙ уровень *H* (высокий положительный потенциал, от англ. *High* — высокий), то он будет действовать и на выходе. При положительной логике схема работает как логический элемент ИЛИ.



Вар	B	A	Z
1	L	L	L
2	L	H	H
3	H	L	H
4	H	H	H

Рис. 6.28. ДТЛ-схема (ИЛИ-элемент при положительной логике).

В схеме, изображенной на рис. 6.29, ВЫСОКИЙ уровень на выходе вентиля сформируется только в том случае, если он будет присутствовать на обоих входах. Если на каком-либо его входе будет НИЗКИЙ уровень *L* (например нулевой потенциал, заземление, от англ. *Low* — низкий), то он будет действовать и на выходе.

Вар.	B	A	Z
1	L	L	L
2	L	H	L
3	H	L	L
4	H	H	H

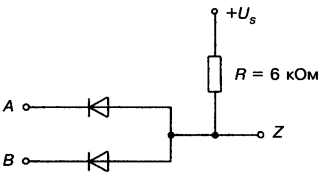


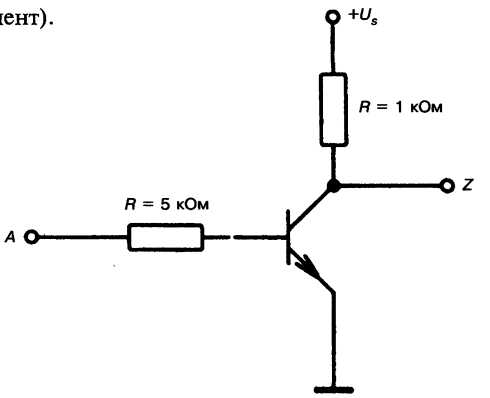
Рис. 6.29. ДТЛ-схема (И-элемент при положительной логике).

Разберем подробнее схему на рис. 6.29. Допустим, что напряжение питания U_S равно 6 В. Напряжение +6 В относится к зоне ВЫСОКИХ уровней. Зона НИЗКИХ уровней (*L*-зона) находится около 0 В (заземление). Если на обоих входах присутствует ВЫСОКИЙ уровень *H*, то диоды заперты, следовательно, на выходе *Z* также будет присутствовать ВЫСОКИЙ уровень (от $+U_S$ через *R*).

Если теперь на вход *B* подать НИЗКИЙ уровень ($\approx$ 0 В), диод в ветви *B* откроется. Ток от источника $+U_S$ потечет через резистор *R* и диод к нулевому потенциалу (заземлению). Величина падения напряжения на диоде составит +0,7 вольта (для кремниевого диода). Такое же напряжение будет приложено к выходу *Z*. Это значение напряжения относится к зоне НИЗ-

Рис. 6.30. ДТЛ-схема (НЕ-элемент).

Вар.	A	Z
1	L	H
2	H	L



КИХ сигналов. Таким образом, выход Z имеет НИЗКИЙ уровень тогда, когда НИЗКИЙ уровень действует на одном из входов. Только если на обоих входах присутствует ВЫСОКИЙ уровень, на выходе Z будет ВЫСОКИЙ уровень. Таким образом, при положительной логике схема работает как логический элемент И.

Схема на рис. 6.30 работает как инвертор, то есть логический элемент НЕ. Если на входе A действует ВЫСОКИЙ уровень, то транзистор открыт. К выходу Z прикладывается напряжение 0,2—0,3 В, которое соответствует НИЗКОМУ уровню. Если на входе A действует НИЗКИЙ уровень, то транзистор запирается. Сопротивление эмиттер-коллектора высокоомно (например, 10 МОм) и почти все напряжение питания прикладывается к выходу, имеющему таким образом ВЫСОКИЙ уровень. Соответствующая таблица истинности представлена на рис. 6.30.

Как ведет себя вход, к которому не приложен ни ВЫСОКИЙ (≈ +6 В), ни НИЗКИЙ (≈ 0 В, заземление) уровень?

На схеме (рис. 6.28) такой открытый вход действует как НИЗКИЙ уровень. При этом на выходе может быть ВЫСОКИЙ уровень только тогда, когда ВЫСОКИЙ уровень действует на одном из входов.

Если вход инвертора, изображенного на схеме 6.30, открыт, то это эквивалентно прикладыванию на вход НИЗКОГО уровня. Ведь транзистор не может проводить с неподключенной базой.

Если открытым остается вход схемы на рис. 6.29, то это соответствует приложению ко входу ВЫСОКОГО уровня. При открытом входе выход не может принять НИЗКИЙ уровень. Если вход A остается открытым, а на входе B действует ВЫСОКИЙ уровень, то на выходе также действует ВЫСОКИЙ уровень H. Логические элементы ИЛИ (см. рис. 6.28) и И (см. рис. 6.29) являются при положительной логике пассивными элементами, т. е. они не содержат в себе активного элемента (усилителя). Если последовательно соединить несколько пассивных логических элементов, то существует опасность, что уровни напряжения выйдут из допустимых областей. Прежде всего, может недопустимо упасть ВЫСОКИЙ уровень. Чтобы избежать этой опасности, применяют так называемые активные элементы, то есть элементы с усилителями.

На рис. 6.31 изображен активный логический элемент, который работает в качестве вентиля И при положительной логике. В точке X находится

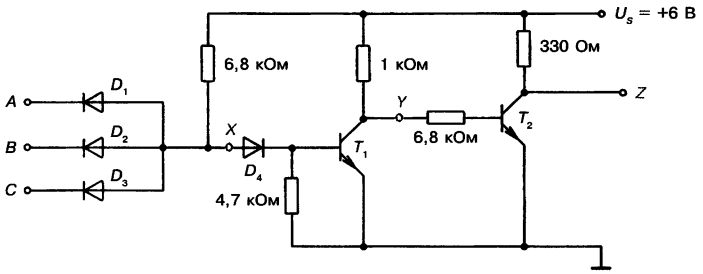


Рис. 6.31. ДТЛ-схема (активный И-элемент при положительной логике).

И-сумматор. Затем подключен инвертор с выходом *Y*, за которым следует еще один инвертор с выходом *Z*. Оба последовательно включенных инвертора компенсируют действие друг друга, так что на выходе *Z* снова оказывается сигнал сумматора И. Таблица истинности для данной схемы при положительной логике приведена на рис. 6.32.

Вар.	*C*	*B*	*A*	*X*	*Y*	*Z*
1	*L*	*L*	*L*	*L*	*H*	*L*
2	*L*	*L*	*H*	*L*	*H*	*L*
3	*L*	*H*	*L*	*L*	*H*	*L*
4	*L*	*H*	*H*	*L*	*H*	*L*
5	*H*	*L*	*L*	*L*	*H*	*L*
6	*H*	*L*	*H*	*L*	*H*	*L*
7	*H*	*H*	*L*	*L*	*H*	*L*
8	*H*	*H*	*H*	*H*	*L*	*H*

Рис. 6.32. Рабочая таблица к схеме рис. 6.31.

Задачей диода D_4 является недопущение отпирания транзистора T_1 при открытых диодах входной цепи или при НИЗКОМ уровне на их входах, когда к точке *X* приложено пороговое падение напряжения на входных диодах (≈ 0,7 В). В отсутствие диода это напряжение могло бы привести к отпиранию транзистора T_1. Этот диод называют **сдвигающим уровень** (сдвиговые диоды). При его введении для открытия транзистора необходимо иметь в точке *X* минимальное напряжение примерно 1,4 вольта (≈ 0,7 В пороговое падение напряжения на диоде D_4 и ≈ 0,7 В пороговое напряжение база-эмиттер транзистора T_1).

Если в схеме на рис. 6.31 убрать один из инверторов, то при положительной логике получается логический элемент И-НЕ. Интересная схема логического элемента И-НЕ ДТЛ изображена на рис. 6.33. Транзистор T_1 работает как эмиттер, то есть усилитель без инвертирования. За усилителем подключен инвертор. Такая ДТЛ-схема (рис. 6.33) применяется особенно часто.

Для обзора имеющихся в настоящее время в продаже ДТЛ-схем следует обратиться к каталогам фирм-производителей. ДТЛ-схемы предлагаются в стандартном исполнении с напряжениями питания от 5 до 6 В (например от Valvo).

Схемы семейства ДТЛ отличаются относительно высоким быстродействием. Быстродействие характеризует время реакции логического элемента на изменение сигналов на входах и составляет для ДТЛ-элементов примерно 30 нс (10^{-9} с). Однако элементы ТТЛ-семейства (см. разд. 6.6) работают в три раза быстрее. Их быстродействие составляет около 10 нс. Это

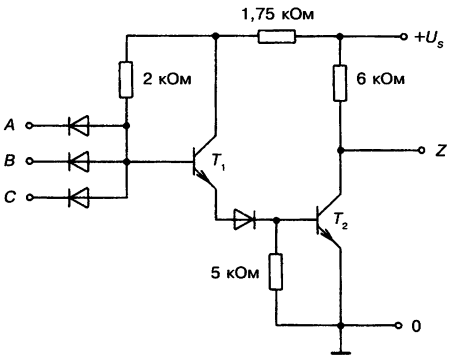


Рис. 6.33. ДТЛ-схема (И-НЕ-элемент при положительной логике).

значит, что ДТЛ-элементы чаще всего применяют там, где не предъявляются особенно высокие требования к быстродействию. Преимуществом ДТЛ-схем перед ТТЛ-схемами является их более высокая помехоустойчивость.

Ниже приведены типичные характеристики для ДТЛ-схемы, изображенной рис. 6.33:

Напряжение питания	6 В
Потребление мощности на каждый логический элемент	9 мВт
Быстродействие	30 нс
Статическая помехоустойчивость	1,2 В
Температурный диапазон	от 0 до +75 °С
Коэффициент нагрузки на входе	1
Коэффициент нагрузки на выходе	8
Входное напряжение ВЫСОКОГО уровня (нижняя граница)	3,6 В
Входное напряжение НИЗКОГО уровня (верхняя граница)	1,4 В
Выходное напряжение ВЫСОКОГО уровня (нижняя граница)	4,0 В
Выходное напряжение НИЗКОГО уровня (верхняя граница)	0,5 В

6.5.3. МПЛ-схемы

С появлением ДТЛ-схем стала развиваться так называемая «медленная, помехоустойчивая логика», сокращенно МПЛ. Сдвиговые диоды на рис. 6.31 и 6.33 заменяются лавинными *Z*-диодами (диод Зенера) (рис. 6.34). Мини-

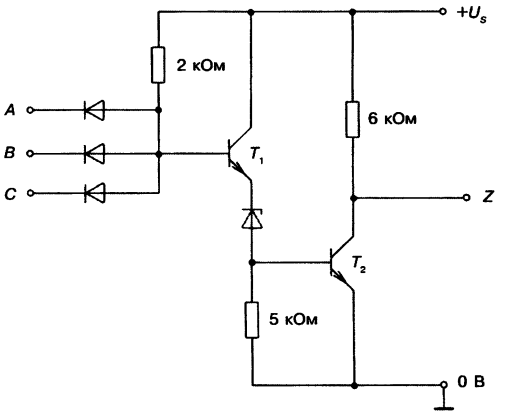


Рис. 6.34. ДТЛ-схема с Z-диодом (И-НЕ-элемент при положительной логике).

мально необходимый входной ВЫСОКИЙ уровень таким образом повышается до напряжения срабатывания *Z*-диода. Наличие *Z*-диода увеличивает время задержки прохождения сигнала через логический элемент. ДТЛ-схемы с диодами Зенера также называются ДТЛЗ схемами.

С повышением напряжения питания можно существенно увеличить ВЫСОКИЙ уровень входного напряжения, таким образом, увеличив «зазор» между ВЫСОКИМ и НИЗКИМ уровнями, обеспечивая большую статическую помехоустойчивость. Также возрастает и динамическая помехоустойчивость благодаря увеличенному времени задержки.

МПЛ-элементы производятся для напряжений питания 12 и 15 В. На рис. 6.35 представлена схема типичного МПЛ-элемента. Входной ВЫСОКИЙ уровень составляет от 7,5 до 15 В, выходной НИЗКИЙ уровень — от 0 до 4,5 В (рис. 6.36).

Типичный ВЫСОКИЙ уровень составляет 14,3 В, типичный НИЗКИЙ уровень — 1,0 В. Чтобы превратить НИЗКИЙ уровень в ВЫСОКИЙ или понизить ВЫСОКИЙ до уровня НИЗКОГО, требуются значительные по-

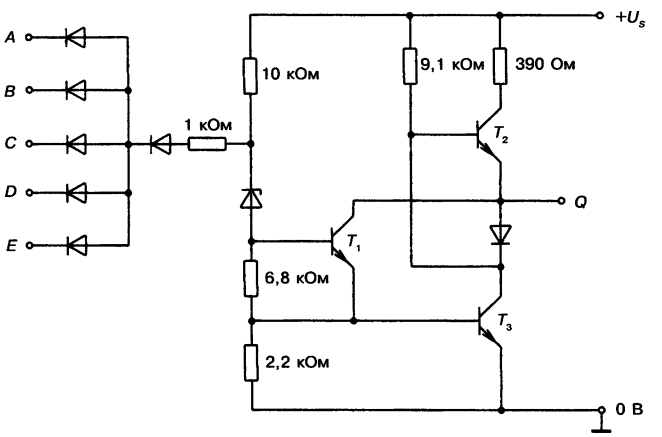


Рис. 6.35. Схема МПЛ-элемента (FZH 125, Siemens).

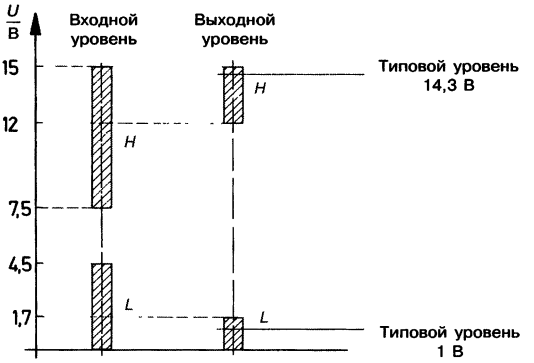


Рис. 6.36. Диапазоны уровней схемы FZH 125.

мехи. Таким образом, схема имеет высокую помехоустойчивость. Время задержки прохождения сигнала составляет примерно 200 нс, что существенно больше, чем у обычных ДТЛ-элементов.

Интегральная микросхема FZH 125 содержит два логических элемента И-НЕ (при положительной логике), каждый с 5 входами. Схема подключения 16-полюсной сдвоенной микросхемы изображена на рис. 6.37.

Схема, представленная на рис. 6.35, имеет так называемый нагрузочный выход. Если транзистор T_3 заперт, то транзистор T_2 отпирается. При запертом транзисторе T_3 на выходе Q действует ВЫСОКИЙ уровень, примерно 14,3 В. Для управления последующими элементами от источника U_S через сопротивление 390 Ом и T_2 течет относительно большой ток. Таким образом, схема может обеспечить ВЫСОКИЙ уровень на большом количестве подключаемых далее элементов.

Если заперт транзистор T_2, то отпирается транзистор T_3. На выходе Q через диод и переход коллектор-эмиттер транзистора T_3 может протекать относительно большой ток без повышения напряжения. Таким образом, схема может питать несколько подключаемых далее логических элементов напряжением НИЗКОГО уровня.

К основным параметрам подобных схем относятся коэффициенты нагрузки на выходе по ВЫСОКОМУ и НИЗКОМУ уровню. Под коэффициентом нагрузки на выходе по ВЫСОКОМУ уровню понимают возможное количество входов подключенных логических элементов с подачей ВЫСОКОГО уровня на вход. Под коэффициентом нагрузки на выходе по НИЗ-

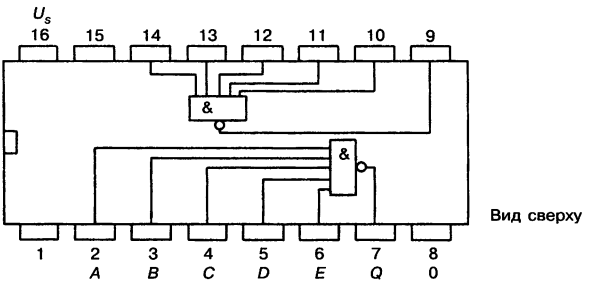


Рис. 6.37. Цоколевка схемы FZH 125.

КОМУ уровню понимают возможное количество входов, подключенных логических элементов с подачей НИЗКОГО уровня на вход.

Технические данные для схемы FZH 125 представлены на рис. 6.38. Кроме информации о напряжении и помехоустойчивости, в таблице есть данные о входном токе на ВЫСОКОМ и НИЗКОМ уровнях. По этим характеристикам можно определить ранее рассмотренное в разд. 6.4.4 потребление тока:

НИЗКИЙ уровень на входе $-I_{IL} = 1$ мА

ВЫСОКИЙ уровень на входе $I_{IH} = 1$ мкА

Статические параметры для диапазона 15 В
в температурных зонах 1 и 5

1: от 0 °C до + 70 °C 5: от −25 °C до +85 °C		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S			13,5	15,0	17,0	В
Входное напряжение *H*-уровня	U_{IH}	$U_S = U_{SB}$	1	7,5			В
Входное напряжение *L*-уровня	U_{IL}	$U_S = U_{SB}$ и U_{SA}	2			4,5	В
Выходное напряжение *H*-уровня	U_{QH}	$U_S = U_{SB}$ и $U_{SA}U_{IL} = 4{,}5$ В, $-I_{QH} = 0{,}1$ мА	2	12,0	14,3		В
Выходное напряжение *L*-уровня	U_{QL}	$U_S = U_{SB}$, $U_{IH} = 7{,}5$ В $I_{QL} = 18$ мА	1		1,0	1,7	В
Статическая помехоустойчивость							
H-сигнал	U_{SS}			4,6	8,0		В
L-сигнал	U_{SS}			2,8	5,0		В
H-входной ток на канал	I_{IH}	$U_S = U_{SA}$, $U_I = U_{IHA}$	3			1,0	мкА
L на канал	$-I_{IL}$	$U_S = U_{SA}$, $U_{IL} = 1{,}7$ В	4		1,0	1,8	мА
Ток короткого замыкания на канал выхода	$-I_Q$	$U_S = U_{SA}$, $U_I = 0$ В	5	15,0	37,0	60,0	мА
H-потребляемый ток на элемент	I_{SH}	$U_S = U_{SA}$, $U_I = 0$ В	6		1,2	2,1	мА
L-потребляемый ток на элемент	I_{SL}	$U_S = U_{SA}$, $U_I = U_{IHA}$	7		2,3	4,0	мА
Потребление мощности на элемент	P	$U_S = U_{SA}$ Скважность 1 : 1			27	52	мВт
Время переключения при $U_S = 15$ В, $F_Q = 1$, $T_U = 25$ °C							
Время прохождения сигнала (быстродействие)	t_{PLH}	$C_L = 10$ пФ при 4,5 В	26		195		нс
	t_{PHL}				140		нс
Переходное время сигнала	t_{TLH}	$C_L = 10$ пФ			410		нс
	t_{THL}				75		нс

Рис. 6.38. Таблица данных схемы FZH 125 (Siemens).

В таблице также приведен выходной ток короткого замыкания, очень большой при относительно высоких напряжениях, поэтому максимальное время короткого замыкания составляет 1 секунду. При превышении этого времени микросхема будет выведена из строя.

Заметим, что потребляемый каждым отдельным логическим элементом ток при ВЫСОКОМ состоянии выхода имеет другое значение, чем при НИЗКОМ состоянии выхода. Типичный потребляемый ток при ВЫСОКОМ состоянии выхода — 1,2 мА, при НИЗКОМ — 2,3 мА. Следовательно, потребляемая мощность зависит от соотношения времен нахождения выхода в ВЫСОКОМ и НИЗКОМ состояниях. Это соотношение называется **импульсным**. Потребляемая каждым логическим элементом мощность указывается при импульсном соотношении 1 : 1.

ДТЛ-схемы семейства медленной помехоустойчивой логики применяются прежде всего для управления двигателями, так как в помещении с приводными двигателями помехоустойчивость схем управления особенно важна. В машинных залах часто наводятся сильные помехи по напряжению.

6.6. ТТЛ-схемы

6.6.1. Строение и принцип действия ТТЛ-элементов

Сокращение ТТЛ обозначает транзисторно-транзисторная логика. Логические элементы схем этого семейства строятся исключительно на биполярных транзисторах. Только для сдвига уровня и отвода напряжения применяются диоды. Сопротивления служат в качестве делителей напряжения и ограничителей тока.

> TTL-элементы производятся исключительно в виде интегральных микросхем.

Новым элементом является так называемый мультиэмиттерный транзистор. Структура такого транзистора показана на рис. 6.39. С общей базой граничат три эмиттерных зоны. Таким образом, образуются три разделенных PN-перехода между базой и эмиттерами. Эти PN-переходы можно рассматривать как диоды.

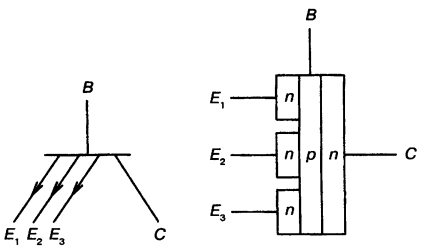


Рис. 6.39. Мультиэмиттерный транзистор.

К базе приложено напряжение чуть выше 0,7 В относительно земли. При замыкании эмиттера на землю начинает течь базовый ток. Величина базового тока определяется значением базового сопротивления R_1 и напряжения питания U_S (рис. 6.40).

> Величина базового тока выбирается такой, чтобы мультиэмиттерный транзистор надежно управлялся в режиме насыщения.

Режим насыщения является высоковероятностным режимом (см. Электроника 3, Схемы на транзисторах). Напряжение коллектор-эмиттер U_{CE} падает при этом до значения насыщения U_{CEsat}. Оно равно примерно 0,2 В.

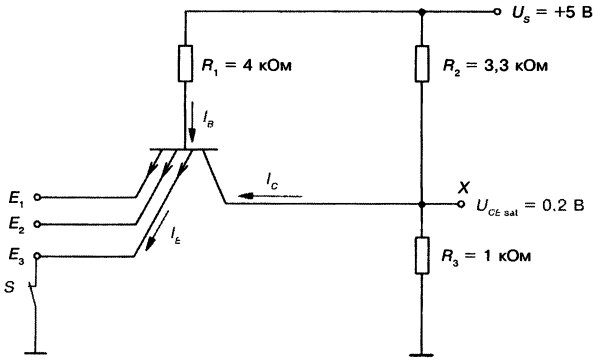


Рис. 6.40. Схема на мультиэмиттерном транзисторе.

Сопротивление R_2 на рис. 6.40 должно быть равно примерно 3,3 кОм. Ток I_C будет тогда небольшим. Ток эмиттера I_E равен приблизительно току I_B. Обычно токи эмиттеров выбираются между 1 мА и 1,6 мА.

Если два или все три эмиттера в схеме на рис. 6.40 заземлены, то напряжение в точке X практически не меняется. X остается на уровне примерно 0,2 В. Что изменится, если эмиттер E_3 заземлить, а на эмиттеры E_1 и E_2 подать напряжение питания +5 В? PN-переходы между эмиттерами E_1, E_2 и базой будут закрыты (плюс на n-зоне). PN-переход E_3 к базе остается открытым. Транзистор продолжает оставаться в состоянии насыщения. Напряжение в точке X не меняется.

Если уровню L поставить в соответствие напряжение от 0 В до 0,4 В, то при приложении хотя бы к одному из эмиттеров схемы на рис. 6.40 уровня L на выходе X будет также действовать низкий уровень L.

Совершенно иначе работает схема (рис. 6.41), если на все эмиттеры подать напряжение питания (ВЫСОКИЙ уровень H). К эмиттерным зонам приложен потенциал +5 В. В точке X благодаря делителю напряжения R_2/R_3 будет напряжение примерно 0,45 В, то есть напряжение, которое соответствует уровню L.

К коллектору приложено напряжение примерно 0,45 В, к эмиттеру — +5 В. Теперь мультиэмиттерный транзистор работает в инверсном режиме, т. е. эмиттер и коллектор обменялись функциями. Эмиттеры работают как коллекторы, коллектор работает как эмиттер.

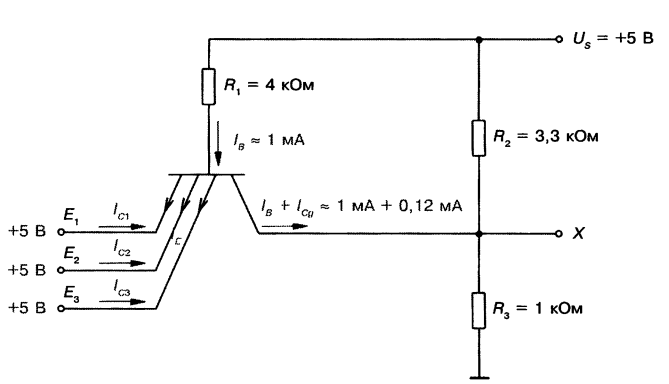


Рис. 6.41. Схема на мультиэмиттерном транзисторе в инверсном режиме.

Если ко всем эмиттерам приложен высокий *H*-уровень, то мультиэмиттерный транзистор работает в инверсном режиме.

Базовый ток течет от источника питания через R_1 и R_3 на землю (см. рис. 6.41). Типичное значение базового тока равно 1 мА. Однако токи коллектора I_{C1}, I_{C2} и I_{C3} от эмиттерных входов E_1, E_2 и E_3 относительно малы. Каждый ток коллектора не превышает 40 мкА. Благодаря специальной технологии добились, что так называемое инверсное усиление тока мультиэмиттерного транзистора очень мало. Следовательно, управляемые элементы потребляют относительно небольшой ток.

Инверсное усиление по току мультиэмиттерного транзистора много меньше единицы.

На выходе *X* представленной на рис. 6.41 схемы напряжение поднимется с 1 до примерно 2 В. Этого напряжения недостаточно для достижения высокого уровня *H*. Оно слишком низкое. Однако этим напряжением можно управлять последующим транзистором, как показано на рис. 6.42. Сопротивления делителя напряжения R_2 и R_3 можно тогда не использовать. Функцию резистора R_3 берет на себя участок база-эмиттер T_2. Высокоомное сопротивление R_2 заменяется R_C и участком база-коллектор транзистора T_2. Схема имеет входы *A*, *B*, *C* и выход *Z*.

Если на входах *A*, *B* и *C* действует уровень *H*, то T_1 работает в инверсном режиме. Транзистор T_2 войдет в режим насыщения. На выходе *Z* будет напряжение примерно 0,2 В. Такое напряжение относится к уровню *L*.

Если по меньшей мере на одном из входов действует уровень *L*, то мультиэмиттерный транзистор T_1 работает нормально в режиме насыщения. Напряжение на его коллекторе падает примерно на 0,2 В. T_2 запирается. На выходе *Z* действует уровень *H*. Рабочая таблица схемы рис. 6.42 представлена на рис. 6.43. При положительной логике получается логический элемент И-НЕ.

Как будет работать схема при открытом входе? Под открытым входом понимают вход, который не подключен ни к уровню *L*, ни к уровню *H*. То есть ко входу ничего не подключено, и он висит в воздухе. Такой вход не в состоянии понизить напряжение в точке *X* схемы на рис. 6.42 до 0,2 В. То есть он не может заставить мультиэмиттерный транзистор проводить ток.

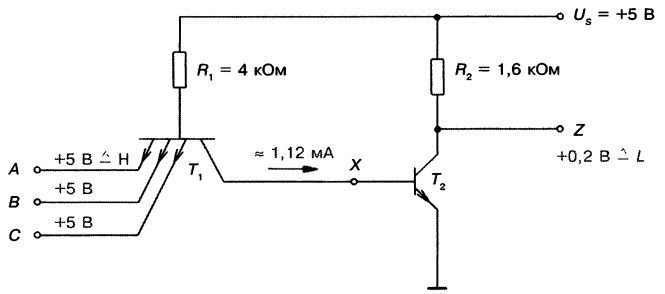


Рис. 6.42. Простой ТТЛ-элемент (И-НЕ при положительной логике).

Вар.	C	B	A	Z
1	L	L	L	H
2	L	L	H	H
3	L	H	L	H
4	L	H	H	H
5	H	L	L	H
6	H	L	H	H
7	H	H	L	H
8	H	H	H	L

Рис. 6.43. Рабочая таблица к схеме на рис. 6.42.

Если на двух входах схемы рис. 6.42 действует уровень *H*, а третий вход открыт, то мультиэмиттерный транзистор будет работать в инверсном режиме — так же, как если бы на все три входа действовал *H*-уровень.

> Для ТТЛ-схем открытый вход приравнивается ко входу с высоким *H*-уровнем.

Посмотрим внимательнее на схему (рис. 6.42). Заметно, что мультиэмиттерный транзистор никогда не заперт. Он всегда проводит — либо в прямом, либо в инверсном режиме. Базовый ток есть всегда. В прямом режиме он течет к входам, которые лежат на уровне *L*. В инверсном он течет к базе транзистора T_2. Значит, носители зарядов базовой зоны при переключении режимов не должны рассасываться, и переключение из одного состояния в другое происходит очень быстро.

> Мультиэмиттерный транзистор очень быстро переключается из прямого режима в инверсный и наоборот, так как носителям зарядов в базе при переключении не нужно рассасываться.

Время переключения для транзистора T_2 получается также малым. Носители заряда базы T_2 как раз успевают рассосаться за время перехода мультиэмиттерного транзистора T_1 из состояния насыщения в запертое состояние.

Представленная на рис. 6.42 ТТЛ-схема не очень хорошо подходит для управления последовательно подключенными ТТЛ-элементами. Выход *Z* в состоянии *L* должен принимать от каждого присоединенного входа ток примерно 1,6 мА (рис. 6.44). При десяти присоединенных входах (коэффициент разветвления 10) это уже 16 мА. Эти 16 мА могут попасть на землю через проводящий транзистор T_2.

Более сложный случай, если выход *Z* имеет *H*-состояние и должен управлять десятью присоединенными элементами. Вытекающий из выхода *Z* ток вызывает падение напряжения на R_C. На значение этого напряжения уменьшается выходной уровень *Z*. Это неудобно. Уменьшение выходного уровня можно предотвратить, используя схему на дополнительном транзисторе, так называемый двухтактный выходной каскад. Такой каскад представлен на рис. 6.45.

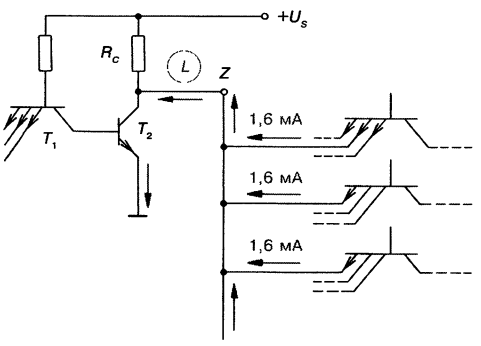


Рис. 6.44. Управление ТТЛ-элементами при состоянии выхода *L*.

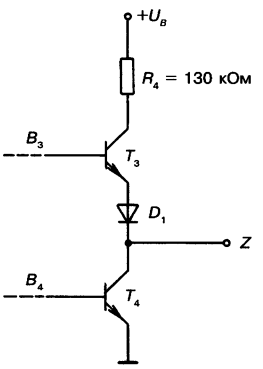


Рис. 6.45. Двухтактный выходной каскад.

Один из транзисторов T_3 и T_4 должен быть заперт, другой проводить. Если заперт T_3 и проводит T_4, то на выходе Z действует уровень L. Поступающий на выход Z ток течет через T_4 на землю.

Если заперт T_4 и проводит T_3, то на выходе Z действует уровень H. Ток управления, необходимый для следующих элементов, течет от $+U_B$ через R_4, T_3, диод D_1 — к выходу Z. Если выход Z нагружается сильнее, т. е. через выход Z течет сильный ток, то выходной уровень падает только на значение падения напряжения на резисторе R_4. На T_3 и D_1 при возрастании тока не возникает дополнительного падения напряжения.

Двухтактная схема может выдавать и принимать относительно большие токи.

При переключении выходных состояний оба транзистора T_3 и T_4 могут кратковременно проводить. Сопротивление R_4 в этом случае ограничивает ток.

Диод D_1 служит для создания нужного уровня напряжения. Как он понижает напряжение, наглядно показано на схеме на рис. 6.46.

Если транзистор T_2 проводит, то к точке B_4 приложено напряжение примерно 0,7 В (напряжение база-эмиттер T_4). Транзистор T_4 будет полностью открыт. К точке Z приложено напряжение примерно +0,2 В. К T_2 тоже приложено напряжение коллектор-эмиттер 0,2 В, так что в точке B_3 потенциал относительно земли получается 0,9 В. Без диода D_1 на транзистор T_3 подавалось бы напряжение базис-эмиттер $U_{BE} = 0{,}7$ В (эмиттер +0,2 В, база +0,9 В). Транзистор T_3 также открылся бы.

Так как на диоде D_1 падение напряжения составляет примерно 0,7 В, то потенциал эмиттера T_3 поднимается до уровня примерно 0,9 В. При этом U_{BE} транзистора T_3 будет равно примерно 0 В, и T_3 надежно закрывается.

При переключении от ТТЛ-элементов на выходе получаются довольно крутые импульсы напряжения. Переходное время сигнала (см. разд. 6.4.3) достаточно мало и составляет примерно 5 нс. Вследствие этого могут возникнуть так называемые «колебания напряжения». Переходный процесс при переключении входа ТТЛ-элемента с H на L изображен на рис. 6.47. Напряжение на входе может доходить кратковременно до −2 В. Диоды D_2, D_3 и D_4 в схеме на рис. 6.46 должны подавлять колебания напряжения и принимать на себя отрицательное напряжение. Они называются **отводными диодами**.

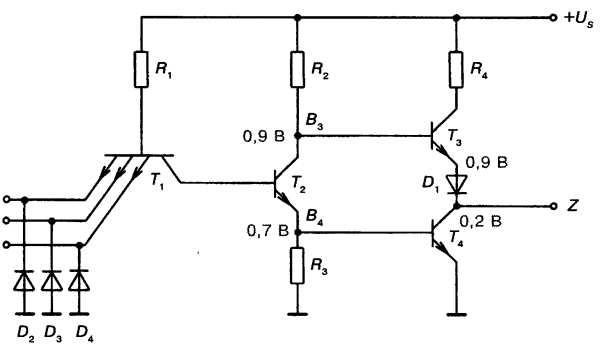


Рис. 6.46. Типовой ТТЛ-элемент с двухтактным выходным каскадом.

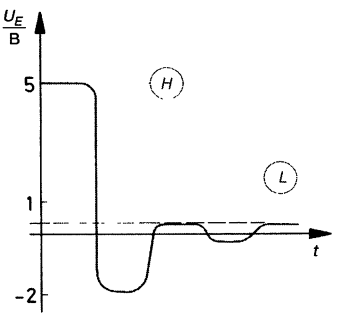


Рис. 6.47. Колебания входного напряжения при переходе с H на L.

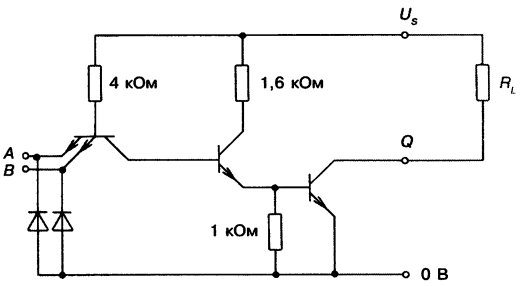


Рис. 6.48. ТТЛ-элемент И-НЕ с открытым коллектором.

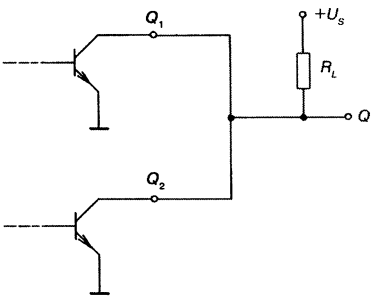


Рис. 6.49. Совместное подключение ТТЛ-элементов с открытым коллектором.

В ТТЛ-схемах имеются элементы с так называемым «открытым коллектором». В этих элементах отсутствует сопротивление коллектора. Вывод коллектора подключен к выводу на корпусе элемента (рис. 6.48).

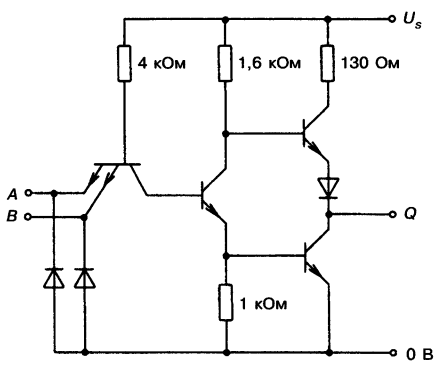
При синтезе схем следует правильно подбирать сопротивление коллектора. Элементы с открытым коллектором предназначены для проводных операций.

Можно соединить открытые коллекторы нескольких элементов вместе и затем общую точку соединить с источником питания (рис. 6.49). Величина совместного сопротивления выбирается согласно рекомендациям производителя. При этом большое значение имеет число совместно подключенных элементов.

Представленная на рис. 6.49 схема является схемой проводного И при положительной логике. Если выход имеет уровень *L*, т. е. выходной транзистор открыт, то общая точка *Q* будет всегда на уровне *L*. Только если все выходные транзисторы заперты, или все выходы имеют *H*-уровень, общая точка имеет *H*-уровень.

ТТЛ-семейство делится на ряд более мелких подсемейств. Схемы отдельных подсемейств различаются между собой прежде всего потреблением мощности и быстродействием, а также помехоустойчивостью.

6.6.2. Стандартные ТТЛ-схемы



Подсемейство «стандартные ТТЛ» играет большую роль в схемотехнике. Типичная схема из стандартных ТТЛ показана на рис. 6.50. Элемент производит при положительной логике логическую операцию И-НЕ.

Рис. 6.50. Стандартный ТТЛ-элемент схемы FLH 101-7400 (И-НЕ при положительной логике).

6.6.2.1. Схемы

Почти все ТТЛ-элементы выпускаются в виде интегральных микросхем. Схема FLH 101-7400 содержит, например, четыре элемента И-НЕ, что следует из схемы на рис. 6.51.

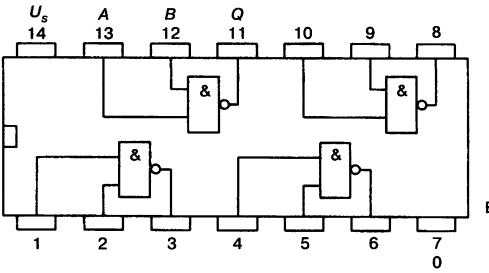


Рис. 6.51. Схема подключения интегральной микросхемы FLH 101-7400.

В основном ТТЛ-элементы выпускаются в DIP-корпусах (рис. 6.52).

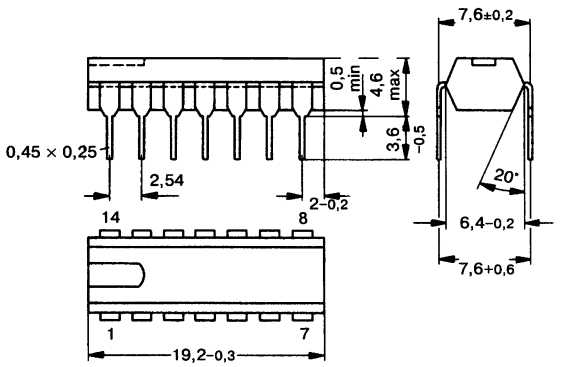
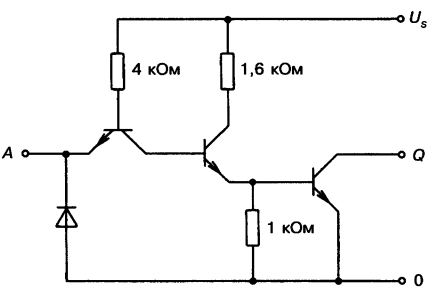


Рис. 6.52. Корпус DIP (с двухсторонним расположением выводов).

Для проводных И используются элементы с открытым коллектором. Схема элемента НЕ с открытым коллектором показана на рис. 6.53. Интегральная схема FLH 271-7405 содержит шесть таких элементов НЕ (рис. 6.54).

Рис. 6.53. Схема элемента НЕ с открытым коллектором (Siemens).



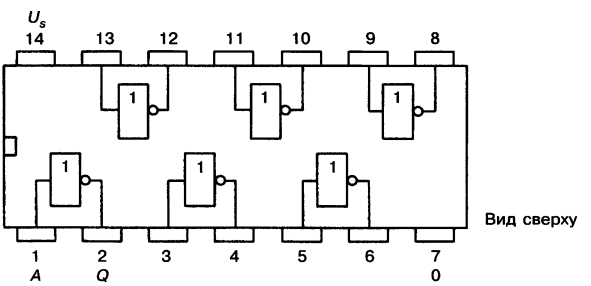


Рис. 6.54. Схема подключения интегральной микросхемы FLH 271-7405 (Siemens).

Элементы И-НЕ от двух до восьми входов имеются в различных исполнениях: с двухтактной схемой, с открытым коллектором или с силовым выходом. Схема И-НЕ-элемента с пятью входам представлена на рис. 6.55. Микросхема FLH 331-4931 содержит два таких элемента.

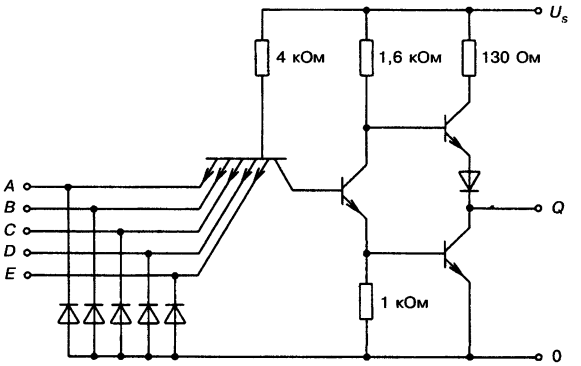


Рис. 6.55. ТТЛ-элемент И-НЕ с пятью входами и схема соединений схемы FLH 331-4931.

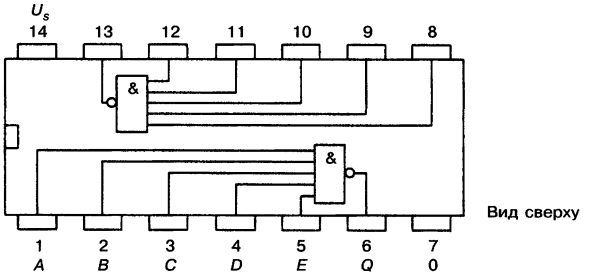
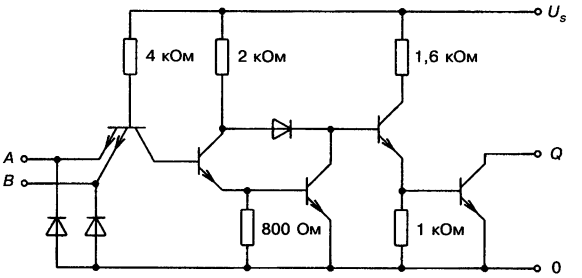
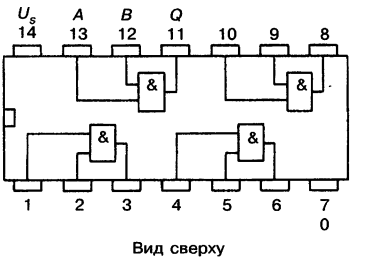


Рис. 6.56. ТТЛ-элемент И с открытым коллектором и схема соединений схемы FLH 391-7409.





Элементы И применяются реже. Их можно легко сделать из элементов И-НЕ структурирования. Для простых схем используется схема FLH 391-7409. Она состоит из четырех элементов И с открытым коллектором (рис. 6.56) и предназначена для проводных И-соединений.

6.6.2.2. Предельные значения и параметры схем

Предельными называются величины, которые ни в коем случае не должны быть превышены. Если это произойдет, то элемент будет выведен из строя. Для стандартных ТТЛ-схем действуют следующие общие предельные значения:

	Нижний предел	Верхний предел
Напряжение питания U_S	–0,5 В	7,0 В
Входное напряжение U_I	–1,5 В	5,5 В
Разность напряжения между входами U_D		5,5 В
Выходное напряжение U_Q	–0,8 В	5,5 В
Рабочая температура T_U Зона 1 Зона 5	 0 °C –25 °C	 70 °C 85 °C
Температура хранения T_S	–65 °C	150 °C

К основным параметрам относятся статические характеристики, быстродействие и логические данные. Одним из главных параметров является напряжение питания. Оно может варьироваться между 4,75 В и 5,25 В. Типичным значением является 5 В.

Для всех элементов указывается нижняя граница входного напряжения *H*-уровня. Она составляет обычно 2 В. При самом малом напряжении уровня *H* U_{IH}, равном 2 В, выходное напряжение U_{QL} не может превышать 0,4 В, даже когда выходной ток достигает максимального значения 16 мА. Для проверки соблюдения этого условия существует тестовая схема 1 на рис. 6.57. Условия испытаний приведены в таблице на рис. 6.58.

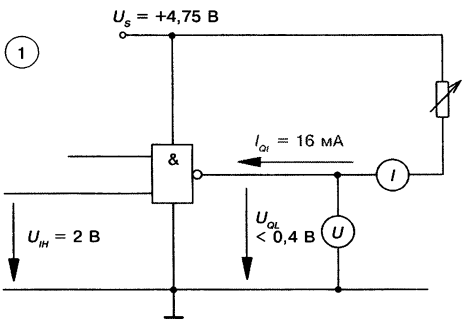


Рис. 6.57. Тестовая схема 1.

Статические параметры		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Входное напряжение высокого уровня *H*	U_{IH}	U_S = 4,75 В	1	2,0			В
Выходное напряжение низкого уровня *L*	U_{QL}	U_S = 4,75 В U_{IH} = 2 В I_{QL} = 16 мА	1		0,22	0,4	В
Входное напряжение низкого уровня *L*	U_{IL}	U_S = 4,75 В	2			0,8	В
Выходное напряжение высокого уровня *H*	U_{QH}	U_S = 4,75 В U_{IL} = 0,8 В $-I_{QH}$ = 16 мкА	2	2,4	3,3		В

Рис. 6.58. Выдержка из протокола испытаний.

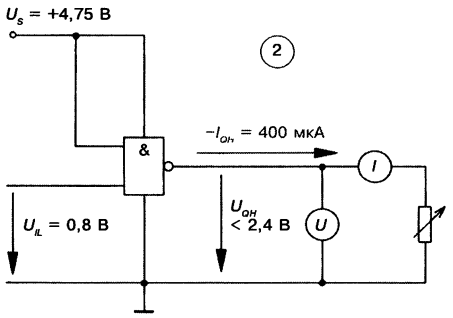


Рис. 6.59. Тестовая схема 2.

При самом высоком напряжении уровня L U_{IL}, равном 0,8 В, выходное напряжение U_{QH} не может быть ниже 2,4 В. Значения для верхней границы U_{IL} и нижней границы U_{QH} определяются с помощью тестовой схемы 2 на рис. 6.59. Все остальные нетестируемые входы включаются на H-уровень, так как это соответствует самому неблагоприятному варианту работы схемы.

Входные токи различаются в зависимости от уровня на входе. H-входной ток I_{IH} является током входа, который протекает при H-уровне 2,4 В. Он может быть равен максимально 40 мкА. Он измеряется при условиях таблицы на рис. 6.60 по тестовой схеме 3 (рис. 6.61). При максимально допустимом напряжении входа U_I, равном 5,5 В, входной ток может быть равен максимально 1 мА.

Для определения L-входного тока применяют тестовую схему 4. При U_{IL} равном 0,4 В входной ток может быть равен максимально 1,6 мА (рис. 6.62).

Статические параметры		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Входной ток высокого уровня на каждом входе *H*	I_{IH}	U_{IH} = 2,4 В	3			40	мкА
	I_I	U_I = 5,5 В U_S = 5,25 В	3			1	мА
Входной ток низкого уровня *L*	$-I_L$	U_S = 5,25 В U_{IL} = 0,4 В	4			1,6	мА
Ток короткого замыкания на каждом выходе	$-I_Q$	U_S = 5,25 В	5	18		55	мА
Ток питания *H*	I_{SH}	U_S = 5,25 В U_I = 0 В	6		4	8	мА
Ток питания *L*	I_{SL}	U_S = 5,25 В U_I = 5 В	6		12	22	мА

Рис. 6.60. Выдержка из протокола испытаний.

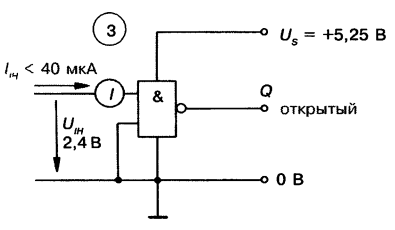


Рис. 6.61. Тестовая схема 3.

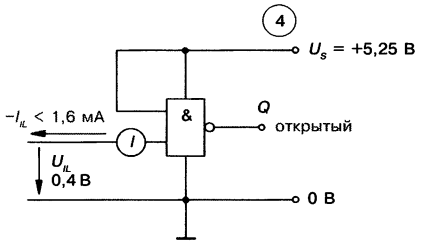


Рис. 6.62. Тестовая схема 4.

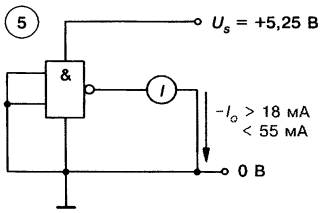


Рис. 6.63. Тестовая схема 5.

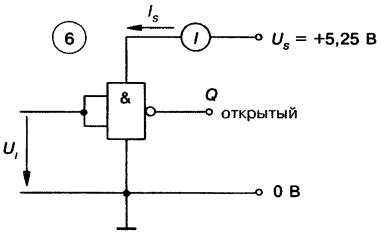


Рис. 6.64. Тестовая схема 6.

Следующим важным параметром является ток короткого замыкания. Он определяется с помощью тестовой схемы 5. Его нижний предел равен 18 мА, верхний равен 55 мА (рис. 6.63). Все входы должны иметь *L*-уровень. По возможности следует избегать короткозамкнутых выходов. Для некоторых элементов короткое замыкание выходов приводит к поломке.

Потребляемый логическим элементом ток определяется при помощи схемы 6 (рис. 6.64). Он имеет различное значение в зависимости от уровня на входе. Максимально допустимое напряжение питания элемента — 5,25 В. В данных на микросхемы всегда указывается общее напряжение питания микросхемы, а не каждого элемента в отдельности.

Следующей характеристикой является статическая помехоустойчивость. Помехоустойчивость уже была подробнее рассмотрена в разд. 6.4.5. Для стандартных ТТЛ-элементов статическая помехоустойчивость составляет обычно 1 В, при неблагоприятных условиях — минимум 0,4 В.

Быстродействие уже обсуждалось в разд. 6.4.3. Оно указано в полной таблице данных на рис. 6.65.

К логическим параметрам относятся коэффициент разветвления по выходу F_Q и коэффициент объединения по входу F_I. Они подробнее разобраны в разделе 6.4.4. Также к логическим параметрам относится логическая функция. Под логической функцией понимают уравнение логической операции, которую выполняет элемент при положительной логике.

На рис. 6.66 приведена полная таблица данных интегральной микросхемы FLH 201-7401. Эта схема содержит четыре элемента И-НЕ с открытым коллектором. В таблице приведены уравнения для вычисления сопротивления коллектора и таблица сопротивлений.

6.6.2.3. Характеристики

Для TTL-элементов существуют несколько характеристик, которые дают представление о работе логического элемента. Особенно важна передаточная характеристика. Типичные передаточные характеристики для стандартных ТТЛ-элементов представлены на рис. 6.67.

> Передаточная характеристика отражает зависимость выходного напряжения от входного.

Из передаточной характеристики могут быть определены зона входных напряжений высокого уровня, зона входных напряжений низкого уровня, зона выходных напряжений высокого уровня и зона выходных напряжений

Четыре двухвходовых элемента И-НЕ

Статические параметры в зоне температур 1 и 5		Условия испытаний	Схема для испытания	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S			4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В	1	2,0			В
Входное напряжение *L*-уровня	U_{IL}	U_S = 4,75 В	2			0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В $-I_I$ = 12 мА				1,5	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В U_{IL} = 0,8 В $-I_{QH}$ = 400 мкА	2	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В U_{IH} = 2 В I_{QL} = 16 мА	1		0,2	0,4	В
Статическая помехоустойчивость	U_{SS}			0,4	1		В
Входной ток высокого уровня *H* на каждом входе	I_{IH}	U_{IH} = 2,4 В U_S = 5,25 В U_I = 5,5 В	3			40	мкА
	I_I		3			1	мА
Входной ток низкого уровня *L* на каждом входе	$-I_{IL}$	U_S = 5,25 В U_{IL} = 0,4 В	4			1,6	мА
Ток короткого замыкания на каждом выходе	$-I_Q$	U_S = 5,25 В	5	18		55	мА
Ток питания *H*	I_{SH}	U_S = 5,25 В U_I = 0 В	6		4	8	
Ток питания *L*	I_{SL}	U_S = 5,25 В U_I = 5 В	6		12	22	мА
Время переключения при U_S = 5 В, T_U = 25 °C							
Быстродействие	t_{PHL}	C_L = 15 пФ R_L = 400 Ом	22		7	15	нс
	t_{PLH}				11	22	нс
Логические параметры							
Коэффициент разветвления по выходу	F_Q					10	
Коэффициент объединения по входу	F_I					1	
Логическая функция		$Q = \overline{A \wedge B}$					

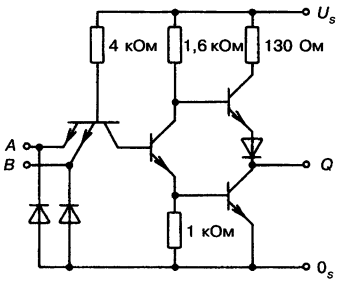
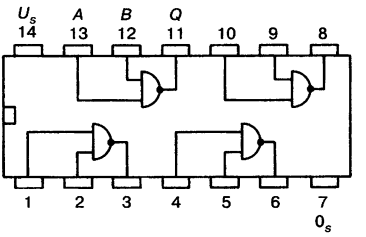


Схема включения (один элемент)



Цоколевка, вид сверху

Рис. 6.65. Полная таблица данных интегральной микросхемы FLH 101-7400 (по данным Siemens).

FLH 201-7401
FLH 201 S-7401 S1
FLH 201 T-7401 S3

FLH 205-8401
FLH 205 S-8401 S1
FLH 205 T-8401 S3

Четыре двухвходовых элемента И-НЕ с открытым коллектором

Элементы FLH 201/205 предназначены для соединений типа проводного И

Статические параметры в зоне температур 1 и 5		Условия испытаний	Схема для испытания	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S			4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В	1	2,0			В
Входное напряжение *L*-уровня	U_{IL}	U_S = 4,75 В	14			0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В $-I_I$ = 12 мА				1,5	В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В U_{IH} = 2 В I_{QH} = 16 мА	1		0,2	0,4	В
Статическая помехоустойчивость	U_{SS}			0,4	1,0		В
	I_{IH}		3			40	мкА
Входной ток высокого уровня *H* на каждом входе	I_I	U_{IH} = 2,4 В U_I = 5,5 В U_S = 5,25 В	3			1	мА
Входной ток низкого уровня *L* на каждом входе	$-I_{IL}$	U_S = 5,25 В U_{IL} = 0,4 В	4			1,6	мА
Выходное напряжение *H*-уровня на каждом входе	I_{QH}	U_S = 4,75 В U_{QH} = 5,5 В U_{IL} = 0,8 мА	14			250	мкА
Ток питания *H*	I_{SH}	U_S = 5,25 В U_I = 0 В	6		4	8	мА
Ток питания *L*	I_{SL}	U_S = 5,25 В U_I = 5 В	6		12	22	мА

Статические параметры в зоне температур 1 и 5		Условия испытаний	Схема для испытания	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Время переключения при U_S = 5 В, T_U = 25 °C							
Быстродействие	t_{PHL}	R_L = 400 Ом R_L = 4 кОм C_L = 15 пФ	22		8	15	нс
	t_{PLH}				35	45	нс
Логические параметры							
Коэффициент разветвления по выходу	F_{QL}					10	
Коэффициент объединения по входу	F_I					1	
Логическая функция		$Q = \overline{A \wedge B}$					

FLH 201 S, FLH 205 S: как FLH 201/205 только выход 15 В/250 мкА
FLH 201 T, FLH 205 T: как FLH 201/205 только выход 5,5 В/50 мкА

Расчет сопротивления коллектора R_L

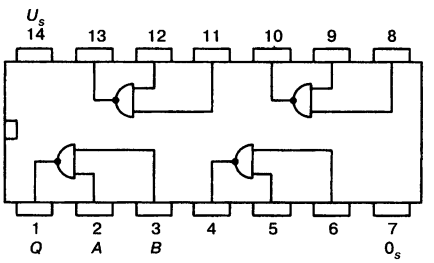
Расчет проводится по следующим формулам:

H-уровень *L*-уровень

$$R_{LA} = \frac{U_S - 2{,}4\ \text{В}}{n250\ \text{мкА} + N40\ \text{мкА}}\ (\text{МОм});\quad R_{LB} = \frac{U_S - 0{,}4\ \text{В}}{16\ \text{мА} - N1{,}6\ \text{мА}}\ (\text{кОм}),$$

где: U_S — напряжение питания; n — количество FLH 201 в логическом сложении И (Численные значения; см. таблицу); N — количество присоединенных входов.

При U_S = 5 В и различных n и N получаются нижеследующие предельные значения R_L. Номинал применяемого в схеме сопротивления должен находиться между этими предельными значениями.



Цоколевка корпуса, вид сверху

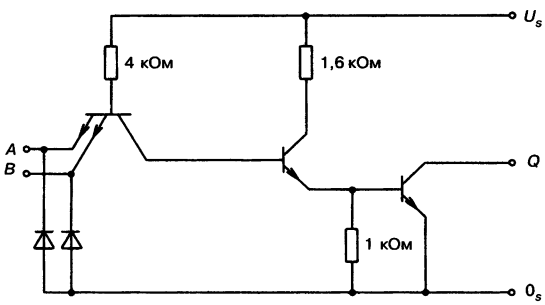


Схема включения (один элемент)

Рис. 6.66. Полная таблица данных интегральной микросхемы FLH 201-7401 (по данным Siemens).

	1	2	3	4	5	6	7	1 до 7
	Верхнее предельное значение R_{LA}, Ом							
1	8965	4814	3291	2500	2015	1688	1452	319
2	7878	4482	3132	2407	1954	1645	1420	359
3	7027	4193	2988	2321	1897	1604	1390	410
4	6341	3939	2857	2241	1843	1566	1361	479
5	5777	3714	2736	2166	1793	1529	1333	575
6	5306	3513	2626	2096	1744	1494	1306	718
7	4905	3333	2524	2031	1699	1460	1280	958
8	4561	3170	2419	1969	1656			1437
9	4262	3023	Зона недопустимых значений					2875
10	4000							4000

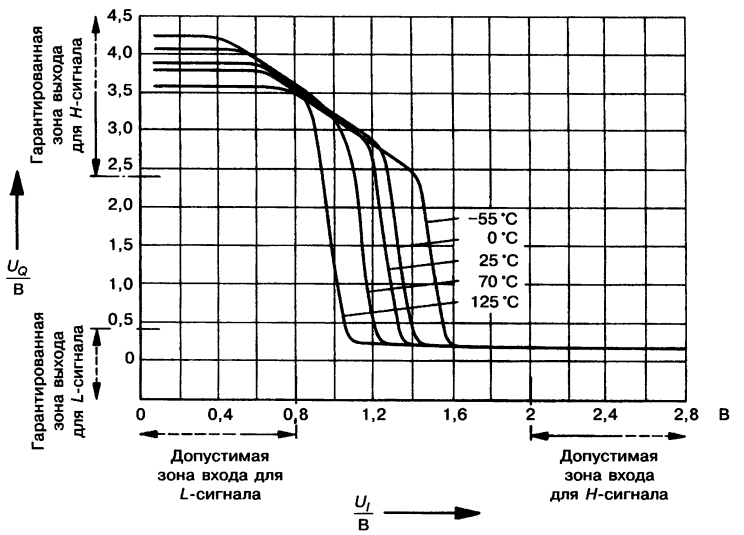


Рис. 6.67. Передаточные характеристики (при $U_S = 5$ В, $F_Q = 10$, Siemens).

низкого уровня. Для различных рабочих температур получаются различные передаточные характеристики.

Следующей важной характеристикой является входная характеристика (рис. 6.68).

Входная характеристика отражает связь между входным током и входным напряжением.

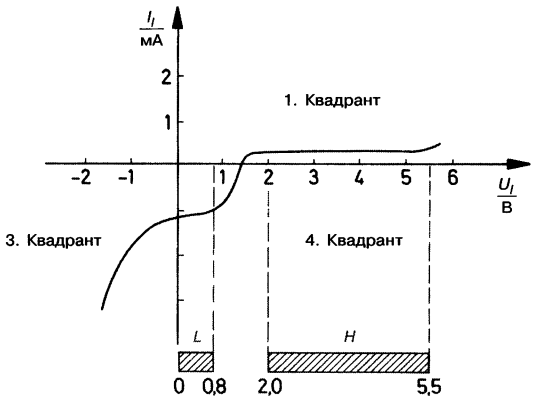


Рис. 6.68. Входная характеристика (при U_S = 5 В, Siemens) при 25 °С.

Если напряжение входа U_I примерно 1,5 В, то оно может быть рассмотрено как напряжение высокого H-уровня. Входной ток равен примерно 40 мкА. Общепринято, что входное напряжение можно рассматривать как напряжение высокого H-уровня только начиная с 2 В.

Если напряжение менее 1,4 В, ток вытекает из входа. Его значение в области входных напряжений L-уровня (0—0,8 В) достигает примерно 1 мА. Максимально допустимое значение тока 1,6 мА. Входное напряжение также может быть немного отрицательным, до −1,5 В. Значение −1,5 В является нижним пределом для U_I. Входная характеристика на рис. 6.68 приведена для температуры окружающей среды 25 °С. При других температурах характеристика немного сдвигается.

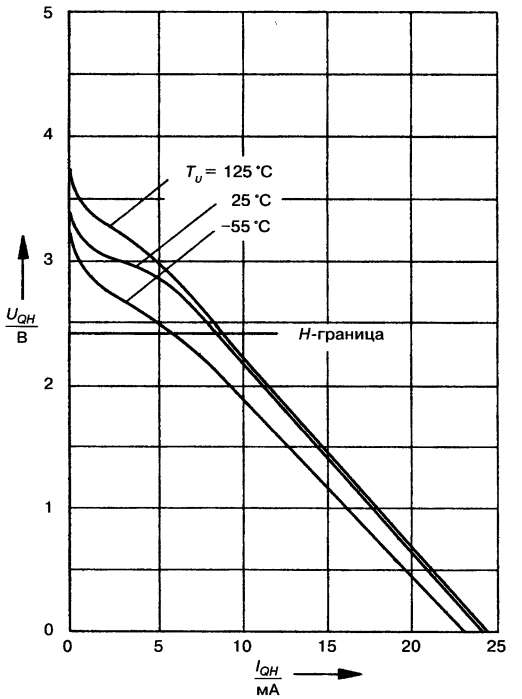


Рис. 6.69. Выходная характеристика для H-уровня при U_S = 5 В и U_I = 0,4 В (Siemens).

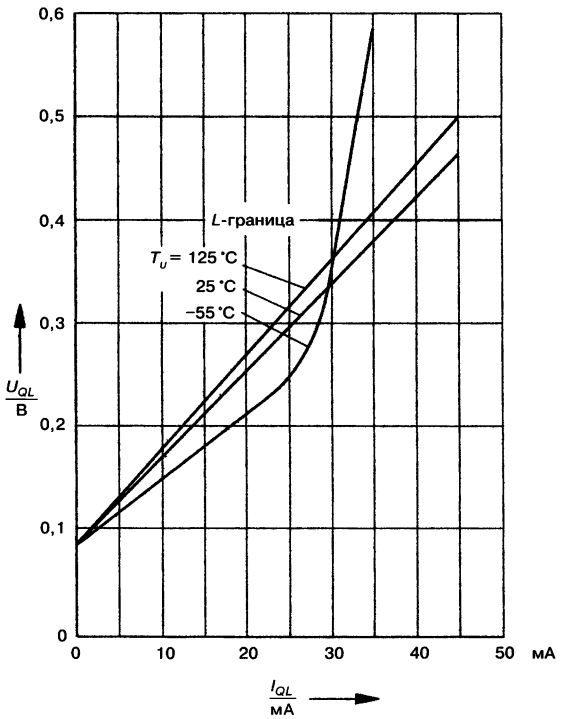


Рис. 6.70. Выходная характеристика для L-уровня при $U_S = 5$ В, $U_I = 2{,}4$ В.

Зависимость между выходным током и выходным напряжением представляется двумя видами характеристик. Один вид — для состояния H на выходе, другой — для состояния L на выходе. Рис. 6.69 показывает выходные характеристики для H-уровня при различных температурах. Напряжение U_{QH} не должно превышать границы высокого уровня (2,4 В). При температуре окружающей среды 25 °C с выхода можно потреблять максимально 8 мА.

На рис. 6.70 изображены выходные характеристики для низкого уровня L. Ток I_{QL} течет внутрь входа. При температуре окружающей среды 25 °C максимально допустимый втекающий ток равен 34 мА. Если ток превысит это значение, то U_{QL} выйдет за пределы области выходного напряжения L-уровня (верхняя граница 0,4 В).

6.6.2.4. Энергопотребление

Стандартные интегральные ТТЛ-микросхемы отличаются сравнительно высоким энергопотреблением. Например, схема FLH 101-7400 потребляет при напряжении питания 5,25 В ток порядка 8 мА, что соответствует потребляемой мощности 42 мВт. Схема содержит четыре элемента И-НЕ. Значит, каждый вентиль И-НЕ потребляет примерно 10 мВт. В общем, это не очень большая величина. Однако микросхема с 10000 логическими элементами будет потреблять уже 100 Вт. Для питания такой микросхемы уже не получится использовать батарейки.

Поэтому стандартные интегральные ТТЛ-микросхемы работают преимущественно от стационарных стабилизированных источников питания.

6.6.3. ТТЛ с пониженным энергопотреблением (Low-Power-TTL, LTTL)

Low Power с английского переводится как пониженное энергопотребление. ТТЛ-элементы с пониженным энергопотреблением потребляют 1/10 мощности, потребляемой стандартными ТТЛ-элементами. Уменьшения энергопотребления можно добиться, увеличивая сопротивления внутри микросхемы. Типичный элемент ТТЛ с пониженным энергопотреблением изображен на рис. 6.71. Видно, что структуры стандартной ТТЛ-схемы и ТТЛ с пониженным энергопотреблением практически идентичны. Однако различие становится ясным при внимательном рассмотрении сопротивлений. Номиналы сопротивлений стандартных ТТЛ указаны серым шрифтом в скобках. Номиналы сопротивлений ТТЛ с пониженным энергопотреблением в десять-двенадцать раз больше.

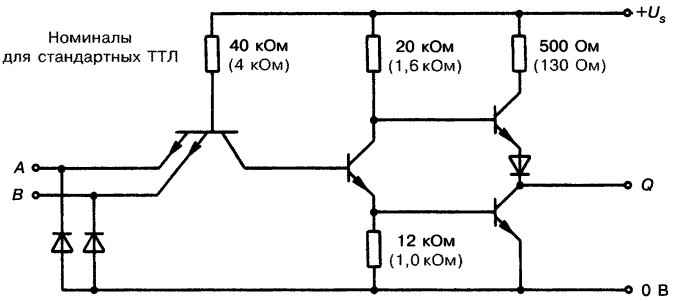


Рис. 6.71. Типовой ТТЛ-элемент с пониженным энергопотреблением (И-НЕ при положительной логике).

Один логический элемент И-НЕ схемы ТТЛ с пониженным энергопотреблением потребляет мощность порядка 1 мВт. Быстродействие элементов в основном определяется временами заряда-разряда емкостей транзисторов. При увеличении сопротивлений время заряда-разряда возрастает и, следовательно, быстродействие ТТЛ с пониженным энергопотреблением ниже, чем у стандартных ТТЛ.

> ТТЛ с пониженным энергопотреблением потребляют 1/10 мощности, потребляемой стандартными ТТЛ. Зато быстродействие ТТЛ с пониженным энергопотреблением примерно в три раза ниже, чем у стандартных ТТЛ.

Среднее время задержки t_p, определяющее время выполнения одной логической операции, составляет для ТТЛ с пониженным энергопотреблением примерно 33 нс.

6.6.4. Высокоскоростные ТТЛ (High-Speed-TTL, HTTL)

Высокоскоростные ТТЛ характеризуются прежде всего высоким быстродействием. Внутренняя структура этого подсемейства логических элементов, как и в случае ТТЛ с пониженным энергопотреблением, не отличается от стандартных ТТЛ. Сопротивления этого подсемейства уменьшены (рис. 6.72). Благодаря этому процессы заряда-разряда емкостей транзисторов протека-

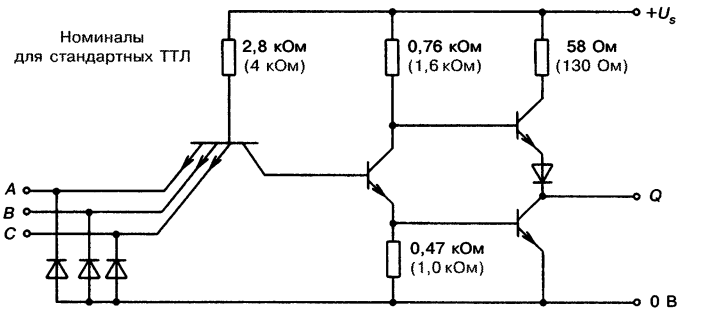


Рис. 6.72. Типовой высокоскоростной ТТЛ-элемент (И-НЕ при положительной логике).

ют быстрее, и быстродействие существенно возрастает. Среднее время задержки t_p составляет примерно 5 нс.

За быстродействие приходится платить повышенным энергопотреблением. Оно более чем в два раза выше, чем для стандартных ТТЛ. Один вентиль И-НЕ, изображенный на рис. 6.72, потребляет примерно 23 мВт.

> Быстродействие высокоскоростных ТТЛ-элементов в два раза выше, чем у стандартных ТТЛ. Однако они потребляют энергии более чем в два раза больше.

Компьютер, построенный на высокоскоростных ТТЛ, работает в два раза быстрее, чем компьютер на стандартных ТТЛ. Он выполнит за то же время двойную работу. Это очень хорошо. Хуже, что для этого ему потребуется больше энергии.

6.6.5. Шотки-ТТЛ (ТТЛШ)

При создании одновременно быстрых и экономичных микросхем используется свойство транзисторов быстро переключаться в ненасыщенном состоянии (см. Бойт/Шмуш, Электроника, ч. 3). При подключении диода по рис. 6.73 предотвращается насыщение транзистора. Диод должен иметь высокое быстродействие. Поэтому применяют диоды Шотки (см. Бойт, Электроника, ч. 2). Диоды Шотки отличаются высоким быстродействием и пороговым напряжением 0,35 В.

Транзистор на рис. 6.73 может проводить только до тех пор, пока U_{CE} не упадет до 0,4 В. Затем диод Шотки предотвращает дальнейшее нахождение транзистора в открытом состоянии. Он сам проводит в прямом направлении. От базы ток течет через диод и переход коллектор-эмиттер к земле. Этот ток уже не течет через базу транзистора и не служит для его управления.

Начало насыщения транзистора начинается тогда, когда U_{CE} падает до значения U_{BE}, т. е. при $U_{CE} = 0{,}4$ В транзистор уже находится в насыщении, но не в глубоком.

Диод Шотки на рис. 6.73 называется «антинасыщающий диод». Транзистор с диодом Шотки называется транзистором Шотки. Его условное обозначение показано на рис. 6.74. Диод при этом подразумевается по умолчанию, и его можно не обозначать на схеме.

Принципиальная схема типичного элемента ТТЛШ изображена на рис. 6.75. Это элемент И-НЕ в положительной логике.

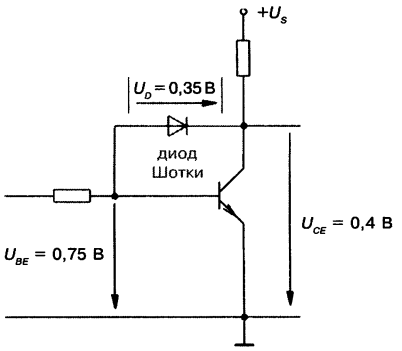


Рис. 6.73. Транзисторный каскад на диоде Шотки.

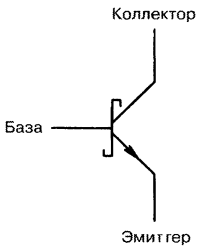


Рис. 6.74. Условное обозначение транзистора Шотки.

Среднее время задержки t_P составляет от 2,5 до 3 нс, или примерно вдвое меньше, чем для элементов подсемейства высокоскоростных ТТЛ.

Так как транзисторы Шотки проводят слабо, выходной уровень L у них выше, чем у стандартных ТТЛ-элементов. Вследствие этого разрыв между уровнями L и H меньше, что означает ухудшение статической помехоустойчивости.

> Логические элементы подсемейства ТТЛШ характеризуются очень высоким быстродействием, плохой помехоустойчивостью и высоким энергопотреблением.

Элемент И-НЕ (рис. 6.75) потребляет мощность 20 мВт — вдвое больше, чем элемент И-НЕ подсемейства стандартных ТТЛ.

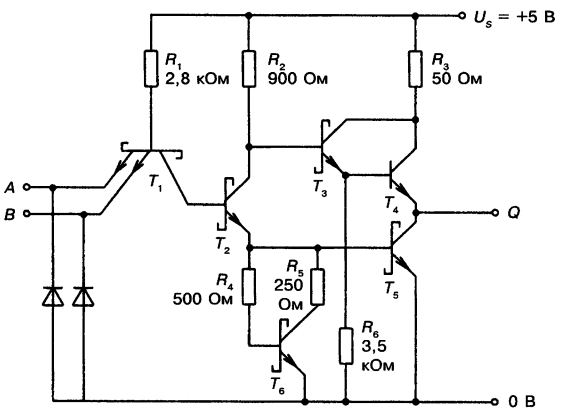


Рис. 6.75. Элемент ТТЛШ 74 S 00 (Texas Instruments).

6.6.6. ТТЛШ с пониженным энергопотреблением (Low-Power ТТЛШ)

Элементы подсемейства ТТЛШ потребляют меньше энергии при увеличении сопротивлений в токовых контурах. Мы уже отмечали этот эффект при рассмотрении подсемейства ТТЛ с пониженным энергопотреблением (разд. 6.6.3).

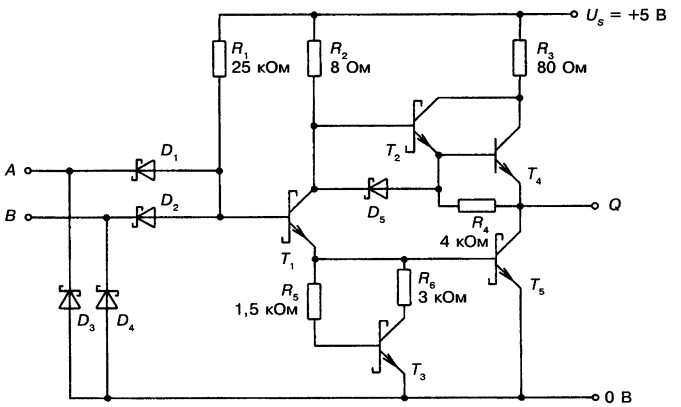


Рис. 6.76. Элемент ТТЛШ 74 LS 00 (Texas Instruments).

При увеличении сопротивлений время заряда-разряда емкостей транзистора возрастает и, следовательно, быстродействие ТТЛШ с пониженным энергопотреблением понижается.

Структура ТТЛШ с пониженным энергопотреблением совпадает со структурой обычных ТТЛШ. Только сопротивления берутся больше. На рис. 6.76 изображена схема типичного ТТЛШ с пониженным энергопотреблением. Среднее время задержки t_p составляет 9,5 нс, энергопотребление — 2 мВт.

> Быстродействие ТТЛШ с пониженным энергопотреблением и обычных ТТЛШ практически одинаковы. При этом они потребляют только 1/5 энергии обычных ТТЛШ.

Недостатком обоих подсемейств ТТЛШ, обычного и с пониженным энергопотреблением, является низкая помехоустойчивость по сравнению со стандартными ТТЛ.

6.6.7. Сравнительная оценка логических элементов

Идеальный логический элемент должен обладать очень высоким быстродействием, другими словами, среднее время задержки должно стремиться к нулю. Также он должен потреблять мало энергии и обладать высокой помехоустойчивостью. Эти три свойства являются взаимоисключающими. Если стремиться к высокому быстродействию, то проигрываешь в энергопотреблении и помехоустойчивости. Если стремиться к малому энергопотреблению, то проигрываешь в быстродействии.

При выборе между быстродействием, энергопотреблением и помехоустойчивостью приходится искать компромисс. Свой компромисс для каждого конкретного случая.

> Для каждого подсемейства ТТЛ был найден компромисс между быстродействием, энергопотреблением и помехоустойчивостью.

Схемотехника ТТЛ постоянно совершенствуется. Появляются новые улучшенные схемы, в которых удается еще немного уменьшить энергопотребление и увеличить быстродействие при сохранении высокой помехо-

Подсемейства ТТЛ	Стандартные ТТЛ	ТТЛ с пониженным энергопотреблением	Высокоскоростные ТТЛ	ТТЛШ	ТТЛШ с пониженным энергопотреблением	Улучшенные ТТЛШ	Улучшенные ТТЛШ с пониженным энергопотреблением
Номер серии	74 00	74 L00	74 H00	74 S00	74 LS00	74 AS00	74 ALS00
Напряжение питания	5 В	5 В	5В	5 В	5 В	5 В	5 В
Энергопотребление (на элемент)	10 мВт	1 мВт	23 мВт	20 мВт	2 мВт	8 мВт	1,2 мВт
Быстродействие	10 нс	33 нс	5 нс	3 нс	9,5 нс	1,7 нс	4 нс
Макс. частота переключения	40 МГц	13 МГц	80 МГц	130 МГц	50 МГц	230 МГц	100 МГц
Разность между уровнями	1 В	1 В	1 В	0,5 В	0,6 В	0,4 В	0,5 В

устойчивости. Это становится возможным благодаря прогрессу в производстве интегральных микросхем. В последнее время появились новые подсемейства ТТЛ: улучшенные ТТЛШ (Adwanced-Schottky-TTL) и улучшенные ТТЛШ с пониженным энергопотреблением (Adwanced-Low-Power-Schottky-TTL). Наиболее важные параметры подсемейств интегральных микросхем приведены в следующей таблице.

6.7. Эмиттерно-связанная логика (ЭСЛ)

Схемы семейства эмиттерно-связанной логики создаются на базе биполярных транзисторов. При создании этого семейства преследовалась цель достижения максимально возможного быстродействия. Большое быстродействие возможно только при переключении транзистора до того, как он войдет в режим насыщения (см. Бойт/Шмуш, Электроника, ч. 3, разд. 5.3).

Кажется, наиболее простым решением было бы построить схемы на базе простых усилителей. В таких усилителях транзисторы никогда не заперты полностью и никогда не проводят полностью. Переключение между состояниями открыт—закрыт происходит чрезвычайно быстро. Однако у усилителей на транзисторах много проблем с помехоустойчивостью. Разница между уровнями L и H мала, и состояние выхода нестабильно. Даже изменение температуры может повлиять на выходной уровень, т. е. вызвать самопроизвольные переходы с L на H и наоборот.

В дифференциальном усилителе (см. Бойт/Шмуш, Электроника, ч. 3, разд. 3.7.2) возможно запереть один транзистор и управлять другим. Для управляемого транзистора имеет место сильная обратная связь по току. Вследствие этого малые изменения тока базы управляемого транзистора не влияют на ток коллектора. Выходной уровень остается стабилен, хотя управляемый транзистор не находится в насыщенном состоянии.

Рассмотрим схему на рис. 6.77. На ней изображена принципиальная схема дифференциального усилителя. К базе транзистора T_2 постоянно приложено напряжение, например +2,7 В. Допустим, к базе T_1 тоже приложено напряжение +2,7 В. Оба транзистора открыты, и по эмиттерному сопротивлению R_E течет общий эмиттерный ток примерно 2 мА.

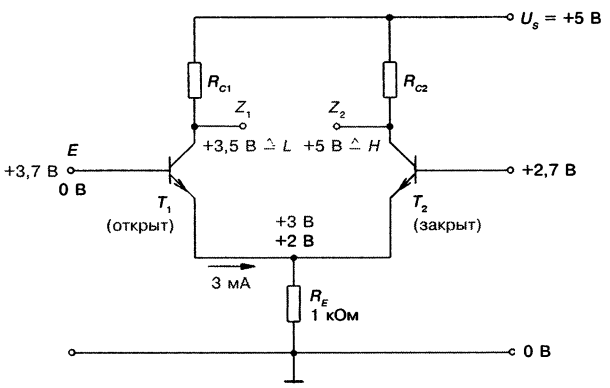


Рис. 6.77. Дифференциальный усилитель.

Если напряжение на базе T_1 становится выше чем +2,7 В, то составляющая тока эмиттера от T_1 увеличивается. На R_E увеличивается падение напряжения, и транзистор T_2 должен закрыться.

Если к базе T_1 приложено напряжение 3,7 В, то ток эмиттера равен примерно 3 мА. Сопротивление R_{C1} должно быть таким, чтобы напряжение коллектора T_1 не упало слишком сильно. T_1 может проводить только до тех пор, пока не приблизится к насыщению. Транзистор T_2 запирается, и ему не нужен эмиттерный ток.

К выходу Z_2 запертого транзистора T_2 прикладывается +5 В. Это напряжение высокого уровня. К выходу Z_1 запертого транзистора приложено напряжение, полученное в результате падения напряжения на R_E и напряжения U_{CE} проводящего транзистора. Оно равно примерно 3,5 В и относится к нижнему уровню.

Если напряжение базы T_1 немного падает или немного растет (например в диапазоне ±0,1 В), то уровни Z_1 и Z_2 практически не меняются. Они остаются стабильными. Если напряжение базы T_1 падает ниже 2,7 В, то происходит «опрокидывание». Транзистор T_2 начинает проводить и вынуждает транзистор T_1 закрыться. Выход Z_1 становится высоким уровнем, а выход Z_2 — низким. Коллекторное сопротивление R_{C2} должно быть подобрано так, чтобы T_2 не входил в состояние насыщения.

Параллельно с транзистором T_1 можно включать другие транзисторы (рис. 6.78). Такое параллельное соединение является эквивалентом проводного ИЛИ. При положительной логике на выходе Z_1 выполняется операция ИЛИ-НЕ. Так как выход Z_2 работает инверсно по отношению к Z_1, то на нем выполняется операция ИЛИ.

Уровни H и L выходов Z_1 и Z_2 не годятся для управления последовательно включенными логическими элементами. Поэтому к каждому из выходов подключают эмиттерный повторитель. За счет этого происходит смещение уровня и становится возможным управление большим количеством последовательно включенных элементов.

Типовое значение коэффициента разветвления по выходу находится в диапазоне от 20 до 30. На рис. 6.79 представлена принципиальная схема элемента ЭСЛ с эмиттерными повторителями на выходах.

Фиксированное постоянное опорное напряжение для транзистора T_2 на схеме (рис. 6.77) может быть получено с помощью делителя напряжения. В схеме FYH 124 (рис. 6.80) используется дополнительный транзис-

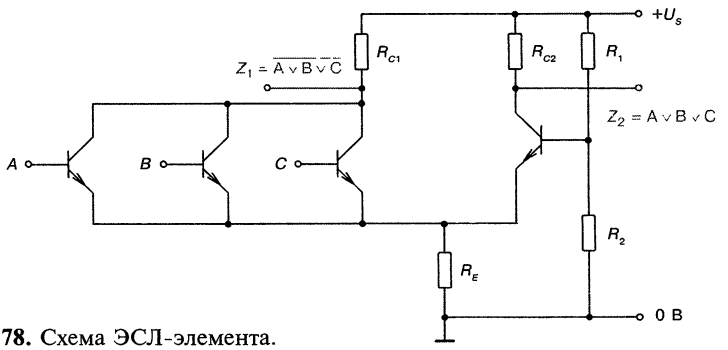


Рис. 6.78. Схема ЭСЛ-элемента.

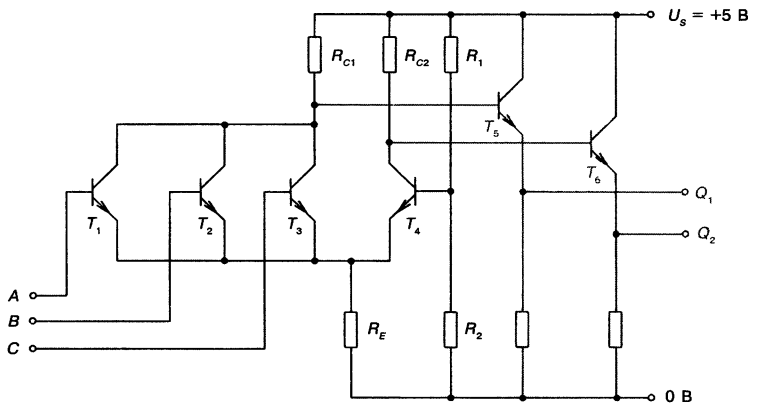
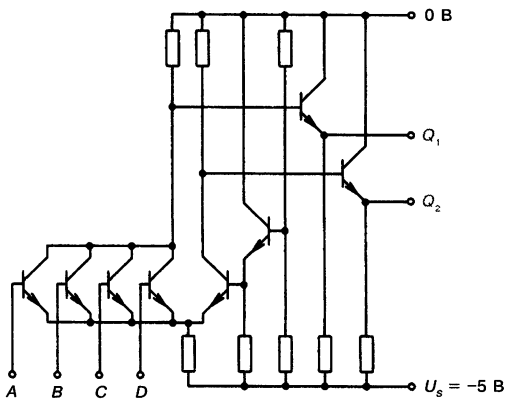


Рис. 6.79. ЭСЛ-элемент с эмиттерным каскадом.



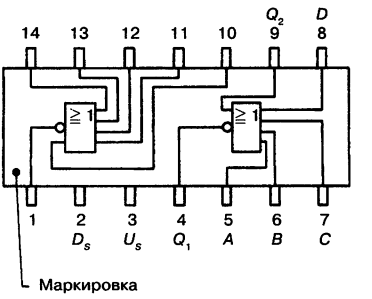


Рис. 6.80. ЭСЛ-элемент FYH 124 (Siemens).

тор, который обеспечивает стабильность опорного напряжения. Опорное напряжение определяет так называемый порог переключения.

Изготовители ЭСЛ-схем используют чаще всего отрицательное напряжение питания −5,0 В. Положительный полюс источника питания заземляется. Для *H*- и *L*-уровней получаются отрицательные значения напряжения. Диапазоны уровней схемы FYH 124 показаны на рис. 6.81. Отрицательное напряжение питания улучшает помехоустойчивость. Разница между напряжениями *L*- и *H*-уровней составляет всего около 0,4 В. Помехоустойчивость вследствие этого мала. Изготовители указывают в таблицах данных статическую помехоустойчивость 0,3 В. Вопрос помехоустойчивости рассмотрен подробнее в разд. 6.4.5.

Особым преимуществом схем ЭСЛ является хорошее быстродействие. Среднее время задержки t_p составляет 2 нс. Улучшенные технологии позволяют уменьшить t_p до 1 нс. В настоящее время разрабатывается такое супербыстродействующее ЭСЛ-подсемейство.

> Логические элементы подсемейства ЭСЛ являются самыми быстродействующими логическими элементами в настоящее время.

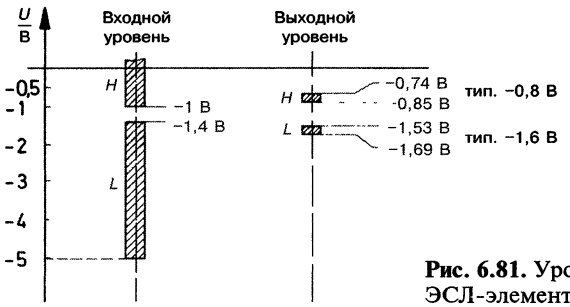


Рис. 6.81. Уровни напряжения ЭСЛ-элемента FYH 124.

Очень высокие скорости переключения вызывают проблемы с проводниками. При скорости 2 нс мы заходим в верхнюю мегагерцевую область (около 250 МГЦ). Провода активно излучают в высокочастотном диапазоне, что приводит к взаимному влиянию рядом проложенных проводников, то есть электромагнитная энергия излучается от одного проводника к другому и наоборот. Время переключения логического элемента сопоставимо с временем прохождения сигнала по проводнику. Возникает проблема электромагнитной совместимости и в соединениях между отдельными логическими элементами. При разводке схем следует также учитывать волновые сопротивления проводников (дорожек на плате).

> Схемы с ЭСЛ-элементами должны рассматриваться как высокочастотные схемы.

Необходимо применять экранирование. Если необходимо применять длинные провода, то они должны быть коаксиальными. Целесообразно применять при монтаже схемы высокоинтегральные микросхемы с высокой плотностью элементов. Дорожки на платах между микросхемами должны быть как можно короче.

В любом рабочем режиме в ЭСЛ-элементах всегда открыты много транзисторов, потребляющих соответствующие токи.

> Элементы ЭСЛ-подсемейства отличаются высоким энергопотреблением.

Каждый элемент потребляет примерно 60 мВт. Это в шесть раз больше, чем у стандартных ТТЛ-элементов. В следующей таблице приведены важнейшие параметры ЭСЛ-элементов:

Схемы подсемейства ЭСЛ	
Напряжение питания	−5 В
Энергопотребление (на вентиль)	60 мВт
Быстродействие	0,5 нс
Макс. частота переключения	1 ГГц
Типичный зазор между уровнями	0,3 В

ЭСЛ-элементы применяются в таких схемах, где высокая рабочая скорость имеет абсолютный приоритет. Наряду с немногими прикладными областями промышленных систем управления это прежде всего военные технологии.

6.8. Логические элементы на МОП-транзисторах

Семейство логических элементов на МОП-транзисторах состоит из полевых МОП-транзисторов. Полевые МОП-транзисторы почти не нуждаются в токе управления. Они имеют малые габариты и относительно просты в изготовлении. Возможно производство в виде интегральных микросхем с высокой плотностью. К недостаткам МОП-транзисторов относится большое время переключения из-за больших входных емкостей транзистора (затвор-исток).

6.8.1. Опасность статического электричества

Полевые МОП-транзисторы восприимчивы к статическим разрядам (см. Бойт, Электроника ч. 2, разд. 8.2). Опасность выхода из строя по причине разряда статического электричества существует вообще для всех интегральных микросхем.

> При работе с МОП-микросхемами нужно принимать особенные меры предосторожности против статических разрядов.

К мерам предосторожности относится электропроводящее напольное покрытие в помещениях, где работают с микросхемами. Каждый рабочий стол должен быть оборудован заземленной пластиной. Персонал не должен носить одежду из синтетики, например нейлона. Целесообразно носить электрически проводящую, заземленную через гибкий провод манжету.

Следующим опасным звеном является пайка. Переходное сопротивление паяльника между жалом и нагревательным элементом составляет примерно 100 кОм. Это сопротивление кажется достаточно большим, однако оно очень мало по сравнению с сопротивлениями между управляющим электродом и подложкой полевого МОП-транзистора. В паяльнике может накапливаться заряд, которые способен повредить МОП-микросхему.

> Для монтажа и демонтажа МОП-микросхем следует использовать особенные безопасные паяльники и паяльные ванны.

МОП-микросхемы, которые подверглись воздействию высоких напряжений, но продолжают на первый взгляд нормально работать, с большой вероятностью имеют внутренние повреждения. Такой вид повреждений называют «стресс полупроводника». Вследствие такого стресса снижается срок службы полупроводниковых элементов и повышается интенсивность отказов. Что произойдет при стрессе полупроводника внутри кристалла микросхемы — предсказать невозможно.

6.8.2. Логические элементы на *p*-канальных МОП-транзисторах (*p*МОП)

Логические элементы *p*МОП-подсемейства строятся на базе самозапирающихся *p*-канальных МОП-транзисторов. Схема простого *p*МОП-элемента показана на рис. 6.82. На выходе Z присутствует уровень L, если по крайней мере один из транзисторов заперт. На выходе Z присутствует уровень H, если

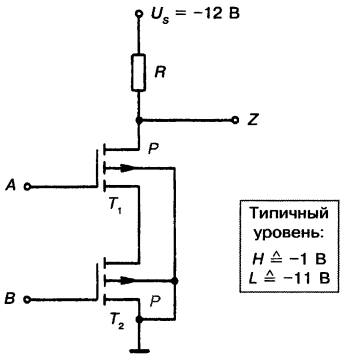


Рис. 6.82. Простой pМОП-элемент (ИЛИ-НЕ при положительной логике).

Вар.	B	A	Z
1	−11 В	−11 В	−1 В
2	−11 В	−1 В	−11 В
3	−1 В	−11 В	−11 В
4	−1 В	−1 В	−11 В

Вар.	B	A	Z
1	L	L	H
2	L	H	L
3	H	L	L
4	H	H	L

Рис. 6.83. Рабочая таблица к схеме на рис. 6.82.

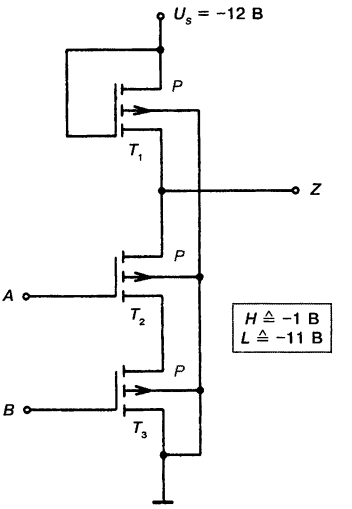


Рис. 6.84. Типовая схема *p*МОП-элемента (ИЛИ-НЕ при положительной логике).

на входах *A* и *B* действует уровень *L*, т. е. оба транзистора открыты. Соответствующая рабочая таблица представлена на рис. 6.83. В позитивной логике данная схема представляет элемент ИЛИ-НЕ.

Изготовление сопротивления *R* в кристалле полупроводника требует дополнительных усилий. Поэтому сопротивление *R* заменяют полевым транзистором с особыми свойствами. Типичная схема *p*МОП-элемента показана на рис. 6.84.

Рассмотрим транзистор T_1, который заменяет нагрузочное сопротивление *R*. Его затвор (управляющий электрод) присоединен к источнику напряжения питания. Если оба ключа-транзистора T_2 и T_3 открыты, напряжение на стоке T_1 и, следовательно, на выходе *Z* — примерно −1 В. Транзистор T_1 также открыт (U_{DC} = −11 В). Возникает опасность протекания слишком большого тока, и вследствие этого чрезмерно высокое напряжение выхода *Z*. Чтобы это не происходило, транзистор T_1 производится таким образом, чтобы сопротивление канала в проводящем состоянии не падало ниже 100 кОм. Транзисторы T_2 и T_3 имеют сопротивления канала от примерно 1 до 2 кОм в открытом состоянии. В закрытом состоянии сопротивление канала T_1 меньше сопротивлений каналов T_2 и T_3. В транзисторе T_1 оно равно примерно 1 МОм, в T_2 и T_3 — примерно 10 МОм. Если транзисторы T_2 и T_3 заперты, или хотя бы один из них заперт, на выходе *Z* действует напряжение примерно −11 В, т. е. напряжение *L*-уровня.

РМОП-элемент на рис. 6.85 производит при положительной логике операцию И-НЕ. Если на входе *A* или *B* действует уровень *L* (например −11 В), то выход *Z* будет иметь *H*-уровень (−1 В). То же самое происходит, если на обоих входах действует уровень *L*. Только если на обоих входах присутствует *H*-уровень, и, следовательно, ни один из транзисторов T_2 и T_3 не проводит, *Z* остается на уровне *L* (рис. 6.86).

На рис. 6.87 представлена схема *p*МОП-элемента НЕ. Транзистор T_1 заменяет нагрузочное сопротивление. Транзистор T_2 работает в качестве ключа. Уровень выхода всегда противоположен уровню входа.

Схемы на *p*МОП-элементах потребляют мало энергии. Напряжение питания может колебаться в большом диапазоне (от −9 В до −20 В). Чем

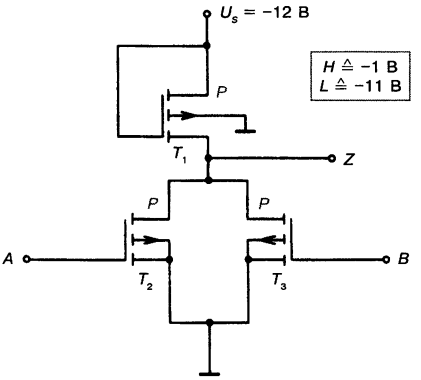


Рис. 6.85. Типовая схема *p*МОП-элемента (И-НЕ при положительной логике).

Вар.	B	A	Z
1	−11 В	−11 В	−1 В
2	−11 В	1 В	−1 В
3	−1 В	−11 В	−1 В
4	−1 В	−1 В	−11 В

Вар.	B	A	Z
1	L	L	H
2	L	H	H
3	H	L	H
4	H	H	L

Рис. 6.86. Рабочая таблица к схеме на рис. 6.85.

выше выбирается напряжение питания, тем больше будет статическая помехоустойчивость, так как с растущим напряжением питания интервал между *L*- и *H*-уровнями увеличивается. Диапазоны уровня для напряжения питания −12 В изображены на рис. 6.88.

Граничные значения и основные параметры приводятся в таблицах производителей. Технологии изготовления постоянно совершенствуются. Соответственно обновляются и параметры.

В следующей таблице приведены важнейшие характеристики *p*МОП-элементов:

Схемы подсемейства *p*МОП	
Напряжение питания	−12 В (от −9 до −20 В)
Энергопотребление (на вентиль)	6 мВт (при выходном уровне *H*) 0 мВт (при выходном уровне *L*)
Быстродействие	40 нс
Максимальная частота переключения	10 МГц
Типичный зазор между уровнями	5 В

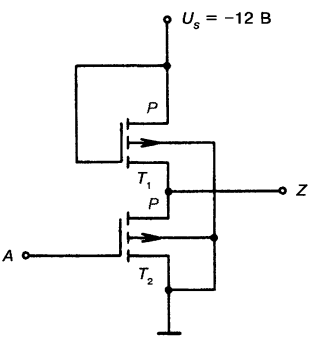


Рис. 6.87. Схема *p*МОП-элемента НЕ.

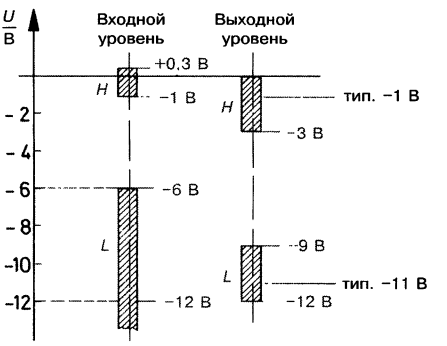


Рис. 6.88. Диапазоны напряжений *p*МОП-элементов (U_s = −12 В).

> *P*МОП-элементы работают медленно, но устойчиво. Они нуждаются в достаточно большом напряжении питания.

*P*МОП-элементы применяются в таких схемах, где высокая рабочая скорость не нужна. Интегральные микросхемы могут выполняться с большой плотностью. Благодаря простой технологии изготовления возможно промышленное производство небольших партий под конкретные задачи по заказу.

6.8.3. Логические элементы на *n*-канальных МОП-транзисторах (*n*МОП)

Логические элементы *p*МОП-подсемейства строятся на базе самозапирающихся *n*-канальных МОП-транзисторов. Применяют другую технологию изготовления, которая позволяет добиться еще большей плотности упаковки, чем у *p*МОП-элементов. Это становится возможным благодаря значительно меньшим сопротивлениям каналов. Их значение составляет примерно треть от значений сопротивлений каналов *p*МОП-элементов. Меньшее сопротивление каналов и меньшие входные емкости в микроструктуре транзисторов приводят к повышению быстродействия.

> *N*МОП-элементы работают примерно так же быстро, как стандартные ТТЛ-элементы. Их быстродействие составляет порядка 10 нс.

Меньшее сопротивление каналов позволяет снизить напряжение питания до 5 В. Это делает возможным совместное использование *n*МОП-элементов и ТТЛ-элементов. На техническом языке говорят, что они совместимы.

> *N*МОП-элементы совместимы с ТТЛ-элементами.

Схемы *n*МОП-элементов аналогичны по структуре схемам *p*МОП-элементов, только применяются *n*-канальные МОП полевые транзисторы. На рис. 6.89 приведены две типичные схемы. Левая схема производит при положительной логике операцию И-НЕ, правая — операцию ИЛИ-НЕ. Соответствующие диапазоны уровней показаны на рис. 6.90.

В следующей таблице приведены важнейшие параметры *n*МОП-элементов:

Схемы подсемейства *n*МОП	
Напряжение питания	+5 В
Энергопотребление (на вентиль)	2 мВт (при выходном уровне *L*) 0 мВт (при выходном уровне *H*)
Быстродействие	5 нс
Максимальная частота переключения	80 МГц
Типичный зазор между уровнями	≈ 2 В

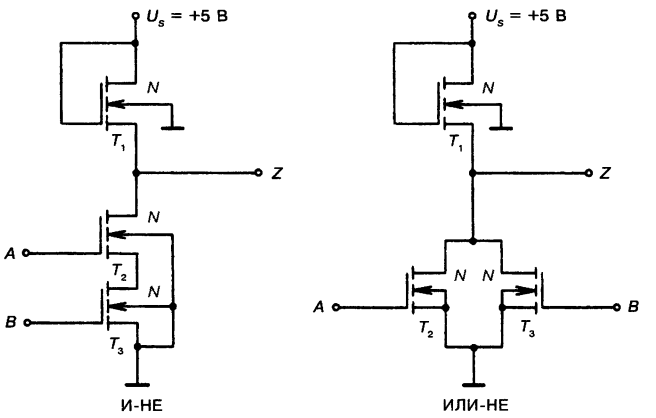


Рис. 6.89. *N*МОП-элементы.

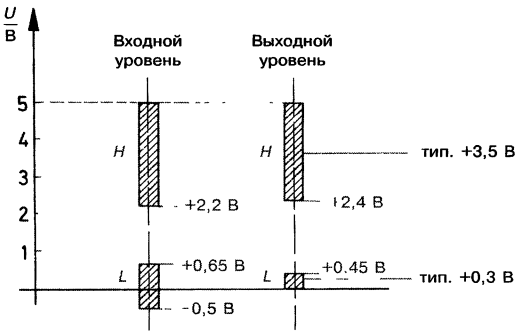


Рис. 6.90. Диапазоны напряжения *n*МОП-элементов рис. 6.89.

Отдельные логические элементы в виде интегральных микросхем сегодня практически не производятся в *n*-канальной МОП-технологии. В виде интегральных микросхем производят целые вычислительные системы, например умножители, кодировщики, счетчики. Производство более крупных интегральных микросхем является особенно выгодно с экономической точки зрения.

6.8.4. Логические элементы на КМОП-транзисторах

Сокращение КМОП означает «комплементарный МОП-транзистор». Также иногда используется сокращение COSMOS, которое обозначает «комплементарная симметричная МОП-структура». Логические элементы этого подсемейства строятся как на *n*-канальных МОП-полевых транзисторах, так и на *p*-канальных МОП-полевых транзисторах. Схемы этого подсемейства характеризуются ярко выраженной симметрией. При разработке схем применяют только самозапирающиеся МОП-транзисторы (см. Бойт, Электроника, ч. 2, разд. 8.2, МОП-полевые транзисторы).

Симметричность схем видна особенно хорошо в схеме элемента НЕ (рис. 6.91). Если на входе *A* действует *H*-уровень, например +5 В, то транзистор T_2 отпирается. На его истоке и подложке 0 В. Напряжение затвор-исток U_{GS} составляет +5 В. К истоку и подложке транзистора T_1 приложены

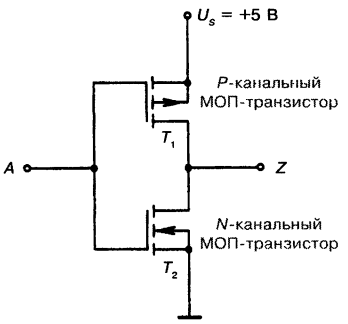


Рис. 6.91. Схема КМОП НЕ-элемента.

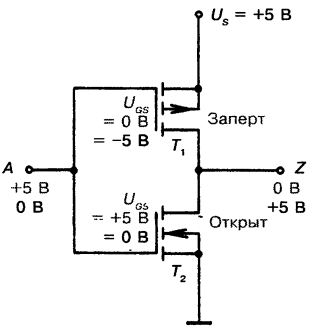


Рис. 6.92. Принцип действия КМОП НЕ-элемента.

+5 В. Если к управляющему электроду также прикладываются +5 В, то напряжение затвор-исток $U_{GS} = 0$ В. Транзистор T_1 заперт. Если T_1 заперт, а T_2 открыт, то выход элемента Z имеет уровень L (рис. 6.92).

Если на входе A действует L-уровень 0 В, то транзистор T_2 запирается и напряжение затвор-исток U_{GS} составляет 0 В. Напряжение затвор-исток транзистора T_1 $U_{GS} = -5$ В, так как напряжение истока +5 В, а затвора 0 В. Транзистор T_1 отпирается. Если T_1 открыт, а T_2 заперт, выход элемента Z имеет уровень H.

В КМОП-НЕ-элементе всегда один транзистор открыт, а другой заперт.

Если на выходе элемента НЕ действует уровень 0, то элемент практически не потребляет ток, так как T_1 заперт. Если на выходе элемента НЕ действует уровень H, то элемент также практически не потребляет ток, так как теперь T_2 заперт. Для управления последовательно включенными элементами также не требуется ток, так как полевые транзисторы практически не потребляют мощность. Только во время переключения от источника питания потребляется небольшой ток, так как оба транзистора одновременно, но недолго открыты. Один из транзисторов переходит из открытого состояния в запертое и еще не полностью заперт, а другой — из запертого в открытое и еще не полностью открыт. Также должны перезарядиться транзисторные емкости.

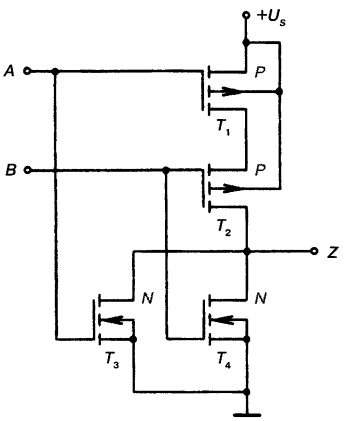
Все КМОП-элементы устроены так, что в токовой ветви один транзистор закрыт, а другой открыт. Энергопотребление КМОП-элементов крайне низко. Оно зависит в основном от количества переключений в секунду или частоты переключения.

КМОП-элементы отличаются малым энергопотреблением.

На рис. 6.93 изображена следующая типичная КМОП-схема. Если на обоих входах действует уровень L, то транзисторы T_1 и T_2 будут открыты, транзисторы T_3 и T_4 заперты. T_1 и T_2 при 0 В на A и B имеют $U_{GS} = -5$ В, а T_3 и T_4 имеют $U_{GS} = 0$ В. На выходе Z действует уровень H.

Если на входе A действует уровень H (+5 В), а на входе B-уровень L (0 В), то T_1 закрывается, а T_2 открывается. Путь от источника питания к выходу Z блокирован запертым транзистором. Одновременно отпирается транзистор T_3

Рис. 6.93. Схема КМОП ИЛИ-НЕ-элемента (при положительной логике).



Вар.	B	A	Z
1	L	L	H
2	L	H	L
3	H	L	L
4	H	H	L

Рис. 6.94. Рабочая таблица схемы рис. 6.93.

и на выходе Z действует примерно 0 В, то есть уровень L. T_4 заперт. Z всегда имеет уровень L, если по крайней мере на одном входе действует уровень H. Соответствующая схеме (рис. 6.93) рабочая таблица представлена на рис. 6.94. Схема производит при положительной логике операцию ИЛИ-НЕ.

Какую логическую операцию производит схема на рис. 6.95? Прежде всего для схемы должна быть составлена рабочая таблица. Если на обоих входах действуют L-уровни (0 В), то транзисторы T_1 и T_2 открываются ($U_{GS} = -5$ В). Транзисторы T_3 и T_4 закрываются ($U_{GS} = 0$ В). На выходе H-уровень.

Если на обоих входах действуют H-уровни (+5 В), то транзисторы T_3 и T_4 открываются, а транзисторы T_1 и T_2 закрываются. На выходе Z будет действовать L-уровень.

Если на один вход приложен H-уровень, а на другой — L-уровень, то один из верхних транзисторов на рис. 6.95 (T_1 или T_2) открывается. Один из нижних (T_3 или T_4) запирается. Через открытые транзисторы к выходу будет прикладываться H-уровень. На рис. 6.96 представлена соответствующая таблица истинности. Схема выполняет при положительной логике функцию И-НЕ.

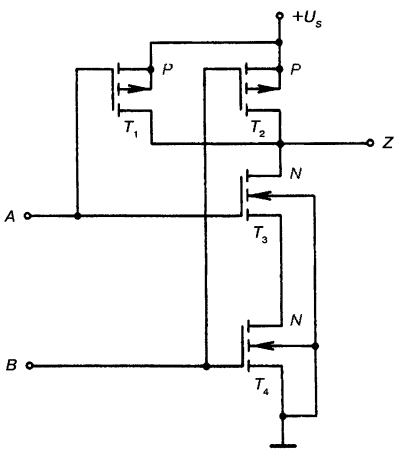


Рис. 6.95. Схема КМОП И-НЕ-элемента (при положительной логике).

Вар.	B	A	Z
1	L	L	H
2	L	H	H
3	H	L	H
4	H	H	L

Рис. 6.96. Рабочая таблица схемы на рис. 6.95.

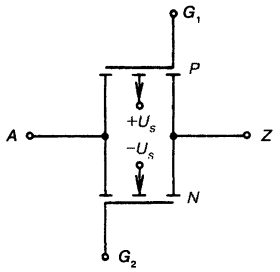


Рис. 6.97. Принципиальная схема передаточного логического элемента.

$L \triangleq 0$ В
$H \triangleq +5$ В

Вар.	G_2	G_1	
1	L	H	R_{AZ} высокоомный
2	H	L	R_{AZ} низкоомный

Рис. 6.98. Рабочая таблица передаточного логического элемента.

КМОП-элементы производятся в основном в виде элементов И-НЕ и ИЛИ-НЕ.

Особым элементом подсемейства КМОП является передаточный элемент. Он состоит из параллельного включенных *n*-канального МОП-транзистора и *p*-канального МОП-транзистора (рис. 6.97).

Передаточный элемент работает как переключатель.

Если к G_1 будет приложен уровень H (например +5 В) и к G_2 — уровень L (0 В), то оба транзистора запираются. В *p*-канальном МОП-транзисторе между управляющим электродом и подложкой приложено напряжение 0 В. Образование проводящего канала между истоком и стоком становится невозможным. Также и в *n*-канальном МОП-транзисторе между управляющим электродом и подложкой приложено напряжение 0 В. Здесь также не может возникнуть проводящий канал. Сопротивление между точками A и Z достигает нескольких сотен МОм.

Если на G_1 действует уровень L (0 В), а на G_2 — уровень H (+5 В), то напряжение затвора *p*-канального МОП-транзистора относительно подложки будет −5 В. Напряжение затвора *n*-канального МОП-транзистора относительно подложки +5 В. При этих напряжениях образуются проводящие каналы между истоком и стоком. Канал между A и Z будет низкоомным (примерно от 200 Ом до 400 Ом). Рабочая таблица представлена на рис. 6.98.

Уровни на входах G_1 и G_2 всегда прикладываются в противофазе. Управление может происходить с помощью элемента НЕ (рис. 6.99). Получается двунаправленный ключ. У полевых транзисторов передаточного элемента исток и сток могут взаимно менять свои функции. Поэтому вывод затвора обозначается в середине его условной линии (рис. 6.99).

Интегрированные КМОП-микросхемы всегда содержат множество логических элементов, которые могут быть использованы по отдельности или как единая сложная логическая функция. На рис. 6.100 показана структура схемы CD 4000 A. Эта схема содержит два элемента ИЛИ-НЕ с тремя входами каждый и элемент НЕ. Схема CD 4012 A (рис. 6.101) содержит два элемента И-НЕ с четырьмя входами каждый.

Интегральные схемы арифметических логических устройств содержат очень много КМОП-элементов. На рис. 6.102 приведена схема 4-битного сдвигающего регистра. Эта схема рассмотрена подробно в гл. 8.

Рис. 6.99. Передаточный логический элемент с управляющим элементом НЕ.

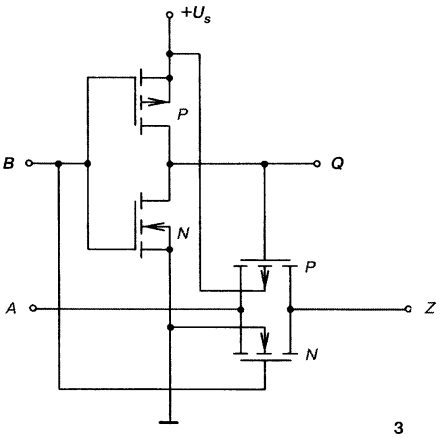
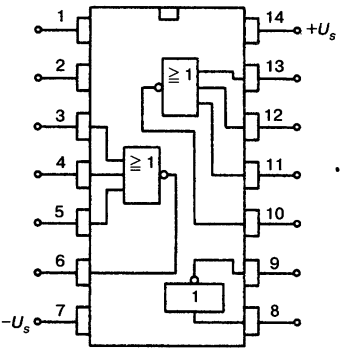
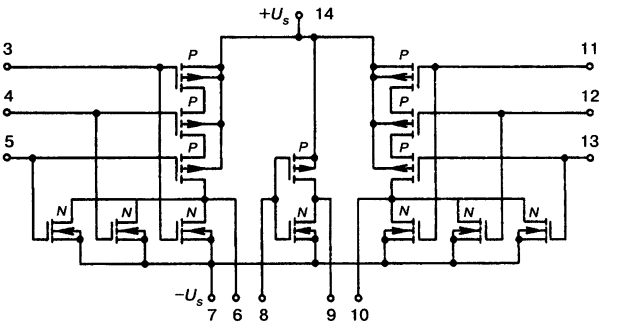


Рис. 6.100. КМОП-микросхема CD 4000 A (RCA).





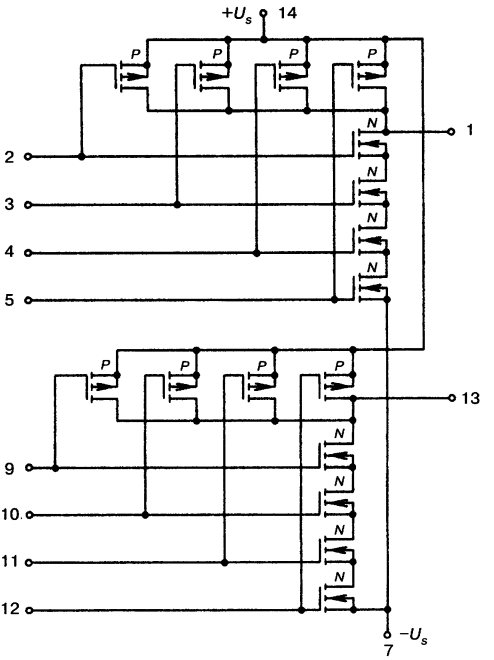


Рис. 6.101. КМОП-микросхема CD 4012 A (RCA).

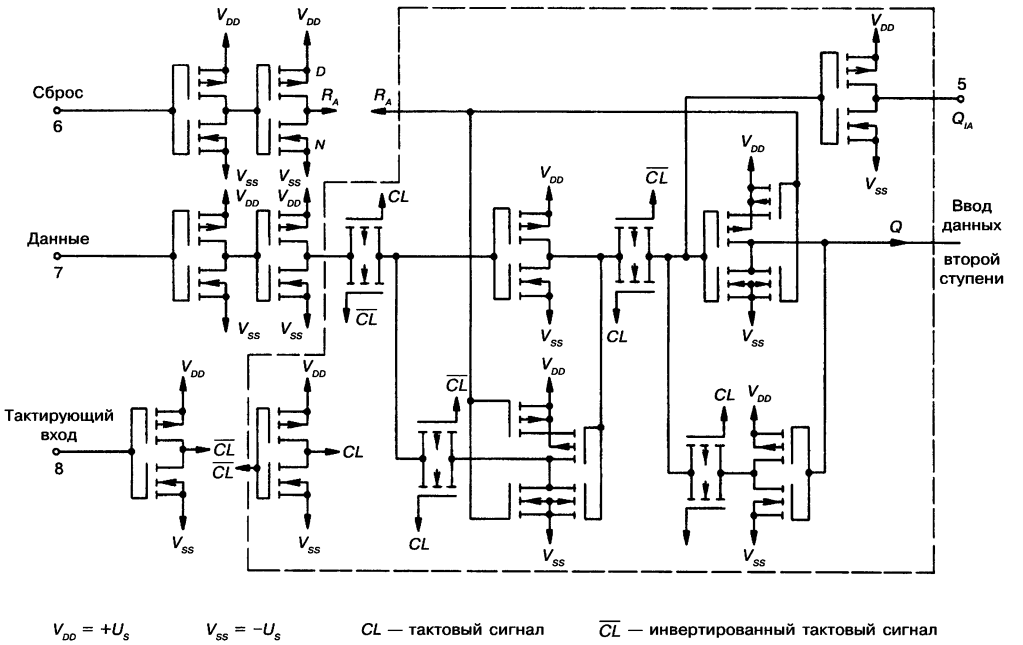


Рис. 6.102. Схема КМОП-4-битного сдвигового регистра CD 4015 A (RCA).

Микросхема CD 4008 A является 4-битным полным сумматором. Полные сумматоры рассматриваются подробно в гл. 10. Схема приведена здесь как пример КМОП-схемотехники (рис. 6.103).

> Интегральные микросхемы в КМОП-исполнении могут производиться с очень большой плотностью элементов.

Можно схему целого калькулятора уместить в одной микросхеме. Дальнейшее совершенствование технологий ведет к повышению возможной плотности компоновки.

> Напряжение питания КМОП-элементов может колебаться в широком диапазоне.

Для серии CD-4000-A (рис. 6.100—6.103) фирма-производитель RCA указывает диапазон напряжений питания от 3 В до 15 В. Типичные передаточные характеристики при ряде напряжений питания показаны на рис. 6.104.

Часто используются напряжения питания +5 В и +10 В. Для этих напряжений питания на рис. 6.105 и 6.106 показаны диаграммы уровней. Для больших напряжений питания характерна лучшая помехоустойчивость.

> Разность между уровнями *L* и *H*, отвечающая за помехоустойчивость, для КМОП-схем составляет примерно от 30 % до 40 % напряжения питания.

В следующей таблице приведены важнейшие параметры КМОП-элементов:

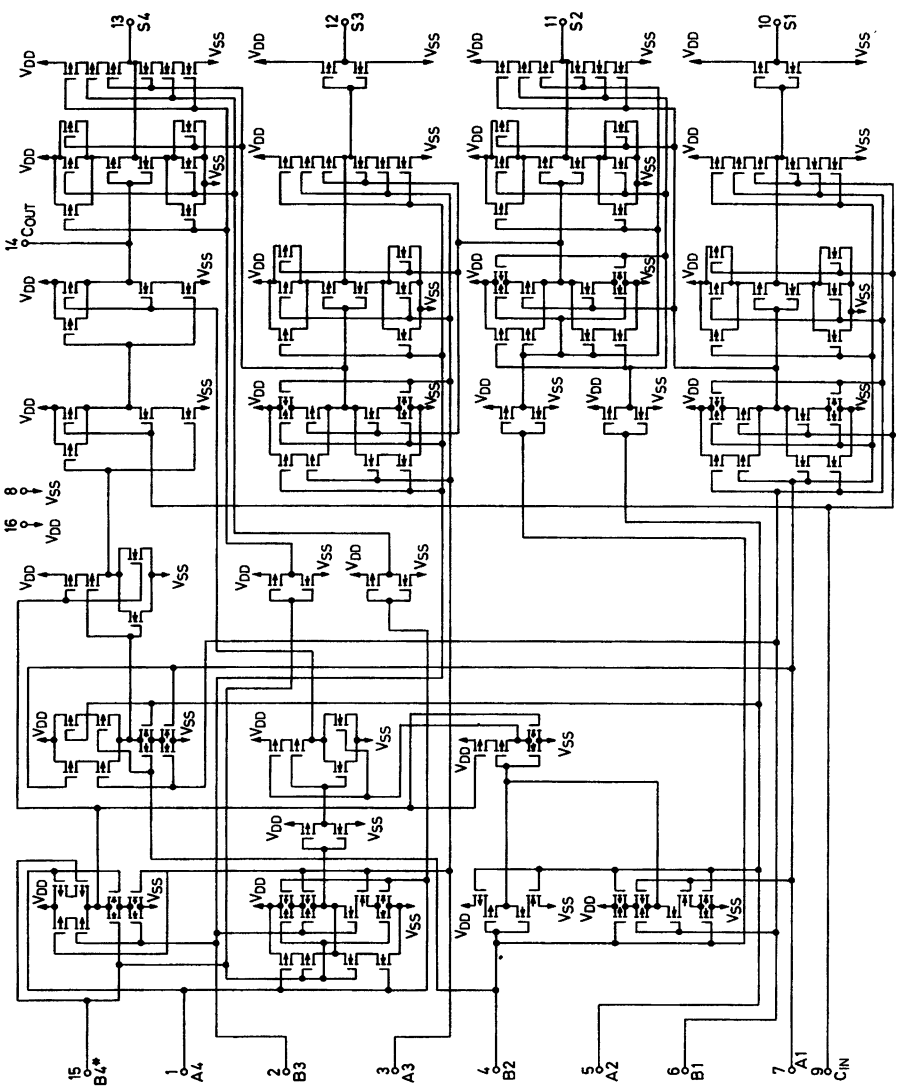


Рис. 6.103. Схема КМОП-4-битного полного сумматора CD 4008 A (RCA).

Схемы подсемейства *n*МОП	
Напряжение питания	например +5 В
Энергопотребление (на вентиль)	от 5 до 10 нВт (в зависимости от частоты переключения)
Быстродействие	8 нс
Максимальная частота переключения	50 МГц
Типичный зазор между уровнями	2 В
Входное сопротивление	10^{12} Ом
Выходное сопротивление	
При *H*-уровне	500 Ом
При *L*-уровне	200 Ом
Коэффициент разветвления	> 50
Входной ток	10 пА (максимально)

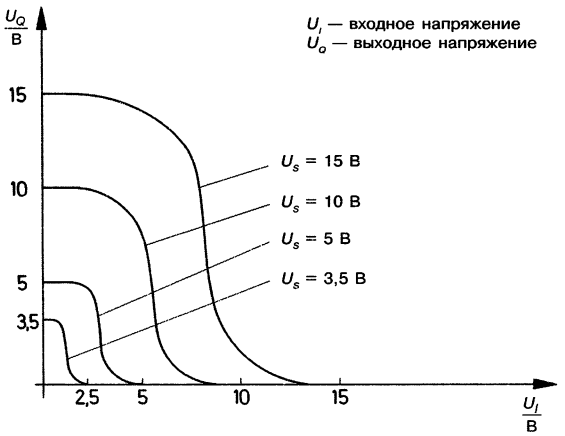


Рис. 6.104. Передаточные характеристики КМОП-элемента для нескольких напряжений питания.

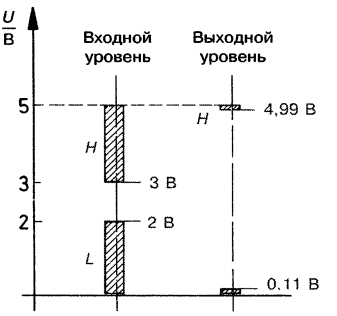


Рис. 6.105. Диаграмма уровней для U_s = 5 В.

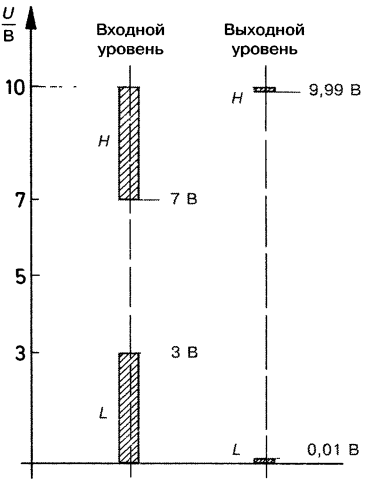


Рис. 6.106. Диаграмма уровней для U_s = 10 В.

Входы современных КМОП-микросхем защищены от воздействия статических разрядов. В качестве защиты используют диодные схемы согласно рис. 6.107. Если напряжение входа превышает напряжение питания на величину примерно 0,7 В (напряжение открытия диода), диод D_1 откроется и лишний ток уйдет к источнику питания. В схеме на рис. 6.107 это происходит при входных напряжениях выше +5,7 В. При отрицательных входных напряжениях, начиная с −0,7 В, открывается диод D_2.

Концентрация основных носителей заряда в полупроводниковых зонах диодов очень мала, поэтому они обладают большим динамическим сопротивлением.

Вследствие этого очень высокое входное сопротивление КМОП-элементов только незначительно уменьшается при добавлении диодов. Кроме того, малое динамическое сопротивление диодов при ошибочной полярности питания привело бы к их мгновенному выходу схемы из строя, так как через D_1 и D_2 потек бы ток короткого замыкания.

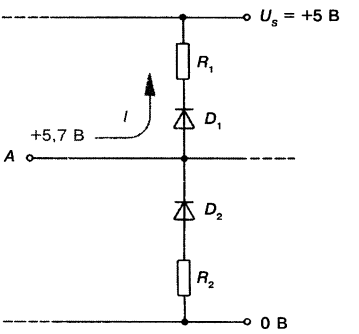


Рис. 6.107. Защитная схема против статического разряда.

Если на вход элемента подается большой статический заряд, то соответствующий полевой транзистор может выйти из строя, несмотря на диодную защиту. Из-за больших динамических сопротивлений заряд может не успеть достаточно быстро рассосаться, так что может возникнуть опасное высокое напряжение. При работе с МОП-схемами следует принимать обычные меры предосторожности против статического электричества.

КМОП-схемы имеют широкую область применения. Они достаточно быстры для решения многих технических задач и отличаются малым энергопотреблением. Даже микросхемы высокой степени интеграции очень выгодны при производстве в больших количествах.

Контрольный тест

1. Что понимают под схемами одного семейства?
2. Назовите основные семейства логических схем.
3. Что такое бинарные уровни напряжения?
4. Что понимают под положительной логикой, под отрицательной?
5. Чем отличается рабочая таблица от таблицы истинности?
6. Составьте для схемы на рис. 6.108 рабочую таблицу.
7. Какую логическую операцию производит схема на рис. 6.108 при положительной и при отрицательной логике?
8. На рис. 6.109 показаны два диапазона уровня. Какой является *H*-диапазоном и какой *L*-диапазоном?

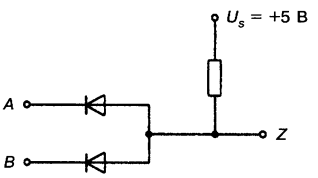


Рис. 6.108. Логический элемент.

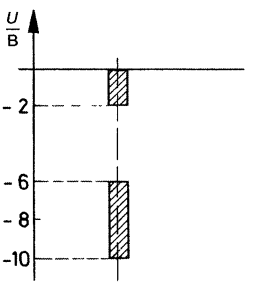


Рис. 6.109. Диапазоны уровней.

9. Что понимают под передаточной характеристикой? Нарисуйте примерную передаточную характеристику.

10. Разъясните понятия «быстродействие» и «время нарастания сигнала».

11. Что такое входной нагрузочный коэффициент?

12. В таблице данных на элемент указано: коэффициент разветвления $F_Q = 10$. Что это значит?

13. Как устроены ДТЛ-элементы?

14. Нарисуйте схему типового ТТЛ-элемента с тремя входами и двухтактным выходом.

15. Какую операцию производит схема на рис. 6.110 при положительной логике?

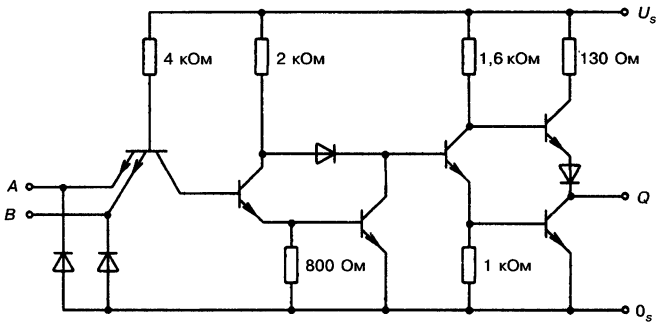


Рис. 6.110.

16. Часто можно услышать понятия «насыщенный элемент» и «ненасыщенный элемент». Что значат эти понятия?

17. Какие преимущества и недостатки имеет подсемейство ТТЛ-схем с пониженным энергопотреблением по отношению к стандартным ТТЛ-элементам?

18. Как устроены *n*МОП-элементы?

19. Упорядочите известные Вам семейства логических элементов

а) по энергопотреблению,

б) по быстродействию.

20. Нарисуйте схему КМОП-элемента с двумя входами, которая производит при положительной логике операцию ИЛИ-НЕ, и объясните принцип действия этой схемы.

ГЛАВА 7

БИНАРНЫЕ СХЕМЫ С ВРЕМЕННО́Й ЗАВИСИМОСТЬЮ

7.1. Введение

Триггерами называют бистабильные ячейки. Они обладают свойством памяти. Устройство бистабильной ячейки рассмотрено в «Электронике», ч. 3, гл. 7.

Триггеры производятся преимущественно в виде интегральных микросхем.

Условное графическое обозначение простого триггера приведено на рис. 7.1. Соотношение сторон прямоугольника может быть выбрано любым (рис. 7.2).

Это графическое представление триггера соответствует DIN 40900, ч. 12. Главным выходом является выход A_1, который, разумеется, может называться как угодно. Выход A_2, инверсный по отношению к A_1, обозначается кругом инверсии. Также до сих пор допустим традиционный способ представления триггеров в виде прямоугольника со средней линией (рис. 7.2а), согласно DIN 40700, часть 14.

Действуют следующие правила:

1. Выводы для подключения напряжения питания не обозначаются.

2. На обоих выходах триггера обычно действуют противоположные состояния.

3. Для описания принципа действия триггера используются логические состояния 0 и 1. Также можно оперировать понятиями уровней L и H. Если нет особых указаний, то по умолчанию все логические операции производятся при положительной логике ($H = 1$, $L = 0$).

4. При подаче на вход триггера E_1 логической 1, выход A_1 переходит в состояние $A_1 = 1$. Этот процесс называют **режимом установки триггера**. Если триггер уже имеет состояние $A_1 = 1$, то подача 1 на входе E_1 ничего не меняет. Переключения триггера не происходит.

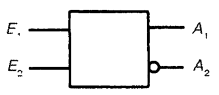


Рис. 7.1. Условное графическое обозначение простого триггера.

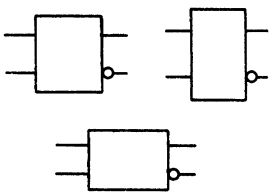


Рис. 7.2. Условное графическое обозначение триггера с разными соотношениями сторон прямоугольника.

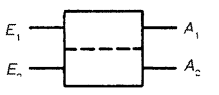


Рис. 7.2а. Условное графическое обозначение простого триггера со средней линией.

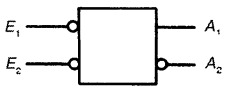


Рис. 7.3. Условное графическое обозначение триггера с управлением 0-состояниями.

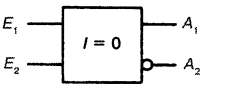


Рис. 7.4. Условное графическое обозначение триггера с фиксированной установкой $I = 0$.

5. Состояние 1 на E_2 переводит выход триггера в $A_2 = 1$. Этот процесс называют **режимом сброса триггера**. Если триггер уже имеет состояние $A_2 = 1$, то 1 на входе E_2 ничего не меняет.

6. Состояния 0 на входах не вызывают никакого воздействия на выход.

7. Состояние A_1 характеризует состояние памяти триггера. Если $A_1 = 1$, то триггер записал значение 1.

Разумеется, можно сделать триггеры, которые управляются 0-состояниями. Эти триггеры имеют особенные, обозначенные кругом отрицания входы (рис. 7.3) и применяются достаточно редко.

Часто применяют триггеры с фиксированной установкой. Условное обозначение такого триггера показано на рис. 7.4. После подачи напряжения питания этот триггер всегда устанавливается в состояние $A_1 = 0$, $A_2 = 1$. Это состояние называется **состояние покоя** или **состояние сброса**. Выход, который имеет в состоянии покоя значение 1, раньше обозначался толстой полосой.

> Условное обозначение фиксированной установки можно опустить, если не возникает неоднозначности.

Состояние $A_1 = 1$, $A_2 = 0$ называется **рабочим состоянием** или режимом установки триггера. Можно так сконструировать триггер, чтобы он после подачи напряжения входил в рабочее состояние ($A_1 = 1$, $A_2 = 0$). Такой триггер обозначается символом $I = 1$ (рис. 7.5).

В настоящее время появились триггеры, которые после подачи напряжения питания имеют состояние, которое они имели при выключении напряжения. Они не теряют хранимую информацию при потере питания. На триггерах этой разновидности стоит символ *NV* (non volatile, nullspannungsgesichert). Такой триггер показан на рис. 7.6.

Различают между статическими и динамическими входами. Рассмотренные до сих пор входы являются статическими.

> Статические входы реагируют на состояния входов.

> Динамические входы реагируют на изменения состояния на входах.

Существуют две разновидности динамических входов. Первая реагирует на изменение входного состояния с 0 на 1. Такой вход называется динамическим входом для прямого фронта импульса (рис. 7.7). Динамический вход второй разновидности реагирует на изменение входного состояния с 1 на 0.

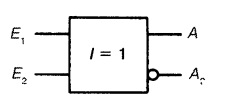


Рис. 7.5. Условное графическое обозначение триггера, который входит в рабочее состояние после подачи напряжения.

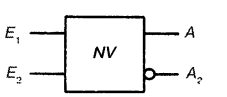


Рис. 7.6. Условное графическое обозначение триггера с запоминанием состояния перед выключением.

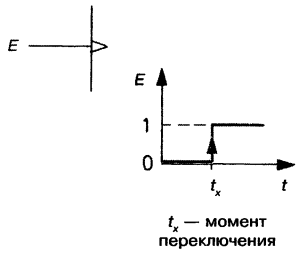


Рис. 7.7. Изображение динамического входа для прямого фронта (0 → 1).

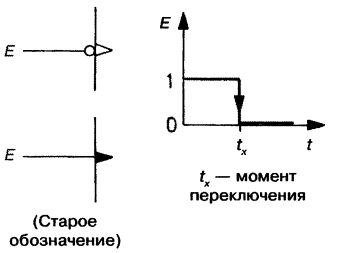


Рис. 7.8. Изображение динамического входа для обратного фронта (1 → 0).

Такой вход называется динамическим входом для обратного фронта импульса (рис. 7.8).

Триггеры могут иметь несколько взаимосвязанных входов.

Триггер на рис. 7.9 имеет входы E_1 и E_3, которые связаны операцией логического умножения И. Если входы связаны операцией логического сложения ИЛИ, то изображается символ логического сложения (рис. 7.10).

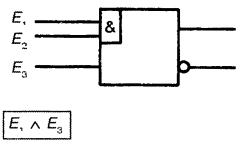


Рис. 7.9. Условное графическое обозначение триггера, у которого входы E_1 и E_3 связаны логическим умножением И.

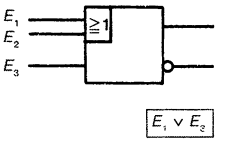


Рис. 7.10. Условное графическое обозначение триггера, у которого входы E_1 и E_3 связаны логическим сложением ИЛИ.

Функции входов триггеров и логические операции условно обозначаются с помощью следующих знаков:

$G \Rightarrow$ Логическое И

$V \Rightarrow$ Логическое ИЛИ

$C \Rightarrow$ Управляющий вход

$S \Rightarrow$ Вход установки

$R \Rightarrow$ Вход сброса

Связанные входы обозначаются буквами с дополнительными цифровыми индексами.

Для управляющих входов индекс ставится после букв.

Для управляемых входов индекс ставится перед буквами.

Поясним на примере. Триггер имеет входы S и R, связанные с третьим входом операцией логического умножения И, как изображено на рис. 7.12, И-элементы могут быть изображены перед триггером или занесены непосредственно в прямоугольник триггера. Такое изображение логических взаимосвязей принято до сих пор.

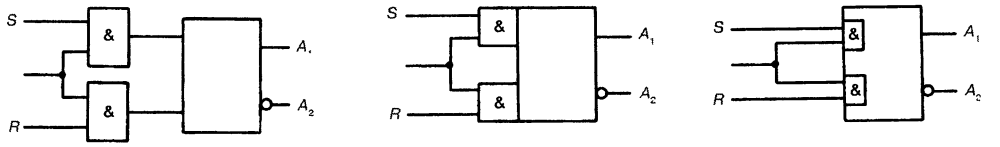


Рис. 7.11. Принятое обозначение логической связи между входами.

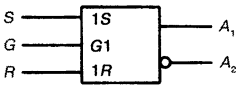


Рис. 7.12. Обозначение логической связи между входами с помощью условного знака.

С помощью условных знаков изображение значительно упрощается. Третий вход называется из-за умножения И входом *G*. Он является управляющим входом, так как связан с обоими другими входами.

Он получает индекс 1 после буквы. Такой же индекс — в данном случае 1 — получают входы *S* и *R* (рис. 7.13). Он пишется перед буквами.

S является входом сигнала установки. Состояние 1 на *S* и состояние 1 на *G* устанавливают триггер в рабочее состояние. *R* является входом сброса. Состояние 1 на *R* и состояние 1 на *G* устанавливают триггер в состояние сброса.

Как поведет себя триггер, если на всех входах *S*, *R* и *G* одновременно будет состояние логической 1? В большинстве триггеров это запрещенная комбинация, которая приводит к неопределенному состоянию выходов. Тем не менее есть триггеры, у которых либо *S*-вход либо *R*-вход являются доминирующими.

На рис. 7.13а изображен триггер с доминирующим *S*-входом. Управляющего И-входа нет и *G*-вход отсутствует. При $S = 1$ и $R = 1$ триггер будет установлен. Индекс 1 отражает так называемую *S*-зависимость. Вместо индекса 1 могло бы стоять 2 или другое число, если 1 был бы уже занят для какой-либо другой зависимости.

На рис. 7.13б представлен триггер с доминирующим *R*-входом. При $S = 1$ и $R = 1$ триггер сбрасывается. Индекс 1 при *R* и на выходах отражает так называемую *R*-зависимость.

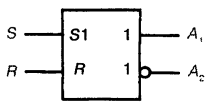


Рис. 7.13а. Триггер с доминирующим *S*-входом.

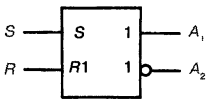


Рис. 7.13б. Триггер с доминирующим *R*-входом.

C-зависимость является так называемой управляющей зависимостью. Она применяется, в частности, при тактовом управлении триггерами. Подробнее этот вопрос рассмотрен в разд. 7.5.2.

7.2. Классификация триггеров

Можно сделать очень большое количество разнообразных триггеров. Хотя они будут обладать одним общим свойством: все они имеют два стабильных состояния. Условия, при которых триггеры переходят из одного стабильного состояния в другое и обратно, очень разнообразны. К настоящему времени появилось так много разновидностей триггеров, что возникла необходимость в их классификации.

Прежде всего, триггеры можно разбить на две большие группы. Первая группа охватывает все триггеры, не управляемые тактовыми импульсами синхронизации. Все тактируемые импульсами триггеры принадлежат ко второй группе. Что понимают под управлением тактовыми импульсами синхронизации? Рассмотрим рис. 7.14. Состояние 1 на E_1 меняет состояние выхода только при условии, что одновременно на входе T действует состояние 1. Это достигается посредством логического умножения состояний входов E_1 и T. Аналогично состояние 1 на E_2 воздействует на состояние выхода только тогда, когда одновременно на входе T действует состояние 1. T является тактовым импульсом (импульсом синхронизации, синхроимпульсом).

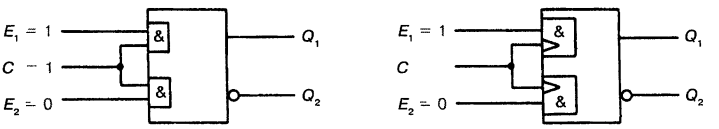


Рис. 7.14. Триггер, управляемый уровнем тактового импульса.

Триггер, согласно рис. 7.14, может быть установлен или сброшен в момент подачи сигнала синхронизации. Входы E_1 и E_2 также называют входами начальной установки триггера. Состояние 1 на E_1 подготавливает установку. Однако установка происходит только в момент подачи сигнала синхронизации T. Можно соединить большое число триггеров и приблизительно одновременно включить их совместным тактовым сигналом синхронизации. Этот способ управления называют тактовым управлением. Входы тактовых синхроимпульсов являются статическими входами.

Триггер на рис. 7.15 имеет два динамических тактовых входа, подключенных к входу C. Эти динамические входы реагируют на передний фронт сигнала синхронизации. Состояние 1 на E_1 воздействует на состояние выхода только если сигнал на C переходит из 0 в 1. То же относится к состоянию 1 на E_2. Тактовые входы, влияющие на несколько других входов, обычно рисуются между управляемыми ими входами (рис. 7.16). Знак логического умножения тогда можно не изображать. Тактируемые по фронту триггеры, можно переключать одновременно с большой точностью.

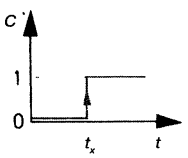
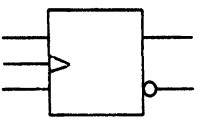


Рис. 7.15. Триггер, управляемый передним фронтом импульса (0 → 1).

Рис. 7.16. Триггер, управляемый фронтом тактового импульса (0 → 1).



Тактируемые по фронту триггеры могут строиться по одноступенчатой и двухступенчатой схемам. Недостатком одноступенчатых триггеров является возможность «проскакивания» сигналов, например логических единиц.

Рассмотрим рис. 7.17. На E_1 действует логическая 1. Если поступает сигнал синхронизации согласно рис. 7.18, то триггер I включается в момент времени t_1. Спустя несколько наносекунд на выходе A появляется состояние 1, например в момент времени t_2. Возрастание тактирующего сигнала, однако, еще не закончено. Триггер II также устанавливается ($B = 1$). Это не

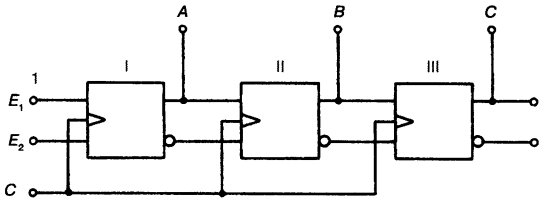


Рис. 7.17. Последовательно включенные одноступенчатые триггеры.

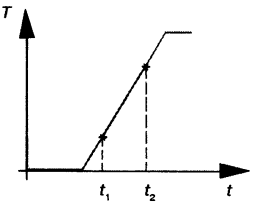


Рис. 7.18. Синхронизирующий тактовый сигнал с относительно плавным передним фронтом.

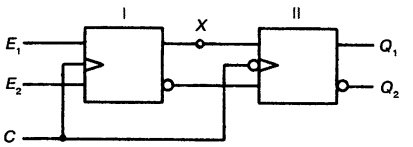


Рис. 7.19. Схема двухступенчатого триггера.

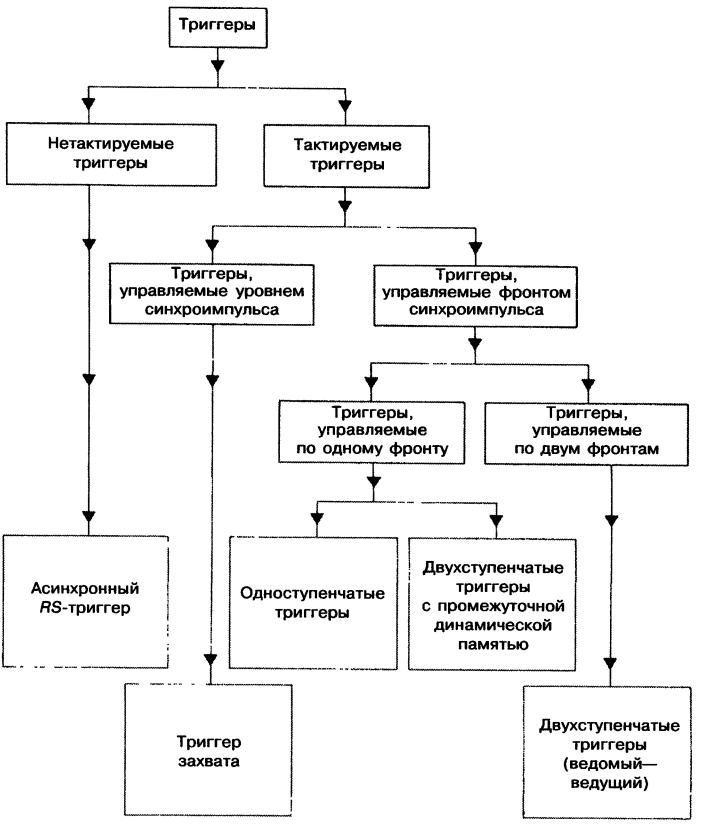


Рис. 7.20. Обзор типов триггеров.

должно происходить. Чтобы предотвратить такое проскальзывание, нужно использовать для одноступенчатых триггеров сигналы синхронизации с очень крутыми фронтами импульса.

Двухступенчатые триггеры состоят в принципе из двух ячеек памяти. Существуют двухступенчатые триггеры, которые состоят из собственно триггера и динамической промежуточной ячейки памяти. Они управляются только одним фронтом тактового синхроимпульса и принадлежат к классу триггеров, управляемых по одному фронту.

Другая группа двухступенчатых триггеров состоит из двух последовательно включенных триггеров (рис. 7.19). Триггер *I* имеет тактовый вход, который реагирует на передний фронт импульса сигнала синхронизации. Вход триггера *II* реагирует на задний фронт импульса сигнала синхронизации. Триггеры этого типа принадлежат к классу триггеров, управляемых по обоим фронтам. Они реализуются по схеме «ведущий-ведомый».

Чаще всего на практике применяют одноступенчатые и двухступенчатые триггеры. На рис. 7.20 показан схематический обзор различных типов триггеров.

7.3. Нетактируемые триггеры

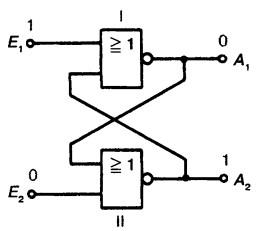
7.3.1. Триггер на элементах ИЛИ-НЕ

Простой нетактируемый триггер может быть построен на двух логических элементах ИЛИ-НЕ (рис. 7.21). Выход элемента ИЛИ-НЕ находится в нулевом состоянии, если по меньшей мере на одном из входов присутствует 1 (рис. 7.22). Если к E_1 прикладывается 1, то A_1 переходит в 0. На входе E_2 должен быть 0. Тогда на обоих входах элемента ИЛИ-НЕ *II* присутствует 0. Выход A_2 перейдет в 1 состояние (вар. 2 таблицы истинности на рис. 7.21).

В случае 3 таблицы истинности ($E_1 = 0$, $E_2 = 1$) на A_2 установится 0, а A_1 переходит в 0.

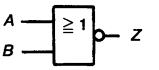
Если на обоих входах действует 0, то состояние выходов триггера остается неизменным. Можно перейти от варианта 2 к варианту 1 или от варианта 3 к варианту 1. Состояния сигналов варианта 2 изображены серым на рис. 7.21. Если E_1 устанавливается в 0, то выходы не меняют своих состояний. Такой режим называется «режим хранения».

Если на обоих входах действует 1, то оба выхода должны перейти в 0. Теперь выходы не имеют противоположных состояний. Режим, соответ-



Вар.	E_2	E_1	A_1	A_2	
1	0	0	*X*	*X*	хранение
2	0	1	0	1	
3	1	0	1	0	
4	1	1	0	0	запрещенная комбинация

Рис. 7.21. Триггер на двух элементах ИЛИ-НЕ с таблицей истинности.



Вар.	B	A	$A \vee B$	$Z = \overline{A \vee B}$
1	0	0	0	1
2	0	1	1	0
3	1	0	1	0
4	1	1	1	0

Рис. 7.22. Таблица истинности элемента ИЛИ-НЕ.

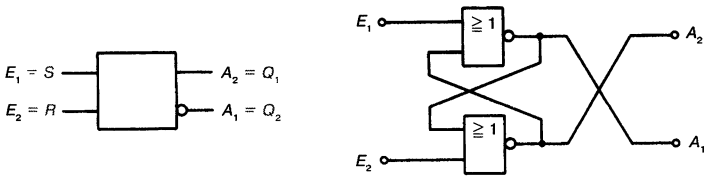


Рис. 7.23. *RS*-триггер на двух элементах ИЛИ-НЕ.

ствующий сигналам $E_1 = 1$, $E_2 = 1$ является так называемым запрещенным режимом. Он никогда не используется.

Согласно правилам для триггеров 1 на E_1 должна устанавливать в 1 противоположный выход. То есть 1 на E_2 в данном случае должна установить в 1 нижний выход. Это достигается путем перестановки выходов согласно рис. 7.23. Так как A_2 относится к верхнему, а A_1 — к нижнему выходам, то выходы переставляются местами:

$$A_2 = Q_1; A_1 = Q_2.$$

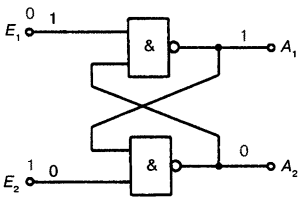
Вход E_1 является входом установки. Он обозначается символом *S*. Вход *R* является сбросом. Он обозначается символом *R*. Триггер на рис. 7.23 называется *RS*-триггер.

7.3.2. Триггер на элементах И-НЕ

Если соединить два элемента И-НЕ последовательно, как это изображено на рис. 7.24, то тоже получится триггер. Посмотрим, как он работает. Для удобства на рис. 7.25 представлена таблица истинности элемента И-НЕ. Если по меньшей мере на одном входе элемента И-НЕ будет нулевой сигнал, то на выходе возникнет логическая 1.

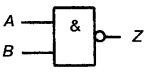
При $E_1 = 0$ и $E_2 = 1$ на выходе триггера установятся $A_1 = 1$ и $A_2 = 0$. Это режим установки. При $E_1 = 1$ и $E_2 = 0$ на выходе триггера установятся $A_1 = 0$ и $A_2 = 1$. Это режим сброса. Если на оба входа подана 1, то состояние выходов останется таким же, как и предыдущее, то есть не изменится. Это режим хранения.

Режима, при котором оба входа E_1 и E_2 равны нулю, следует избегать. В этом режиме оба выхода должны установиться в 1. Триггер из двух элемен-



Вар.	E_2	E_1	A_1	A_2	
1	0	0	1	1	запрещенная комбинация
2	0	1	0	1	сброс
3	1	0	1	0	установка
4	1	1	*X*	*X*	хранение

Рис. 7.24. Триггер на двух элементах И-НЕ с таблицей истинности.



Вар	B	A	$A \wedge B$	$Z = \overline{A \wedge B}$
1	0	0	0	1
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

Рис. 7.25. Таблица истинности элемента И-НЕ.

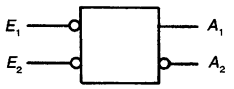
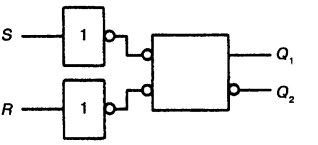


Рис. 7.26. Условное обозначение триггера на двух элементах И-НЕ.



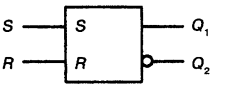


Рис. 7.27. *RS*-триггер.

Вар.	*R*	*S*	Q_{1m}	Q_{2m}	
1	0	0	$Q_1(m-1)$	$Q_2(m-1)$	хранение
2	0	1	1	0	установка
3	1	0	0	1	сброс
4	1	1	1	1	запрещенная комбинация

Рис. 7.28. Таблица истинности *RS*-триггера (рис. 7.27).

тов И-НЕ переключается 0-сигналами. Сказанное продемонстрировано условным обозначением триггера на рис. 7.26. Если перед каждым входом включить элемент НЕ, получится *RS*-триггер (рис. 7.27). Соответствующая таблица истинности изображена на рис. 7.28. Она расширена по сравнению с ранее представленными таблицами истинности. В первом варианте — режиме хранения — добавились $Q_{1(m-1)}$ и $Q_{2(m-1)}$. Индекс $m-1$ обозначает предшествующее состояние выходов. Если $Q_{1m} = Q_{1(m-1)}$ и $Q_{2m} = Q_{2(m-1)}$, значит, состояния выходов в текущий момент времени (Q_{1m}, Q_{2m}) равны предшествующим состояниям ($Q_{1(m-1)}$, $Q_{2(m-1)}$). Так однозначно обозначается режим хранения.

Триггеры на рис. 7.21 и рис. 7.24 являются самыми простыми триггерами. Они поэтому называются еще **базовыми триггерами**.

7.4. Тактируемые триггеры

Состояние нетактируемых триггеров меняется через несколько наносекунд после изменения состояния их входов. Это нежелательно во многих случаях. Чтобы изменять состояния выходов по особой команде, разработали **тактируемые триггеры**, управляемые тактовым синхронизирующим импульсом. Эти триггеры называются также триггерами захвата, так как они преимущественно используются для сбора информации.

7.4.1. *RS*-триггеры

Если к каждому входу *RS*-триггера подключить элемент И, то получится триггер, управляемый тактовым синхронизирующим импульсом (рис. 7.29). Триггер реагирует на сигнал $E_1 = 1$ только тогда, когда на управляющем входе T также действует сигнал 1. При $E_1 = 1$ триггер готов перейти в режим установки. Однако установка происходит только при подаче тактового сигнала T. Аналогичный подход имеет место для режима сброса триггера.

Условные обозначения тактируемых *RS*-триггеров указаны на рис. 7.30. Верхнее условное обозначение используется чаще. И-элементы схемы на рис. 7.29 интегрированы в прямоугольник триггера.

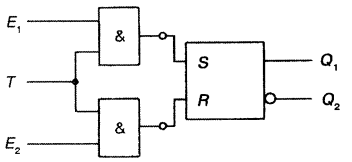
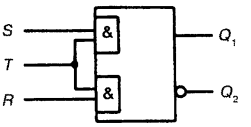


Рис. 7.29. Тактируемый *RS*-триггер.



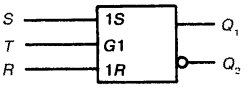


Рис. 7.30. Условные обозначения тактируемого *RS*-триггера.

Нижнее условное обозначение образовывается с помощью записи зависимости (см. раздел 7.1.1). Буква *G* обозначает логическое умножение И. Индекс 1 указывает, какие входы связаны друг с другом операцией логического умножения И.

Тактируемый *RS*-триггер (рис. 7.29) может строиться на четырех И-НЕ элементах. *RS*-триггер построен согласно рис. 7.24 на двух элементах И-НЕ с предварительно подключенными инверторами НЕ (рис. 7.27). Оба элемента НЕ и оба элемента И для возможности тактового управления объединены в элементы И-НЕ (рис. 7.31).

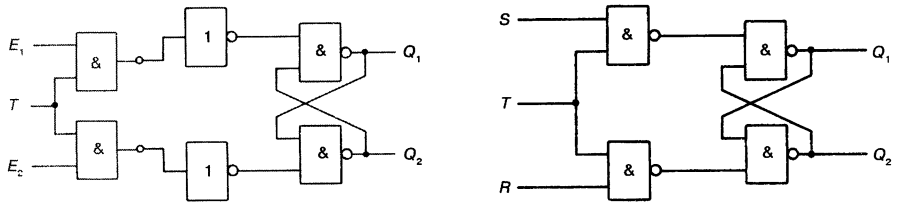


Рис. 7.31. Схема *RS*-триггера на четырех И-НЕ элементах.

Вар.	T	R	S	Q_1	Q_2	
1	0	0	0			состояние триггера не изменяется, хранение
2	0	0	1			
3	0	1	0			
4	0	1	1			
5	1	0	0			
6	1	0	1	1	0	установка
7	1	1	0	0	1	сброс
8	1	1	1	=	=	запрещенная комбинация

Рис. 7.32. Возможная таблица истинности тактируемого *RS*-триггера.

Возможная таблица истинности тактируемых *RS*-триггеров изображена на рис. 7.32. В случаях от 1 до 5 состояние триггера не изменяется. Это режимы хранения. В вариантах с 1 по 4 сигнал синхронизации 0. Поэтому состояние триггера не меняется. В случае 5 $S = 0$ и $R = 0$. Несмотря на то, что тактовый сигнал равен 1, триггер не может ни установиться, ни сброситься.

Вариант 6 является установкой, вариант 7 — сбросом. Вариант 8 запрещен и не должен встречаться.

Таблица истинности строится чаще всего без сигнала синхронизации. Вместо него вводят два временных состояния. Время t_n является временем после n-го синхронизирующего импульса. Время t_{n+1} является временем после следующего синхронизирующего импульса. Если рассматривать определенный синхронизирующий импульс, то можно сказать:

> t_n является моментом времени перед определенным синхронизирующим импульсом, t_{n+1} является моментом времени после определенного синхронизирующего импульса.

Таблица истинности разделяется на области t_n и t_{n+1} (рис. 7.33). Столбец для Q_2 можно не заполнять, так как Q_2 всегда имеет противоположное состояние относительно Q_1. Такая форма таблицы истинности является общепринятой.

Для вычислений в алгебре логики необходима таблица истинности, дающая представление о фактическом состоянии Q_{1n}. Из таблицы истинности на рис. 7.33 мы видим только, что в первом варианте состояние выхода Q_1 остается неизменным. Но каким оно было? Оно могло быть 1 или 0. В подробной таблице истинности Q_{1n} указывается в виде переменной величины (рис. 7.34). Индекс n при Q_{1n} может не указываться, так как t_n показывает принадлежность Q_1 к n-му моменту времени.

	t_n		t_{n+1}
Вар.	R	S	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	=

Рис. 7.33. Типовая таблица истинности тактируемого *RS*-триггера.

	t_n			t_{n+1}	
Вар.	R	S	$Q_{1(n)}$	Q_1	
1	0	0	0	0	состояние триггера не изменяется, хранение
2	0	0	1	1	
3	0	1	0	1	установка
4	0	1	1	1	
5	1	0	0	0	сброс
6	1	0	1	0	
7	1	1	0	=	запретная комбинация
8	1	1	1	=	

Рис. 7.34. Полная таблица истинности тактируемого *RS*-триггера.

В случае 1 Q_1 имел перед рассматриваемым тактом состояние 0. После такта состояние не изменилось. В случае 2 Q_1 имел перед рассматриваемым тактом состояние 1. После такта состояние выхода также не изменилось. Случаи 1 и 2 являются примерами режима хранения.

В случае 3 Q_1 имел перед рассматриваемым тактом состояние 0. После такта состояние изменилось на 1. В случае 4 Q_1 был до такта равен 1 и после такта остался равен 1. Случаи 3 и 4 являются примерами режима установки. Какое бы состояние ни имел Q_1 до такта, после такта синхронизации он равен всегда 1.

Случаи 5 и 6 являются примерами режима сброса. Если Q_1 имел состояние 0, то он сохраняет его и после такта. Если Q_1 имел состояние 1, то он сбрасывается на 0. Какое бы состояние ни имел Q_1 до такта, после такта синхронизации он равен 0.

Запрещенные комбинации 7 и 8 мы не будем рассматривать. Они недопустимы в этой разновидности триггеров, так как возникает неопределенность, в каком состоянии окажутся выходы триггера.

7.4.2. *RS*-триггеры с доминирующим *R*-входом

Наличие запрещенных комбинаций для тактируемых *RS*-триггеров вызвало идею построения триггера, который принудительно устанавливает Q_1 в 0 при $S = 1$ и $R = 1$, при подаче синхроимпульса. Это становится возможным благодаря особенному подключению входов.

На рис. 7.35 показано такое соединение входов. При $S = 1$ и $R = 1$, 1-сигнал не может воздействовать на триггер, так как на выходе элемента НЕ действует 0. И-элемент запирается. 1-сигнал на R разрешает сброс. Ре-

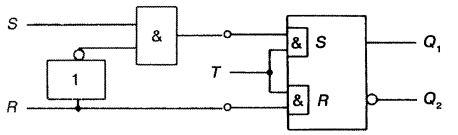


Рис. 7.35. Тактируемый *RS*-триггер с доминирующим *R*-входом.

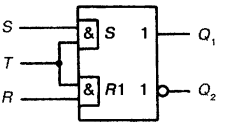


Рис. 7.36. Условное обозначение тактируемого *RS*-триггера с доминирующим *R*-входом.

	t_n		t_{n+1}
Вар.	*R*	*S*	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	0

Рис. 7.37. Таблица истинности тактируемого *RS*-триггера с доминирующим *R*-входом.

жим установки при $S = 1$ и $R = 0$ остается возможен, так как теперь на выходе элемента НЕ действует 1 и И-элемент имеет на выходе 1. Такой триггер называется ***RS*-триггером с доминирующим *R*-входом**, или ***R*-триггером**. Правило его работы следует из условного обозначения на рис. 7.36 (см. также разд. 7.1). Оно гласит: если оба входа триггера *S*, *R* и вход *T* имеют уровень 1, то при сигнале синхронизации 1 Q_1 устанавливается на 0, а Q_2 — на 1.

Таблица истинности тактируемого *RS*-триггера с доминирующим *R*-входом представлена на рис. 7.37. Разумеется, существует также тактируемый *RS*-триггер с доминирующим *S*-входом (см. контрольный тест).

7.4.3. *E*-триггер

Реже используемой разновидностью триггеров является так называемый тактируемый *E*-триггер. Таблица истинности для этого триггера приведена на рис. 7.38. При $E_1 = 1$ и $E_2 = 1$ *E*-триггер переходит в режим хранения. *E*-триггер может быть образован из *RS*-триггера путем введения дополнительных элементов на входе. В результате при $E_1 = 1$ и $E_2 = 1$ на входы *S* и *R* должны поступать сигналы 0. Однако режимы установки (вариант 2) и сброса (вариант 3) не должны быть изменены.

RS-триггер с необходимыми дополнительными элементами представлен на рис. 7.39. При $E_1 = 1$ и $E_2 = 1$ оба И-элемента запираются, так как на выходах НЕ-элементов действует 0. При $E_1 = 1$ и $E_2 = 0$ на *S*-вход поступает сигнал 1. Условия для перехода триггера в режим установки соблюдены. При $E_1 = 0$ и $E_2 = 1$ на *R*-вход поступает сигнал 1. Условия для перехода триггера в режим сброса также соблюдены.

	t_n		t_{n+1}	
Вар.	E_2	E_1	Q_1	
1	0	0	Q_{1n}	хранение
2	0	1	1	установка
3	1	0	0	сброс
4	1	1	Q_{1n}	хранение

Рис. 7.38. Таблица истинности тактируемого *E*-триггера.

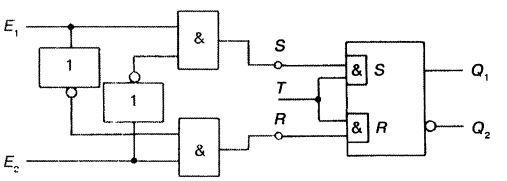


Рис. 7.39. Образование тактируемого *E*-триггера введением дополнительных элементов на входе.

7.4.4. D-триггер

D-триггер применяется чаще, чем *E*-триггер. Тактируемый *D*-триггер также может быть построен на базе *RS*-триггера. Подаваемый на *S*-вход сигнал подводится через элемент НЕ к *R*-входу (рис. 7.40). *R*-вход больше не управляется внешними сигналами.

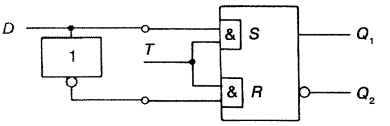


Рис. 7.40. Образование тактируемого *D*-триггера введением дополнительных элементов на входе.

	t_n	t_{n+1}
Вар.	*D*	Q_1
1	0	0
2	1	1

Рис. 7.41. Таблица истинности тактируемого *D*-триггера.

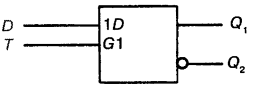


Рис. 7.42. Условное графическое обозначение тактируемого *D*-триггера.

D-триггер называет также **триггером задержки**, так как в нем задерживается прохождение входного сигнала на выход до появления тактирующего синхроимпульса. 1-сигнал на *D*-входе устанавливает Q_1 в 1. 0-сигнал на *D*-входе устанавливает Q_1 в 0. Таблица истинности *D*-триггера приведена на рис. 7.41. Так как в наличии имеется только один вход, таблица истинности содержит только две комбинации.

Условное обозначение тактируемого *D*-триггера показано на рис. 7.42. Буква *G* указывает на логическое умножение И. Цифрой 1 отмечены связанные логическим умножением входы.

7.4.5. Таблицы данных

Производители триггеров издают подробные таблицы данных, в которых приведены все необходимые параметры микросхемы. В рамках этой книги в качестве примеров представлены несколько таблиц данных.

Среди тактируемых триггеров в настоящее время в основном применяются *D*-триггеры. Микросхема FLJ 151-7475 является ТТЛ-схемой (см. разд. 6.6). Она содержит четыре *D*-триггера (рис. 7.43). Цоколевка, структура и таблица истинности приведены на рис. 7.44.

Таблица данных (рис. 7.43) содержит типовые статические параметры ТТЛ-схем, рассмотренные в подразд. 6.6.2.2. По времени переключения можно определить быстродействие схемы (разд. 6.4.3). Напряжение питания 5 В.

Под t_V понимают так называемое время подготовки к работе на входе *D*. Это время, в течение которого сигнал должен находиться на входе перед прибытием такта синхронизации. Оно составляет 20 нс. Если сигнал присутствует на входе *D* менее 20 нс, то запись в память не гарантирована.

Синхронизирующий импульс должен также длиться по меньшей мере 20 нс. Это время обозначается t_{PT}. Времена переключения относительно коротки. Триггеры работают быстро.

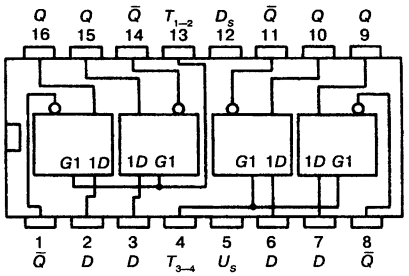
4 *D*-триггера FLJ 151-7475 FLG 155-8475

Триггеры FLJ 151 или FLJ 155 имеют два стабильных состояния, управляемых синхроимпульсом. Пока действует синхронизирующий импульс, информация с *D*-входа переносится на *Q*-выход. Она сохраняется на выходе и после прекращения подачи синхронизирующего импульса. Информация стирается, если синхронизирующий импульс повторяется.

Статические параметры в температурных зонах 1 и 5		Условия испытаний		Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S				4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В		36	2,0			В
Входное напряжение *L*-уровня	U_{IL}			37			0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА					1,5	В
Выходное напряжение *H*-уровня	U_{QH}	$-I_{QH}$ = 400 мкА	U_S = 4,74 В	36,37	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	I_{QL} = 16 мА	U_S = 4,74 В	36,37		0,2	0,4	В
Статическая помехоустойчивость	U_{SS}				0,4	1,0		В
Входной ток на канал	I_I	U_I = 5,5 В	U_S = 5,25 В	38			1	мА
H-входной ток на *D* на *T*	 I_{IH} I_I	 U_{IH} = 2,4 В U_I = 5,5 В	 U_S = 5,25 В U_S = 5,25 В	 38 38			 80 160	 мкА мкА
L -входной ток на *D* на *T*	 $-I_{IL}$ $-I_{IL}$	 U_{IL} = 0,4 В U_{IL} = 0,4 В	 U_S = 5,25 В U_S = 5,25 В	 38 38			 3,2 6,4	 мА мА
Ток короткого замыкания на канал выхода	$-I_Q$	U_S = 5,25 В		39	18		57	мА
Ток питания	I_S	U_S = 5,25 В		40		32	53	мА
Время переключения, при U_S = 5 В, T_U = 25 °C								
Логические параметры								
Длительность синхроимпульса	t_{pT}				20			нс
Время подготовки на *D*	t_v				20			нс

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Время прохождения сигнала (быстродействие) от *D* к *Q*	t_{PHL}	C_L = 15 пФ, R_L = 400 Ом			14	25	нс
	t_{PLH}				16	30	нс
от *D* к $\bar{Q}$	t_{PHL}				7	15	нс
	t_{PLH}				24	40	нс
от *I* к *Q*	t_{PHL}				7	15	нс
	t_{PLH}				16	30	нс
от *T* к $\bar{Q}$	t_{PHL}				7	15	нс
	t_{PLH}				16	30	нс
Коэффициент разветвления по выходу на канал	F_Q					10	
Входной нагрузочный коэффициент на *D*	F_I					2	
Входной нагрузочный коэффициент на *T*	F_I					4	

Рис. 7.43. Таблица данных микросхемы FLJ 151-7475 (Siemens).



Цоколевка корпуса
вид сверху

D — вход
Q, $\bar{Q}$ — выходы
T — вход тактирующего синхроимпульса

t_n	t_{n+1}	
D	*Q*	$\bar{Q}$
H	*H*	*L*
L	*L*	*H*

t_n — момент времени до синхроимпульса
t_{n+1} — момент времени после синхроимпульса

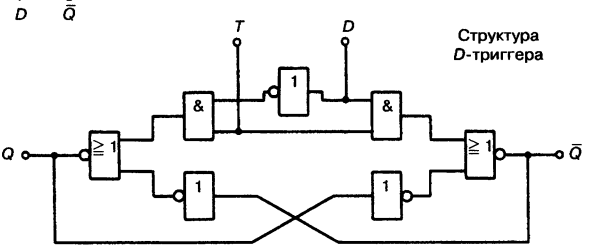


Рис. 7.44. Приложение к таблице данных для интегральной микросхемы FLJ 151-7475 (Siemens).

Содержит 4 *D*-триггера

Модуль FZJ 131/135 содержит четыре тактируемых *D*-триггера. Информация поступает с входа *D* на выход *Q* при *T* = *H*. *D*-вход заперт при *T* = *L*.

Типовое применение: 4-битовый промежуточный буфер памяти.

Статические параметры при 12 В в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S		11,4	12	13,5	В
Входное напряжение *H*-уровня	U_{IH}	$U_S = U_{SB}$	7,5			В
Входное напряжение *L*-уровня	U_{IL}	$U_S = U_{SB}$ и U_{SA}			4,5	В
Выходное напряжение *H*-уровня	U_{QH}	$U_S = U_{SB}$, $-I_{QH}$ = 0,1 мА U_{IH} = 7,5 В	10,0	11,3		В
Выходное напряжение *L*-уровня	U_{QL}	$U_S = U_{SB}$, I_{QL} = 15 мА U_{ID} = 4,5 В, U_{IT} = 7,5 В		0,9	1,7	В
Статическая помехоустойчивость	U_{ssH}		2,5	5		В
	U_{ssL}		2,8	5		В
Входной ток на канал *H*-входной ток на *D*	I_H	$U_I = U_{IHA}$, $U_S = U_{SA}$			1	мкА
L-входной ток на *D*	$-I_{IL}$	$U_S = U_{SA}$, U_{IL} = 1,7 В			3	мА
L-входной ток на *T*	$-I_{IL}$	$U_S = U_{SA}$, U_{IL} = 1,7 В			6	мА
Ток короткого замыкания на канал выхода	$-I_Q$	$U_S = U_{SA}$, $U_I = U_Q$ = 0 В	9	15	25	мА
Ток питания	I_S	$U_S = U_{SA}$, U_I = 0 В		22	32	мА
Энергопотребление	P	$U_S = U_{SA}$, U_I = 0 В		264	432	мВт
Время переключения, при U_S = 5 В, T_U = 25 °C						
Максимальная тактовая частота	f_Z	Скважность 1 : 1	0,5			мГц
Длительность синхроимпульса	t_{pT}	Через землю 4,5 В	0,5			мкс
Время подготовки на *D*						
H-сигнал	t_S		300			нс
L-сигнал	t_S		500			нс
Время промежуточного хранения на *D*						
H-сигнал	t_H		150			нс
L-сигнал	t_H		50			нс

Статические параметры при 12 В в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Время прохождения сигнала (быстродействие)						
от *D* к *Q*	t_{PLH}	C_L = 10 пФ при 4,5 В через землю	90	175	310	нс
	t_{PHL}		30	70	150	нс
от *D* к $\overline{Q}$	t_{PLH}		30	70	150	нс
	t_{PHL}		70	130	290	нс
от *T* к *Q*	t_{PLH}		90	160	310	нс
	t_{PHL}		70	120	210	нс
от *T* к $\overline{Q}$	t_{PLH}		90	150	310	нс
	t_{PHL}		70	120	210	нс
Время нарастания сигнала	t_{TLH}	C_L = 10 пФ	50	90	170	нс
	t_{THL}		15	35	60	нс

Рис. 7.45. Таблица данных микросхемы FZJ 131 (Siemens).

Микросхема FZJ 131 содержит также четыре тактируемых *D*-триггера. Эта микросхема является ДТЛ-схемой подсемейства «медленной помехоустойчивой логики». Из таблицы данных на рис. 7.45 можно увидеть, что время переключения значительно дольше, чем в схеме FLJ 151-7475. Статическая помехоустойчивость очень большая. Необходимое напряжение питания 12 В. Структура *D*-триггеров (рис. 7.46) является типичной для ДТЛ-схем.

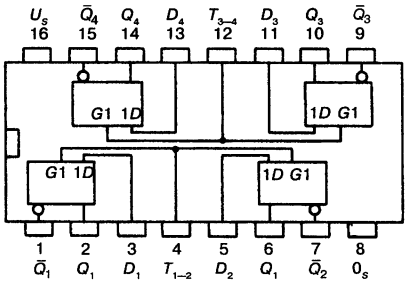
7.5. Триггеры, управляемые по фронту синхроимпульса

Благодаря управлению по фронту (фронтальному управлению) достигают очень точного одновременного переключения нескольких триггеров. Даже при больших допусках при изготовлении почти не происходит отклонения от заданных моментов переключения.

При фронтальном управлении триггеры переключаются синхронно.

Еще одним преимуществом фронтального управления является пониженная подверженность помехам. Помеха на входах может послужить причиной сбоя только в том случае, если она действует в течение очень короткого периода времени переключения триггера. До и после этого периода паразитные сигналы не оказывают воздействия на работу триггера.

При фронтальном управлении триггеры обладают большей помехоустойчивостью.

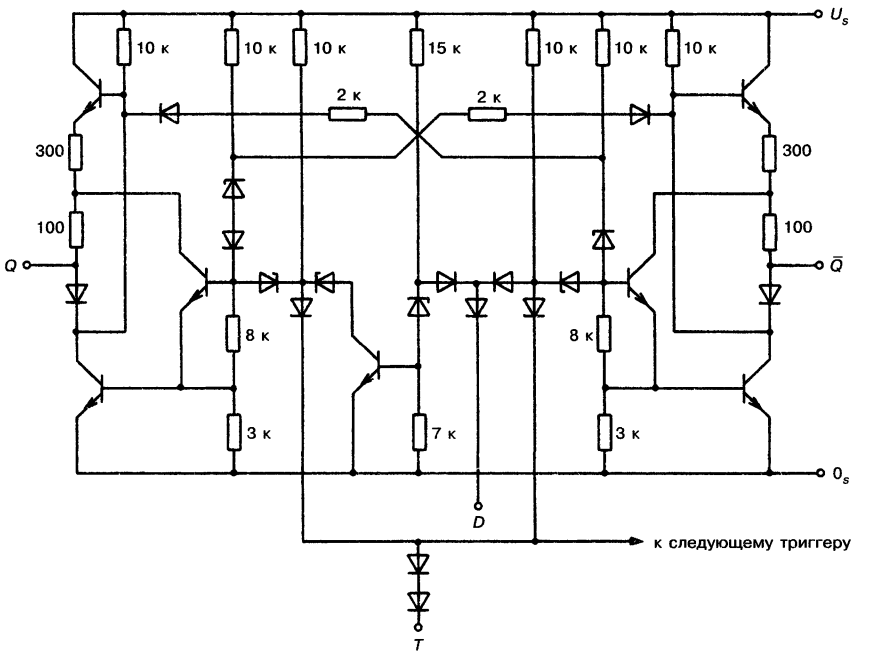


Цоколевка корпуса
вид сверху

D — вход
Q, $\bar{Q}$ — выходы
T — вход тактирующего синхроимпульса

Рис. 7.46. Приложение к таблице данных для интегральной микросхемы FZJ 131 (Siemens).

Принципиальная схема (один триггер)



7.5.1. Импульсные элементы

Для фронтального управления необходимы импульсные элементы. Они имеют статический и динамический входы и работают как элементы И. Импульсный элемент на рис. 7.47 выдает на выходе отрицательный импульс только тогда, когда $A = 1$, а T-сигнал падает с 1 в 0 (обратный фронт).

Возможная структура импульсного элемента (рис. 7.47) представлена на рис. 7 48. На входе A должен действовать сигнал 1. В точке X тогда действует сигнал 0, что соответствует потенциалу 0 В. На T-входе также должен быть сигнал 1 ($\triangleq$ +5 В). Конденсатор C заряжается до 5 В. Если сигнал T становится снова равным 0 В, то отрицательный полюс конденсатора в первый момент времени будет иметь потенциал −5 В. Диод откроется. К выходу Z после уменьшения на величину прямого напряжения диода приложено напряжение −4,3 В. Это напряжение падает после разряда конденсатора до 0.

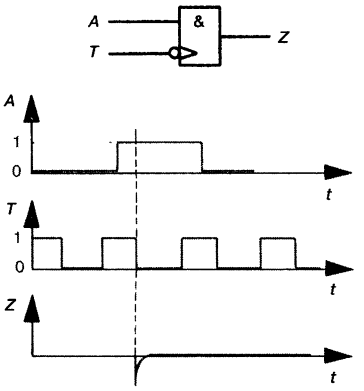


Рис. 7.47. Условное обозначение и временная диаграмма импульсного элемента для варианта с отрицательным выходным импульсом.

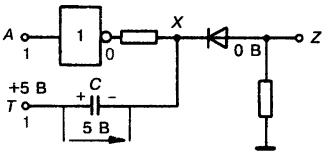


Рис. 7.48. Структура импульсного элемента.

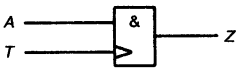
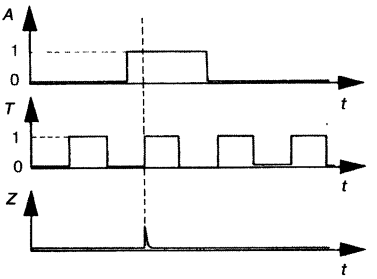


Рис. 7.49. Условное обозначение и временная диаграмма импульсного элемента для варианта с положительным выходным импульсом.



Импульсные элементы второго вида выдают положительные выходные импульсы. Импульсный элемент на рис. 7.49 выдает положительный импульс только тогда, когда на входе *A* действует сигнал 1 и синхроимпульс *T* переходит с 0 на 1.

7.5.2. *RS*-триггеры, управляемые по одному фронту

Нетактируемый *RS*-триггер (см. рис. 7.27) превращается с помощью двух И-элементов, подключенных до триггера в соответствии с рис. 7.29, в тактируемый *RS*-триггер. Если оба этих элемента заменить импульсными элементами, то получают тактируемый фронтально управляемый *RS*-триггер (рис. 7.50). Триггер переключается при переходе сигнала синхронизации с 0 на 1, т. е. с передним фронтом импульса. Условное обозначение этого триггера приведено на рис. 7.51. Для обозначения входа синхронизации (тактирующего входа) используется буква *C* (от англ. clock — часы). *C*-вход воздействует на оба входа триггера и поэтому изображается в середине.

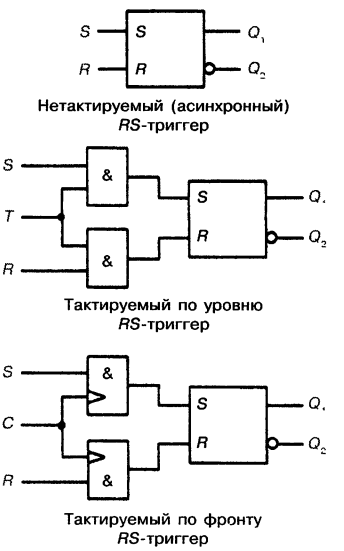


Рис. 7.50. Образование тактируемого по фронту *RS*-триггера.

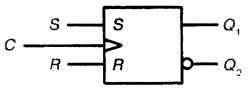


Рис. 7.51. Условное обозначение управляемого по фронту *RS*-триггера, переключающегося передним фронтом синхроимпульса.

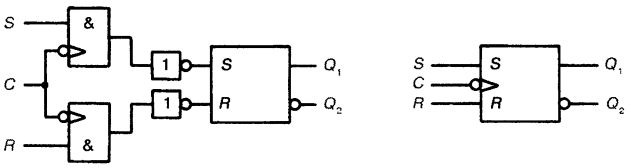


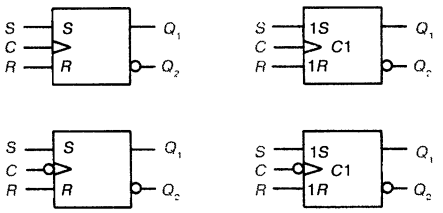
Рис. 7.52. Схема и условное обозначение тактируемого управляемого по фронту *RS*-триггера, переключающегося задним фронтом синхроимпульса.

Если используют импульсные элементы другого вида, то получают *RS*-триггер, который переключается задним фронтом импульса (рис. 7.52). Требуются два дополнительных элемента НЕ или еще можно применить триггер на элементах И-НЕ, показанный на рис. 7.26.

Схемы, применяемые в области интегральных микросхем, являются в основном более сложными. Разработчики стремятся получить большую помехоустойчивость при возможно более высокой рабочей скорости. Дополнительные усложнения в схеме лишь немного повышают издержки при производстве микросхем. Для потребителя внутренняя структура схемы не имеет значения. Важны лишь хорошие параметры и качество предлагаемых триггеров.

Рассмотренный тактируемый по фронту *RS*-триггер называется также **управляемым по одному фронту *RS*-триггером**. Такое название помогает отличать их от рассматриваемых далее управляемых по двум фронтам *RS*-триггеров.

Таблица истинности для управляемых по одному фронту *RS*-триггеров такая же, как для тактируемых по уровню сигнала *RS*-триггеров, так как в таблице истинности не приводится способ тактового управления. Таблица истинности (рис. 7.53) действительна как для *RS*-триггеров, управляемых передним фронтом импульса, так и для *RS*-триггеров, управляемых задним фронтом импульса. Можно использовать условные обозначения с указанием зависимости или без указания (рис. 7.53).



	t_n		t_{n+1}
Вар.	*R*	*S*	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	=

Рис. 7.53. Условное обозначение тактируемого управляемого по одному фронту *RS*-триггера и его таблица истинности.

Чаще всего не требуется обозначать основное положение *RS*-триггеров. Управляемые по одному фронту *RS*-триггеры производятся преимущественно с фиксированным основным положением. Это $Q_1 = 0$, $Q_2 = 1$. Если нужно обозначить основное положение, то способ его условного обозначения обсуждался в разделе 7.1 (рис. 7.54).

Для многих целей нужны *RS*-триггеры, которые могут быть дополнительно установлены и сброшены независимо от тактирующих сигналов. Для этого требуются дополнительные входы. Триггер на рис. 7.55 имеет независимый от синхросигнала вход сигнала установки в единицу S^* и независи-

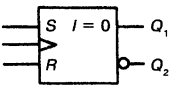


Рис. 7.54. Условное обозначение управляемого по одному фронту *RS*-триггера с обозначением основного положения $Q_1 = 0$, $Q_2 = 1$.

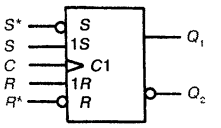


Рис. 7.55. Условное обозначение управляемого по одному фронту *RS*-триггера с независимыми от тактирующего сигнала входами установки *S** и сброса *R**.

мый от синхросигнала вход сброса *R**. Кольца отрицания перед входами указывают, что управление происходит 0-сигналами, 1-сигналы недействительны. Сигнал 0 в *R** возвращает триггер в основное положение. Тактовый сигнал не требуется. Соответственно 0-сигнал в *S** устанавливает триггер в рабочее положение ($Q_1 = 1$, $Q_2 = 0$).

Для обозначения управляемых входов требуется запись зависимости. Фронтально управляемые входы обозначены помимо *S* и *R* еще индексом, одинаковым с управляющим входом *C*. На рис. 7.55 индекс равен 1. Независимые от тактирующего сигнала входы обозначены только как *S* и *R*.

Все *RS*-триггеры имеют существенный недостаток. Следует избегать комбинации входного сигнала $S = 1$ и $R = 1$. Она приводит в отдельных схемах к неопределенности состояний на выходах и поэтому запрещена.

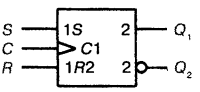


Рис. 7.56. Условное обозначение управляемого по одному фронту *RS*-триггера с доминирующим *R*-входом.

Вариант схемы *RS*-триггер с доминирующим *R*-входом уже обсуждался среди тактируемых триггеров. Такой *RS*-триггер может быть построен также с управлением по одному фронту. Соответствующее условное обозначение представлено на рис. 7.56. Для обозначения *R*-зависимости использовался индекс 2.

7.5.3. *T*-триггеры, управляемые по одному фронту

Часто нужен триггер, который меняет свое состояние при каждом такте синхроимпульса. В качестве управляющего фронта берется передний фронт тактового импульса (0 → 1). Если триггер установлен, например, в $Q_1 = 1$, то при нарастающем фронте синхроимпульса он должен переключиться в $Q_1 = 0$, при следующем переднем фронте в $Q_1 = 1$ и т. д. Такой триггер называется *T*-триггером. Он может быть построен на базе управляемого по одному фронту *RS*-триггера.

Рассмотрим *RS*-триггер на рис. 7.57. Он установлен в $Q_1 = 0$, $Q_2 = 1$ и должен переключиться при следующем переднем фронте синхроимпульса. Это возможно только в случае, если на входе *S* действует 1-сигнал. 1-сигнал

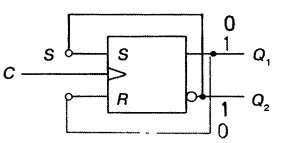


Рис. 7.57. Управляемый по одному фронту *RS*-триггер с дополнительными соединениями.

может быть взят со входа Q_2 (черная линия). При растущем фронте импульса сигнала синхронизации триггер теперь меняет свое состояние.

Теперь *RS*-триггер установлен в $Q_1 = 1$, $Q_2 = 0$ (серым на рис. 7.57). Триггер установлен и находится в рабочем положении. При следующем переднем фронте синхроимпульса триггер вернется в исходное состояние $Q_1 = 0$, $Q_2 = 1$. Для этого необходим 1-сигнал на *R*. Этот сигнал может быть взят со входа Q_1 (серая линия). Теперь триггер может вернуться в исходное состояние. При этом $Q_2 = 1$ и на *S*-входе действует 1-сигнал. При следующем переднем фронте синхроимпульса триггер снова переходит в рабочее состояние. Итак, мы имеем желаемый *T*-триггер.

Условное обозначение *T*-триггера, управляемого передним фронтом синхроимпульса, изображено на рис. 7.58. Вторая разновидность *T*-триггеров управляется задним фронтом синхроимпульса. Их условное обозначение показано на рис. 7.59. Таблица истинности для *T*-триггеров простая, так как у них имеется только один вход (рис. 7.60).

Иногда нужны триггеры, которые могут быть заперты или открыты сигналом с дополнительного входа. Такой триггер может быть образован из управляемого по одному фронту *RS*-триггера. *RS*-триггер должен иметь тогда два *S*-входа и два *R*-входа, которые связаны логическим умножением И (рис. 7.61).

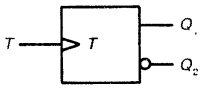


Рис. 7.58. Условное обозначение управляемого по одному фронту *T*-триггера, управляемого передним фронтом синхроимпульса.

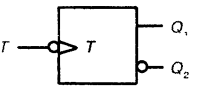


Рис. 7.59. Условное обозначение управляемого по одному фронту *T*-триггера, управляемого задним фронтом синхроимпульса.

	t_n	t_{n+1}
Вар.	Q_1	Q_1
1	0	1
2	1	0

Рис. 7.60. Таблица истинности *T*-триггеров на рис. 7.59 и 7.60.

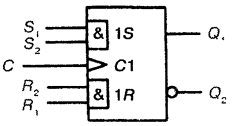


Рис. 7.61. Условное обозначение управляемого по одному фронту *T*-триггера с попарно связанными логическим умножением *S*- и *R*-входами.

S- и *R*-входы будут связаны с выходами Q_1 и Q_2, как на рис. 7.57. Свободный *S*-вход и свободный *R*-вход связываются друг с другом. Они образуют новый *T*-вход (рис. 7.62). Тактирующий вход получает обозначение *C*. Эти обозначения являются типовыми для *T*-триггера. *T*-триггер переключается теперь по *C*-сигналу (по переднему фронту *C*-синхроимпульса), если на *T* действует сигнал 1. При $T = 0$ триггер заблокирован. Состояния выходов триггера тогда не изменяются независимо от *C*-сигнала.

На рис. 7.63 показано условное обозначение такого *T*-триггера и его таблица истинности. При $T = 0$ состояние выхода Q_1 после текущего такта равно состоянию Q_1 перед текущим тактом, т. е. Q_{1n}. При $T = 1$ состояние выхода Q_1 после текущего такта противоположно состоянию Q_1 перед текущим тактом, т. е. равно $\overline{Q}_{1n}$.

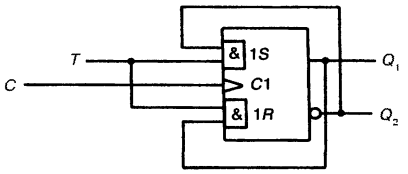
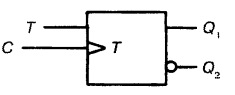


Рис. 7.62. Образование *T*-триггера с *T*- и *C*-входами из *RS*-триггера.



t_n	t_{n+1}
T	Q_1
0	Q_{1n}
1	$\bar{Q}_{1n}$

Рис. 7.63. Условное обозначение *T*-триггера с *T*- и *C*-входами и его таблица истинности.

7.5.4. *JK*-триггеры, управляемые по одному фронту

В поисках максимально универсального триггера за основу был взят управляемый по одному фронту *RS*-триггер. Универсальный триггер должен иметь режимы хранения, установки и сброса *RS*-триггера (см. таблицу истинности на рис. 7.53). В случае запрещенной комбинации 4 с $S = 1$ и $R = 1$ триггер должен переключаться как *T*-триггер.

Такой универсальный триггер мы уже нашли, он изображен на схеме (рис. 7.63). Надо только снова восстановить связь *S*-входа и *R*-входа с *T*-входом (рис. 7.64). Новые входы называются *J* и *K*.

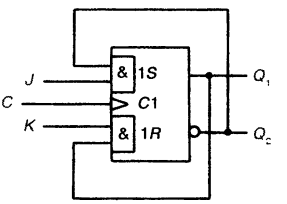
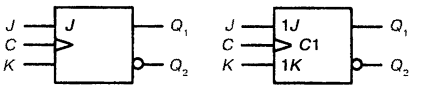


Рис. 7.64. Образование *JK*-триггера из *RS*-триггера.

При комбинации $J = 0$ и $K = 0$ триггер работает в режиме хранения. При $J = 1$ он устанавливается, если $Q_1 = 0$ и $Q_2 = 1$. Сброс происходит при $K = 1$, когда $Q_1 = 1$ и $Q_2 = 0$, все переходы, разумеется, фронтально-управляемые. При $J = 1$ и $K = 1$ триггер ведет себя как *T*-триггер.

Обозначения *J* и *K* взяты произвольно из алфавита и не являются каким-либо сокращением. Условное обозначение управляемого по фронту *JK*-триггера вместе с таблицей истинности приведены на рис. 7.65.



	t_n		t_{n+1}
Вар.	K	J	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	$\bar{Q}_{1n}$

Рис. 7.65. Условное обозначение и таблица истинности управляемого по одному фронту *JK*-триггера (управление по переднему фронту).

Конечно, существуют также *JK*-триггеры, управляемые задним фронтом импульса. Большинство *JK*-триггеров имеют несколько входов и несколько выходов, которые связаны логическим умножением И.

Также часто на триггерах используются нетактируемые входы установки и сброса. На рис. 7.66 показано условное обозначение такого *JK*-триггера. Нетактируемые входы установки и сброса переключаются 0-сигналами, поэтому они обозначаются $\bar{S}$ и $\bar{R}$.

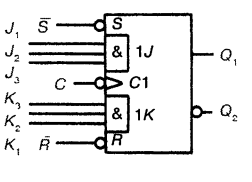


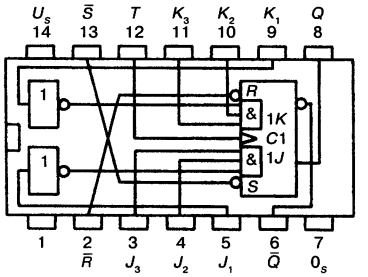
Рис. 7.66. Условное обозначение управляемого по одному фронту *JK*-триггера с 3 *J*- и 3 *K*-входами, нетактируемым входом установки и нетактируемым входом сброса.

Элемент FLJ 101/105 является управляемым по фронту

Статические параметры B в температурных зонах 1 и 5		Условия испытаний	Нижний предел B	Тип	Верхний предел A	Единица измерения
Напряжение питания	U_S		4,75	5,0	5,25	В
Входное напряжение H-уровня	U_{IH}	U_S = 4,75 В	2,0			В
Входное напряжение L-уровня	U_{IL}	U_S = 4,75 В			0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА			1,5	В
Выходное напряжение H-уровня	U_{QH}	U_S = 4,75 В, $-I_{QH}$ = 400 мкА U_{IL} = 0,8 В, U_{IH} = 2,0 В	2,4	3,4		В
Выходное напряжение L-уровня	U_{QL}	U_S = 4,75 В, I_{QL} = 16 мА U_{IL} = 0,8 В, U_{IH} = 2,0 В		0,2	0,4	В
Статическая помехоустойчивость	U_{SS}		0,4	1,0		В
Входной ток на канал	I_I	U_S =5,25 В, U_I = 5,5 В			1	мА
H-входной ток на $\overline{R}$ или $\overline{S}$	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В			80	мкА
на T, J или K	I_{IH}				40	мкА
L-входной ток на $\overline{R}$ или $\overline{S}$	$-I_{IL}$	U_S =5,25 В, U_{IL} = 0,4 В			3,2	мА
на T, J или K	$-I_{IL}$				1,6	мА
Ток короткого замыкания на канал выхода	$-I_Q$	U_S = 5,25 В	18		57	мА
Ток питания	$-I_S$	U_S = 5,25 В		13	26	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						
Время промежуточного хранения	t_H	C_L = 15 пФ R_L = 400 Ом	5			нс
Тактовая частота	f_T		15	20		мГц
Время прохождения сигнала (быстродействие) от $\overline{R}$	t_{PHL}				50	нс
или $\overline{S}$ к Q	t_{PLH}				50	нс
Время прохождения сигнала (быстродействие)	t_{PHL}		10	18	50	нс
от T к Q	t_{PLH}		10	27	50	нс
Логические параметры						
Коэффициент разветвления по выходу на канал	F_Q				10	

Рис. 7.67. Таблица данных микросхемы FLJ 101-7470 (по данным фирмы Siemens).

На рис. 7.67 приведена таблица данных микросхемы FLJ 101-7470. Эта микросхема содержит *JK*-триггер с 3 *J*- и 3 K-входами, нетактируемым входом установки и нетактируемым входом сброса. По цоколевке на рис. 7.68 видно, что входы K_1 и J_1 управляются 0-сигналами.



Цоколевка корпуса
вид сверху

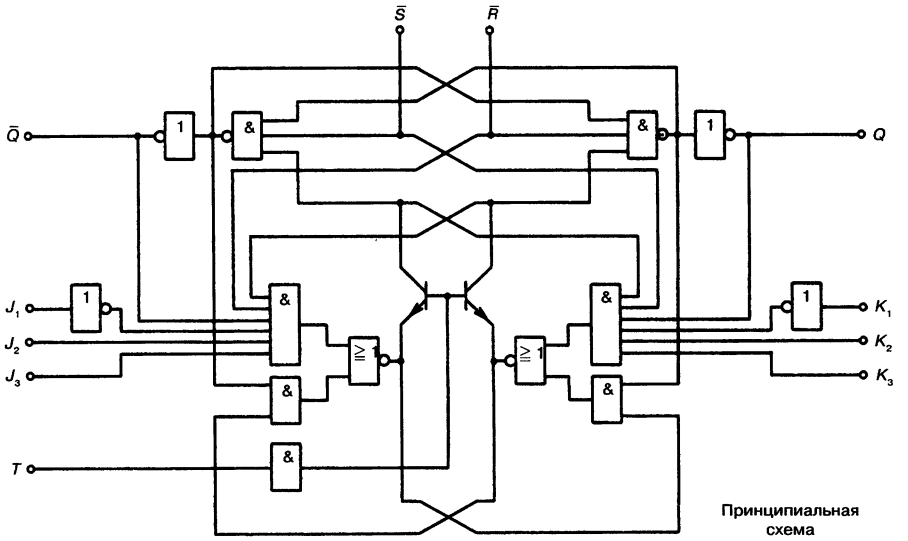
Q, $\bar{Q}$ — выходы ($Q \triangleq Q_1$, $\bar{Q} \triangleq Q_2$)
$\bar{S}$ — вход установки
$\bar{R}$ — вход сброса
T — вход тактирующего синхроимпульса
U_s — положительный полюс напряжения питания
0_s — отрицательный полюс напряжения питания, земля

Рис. 7.68. Цоколевка микросхемы FLJ 101-7470.

Микросхема FLJ 101-7470 принадлежит к ТТЛ-семейству. Это семейство микросхем рассмотрено в подразд. 6.6.2.2. В данном разделе дополнительно вводится так называемое **время промежуточного хранения**. После подачи высокого уровня синхроимпульса (типичное значение 1,5 В для ТТЛ-схем) входные сигналы должны сохранять свое состояние в течение определенного времени. Это время называется временем промежуточного хранения. По истечении времени промежуточного хранения (обычно 5 нс) изменения входного сигнала не имеют значения. Паразитные сигналы могут повлиять на триггер только в течение времени промежуточного хранения. Следовательно, чем меньше время промежуточного хранения, тем более помехоустойчивой является схема.

Как дополнение к таблице данных приводятся блок-схема и таблица уровней (рис. 7.69). В таблице уровню *L* соответствует 0-сигнал и уровню *H* 1-сигнал. Нетактируемые входы установки и сброса обозначаются $\bar{S}$ и $\bar{R}$. Это значит, что эти входы управляются 0-сигналами.

Управляемые по одному фронту *JK*-триггеры часто имеют так называемую динамическую память. Она увеличивает время распространения сигнала и предотвращает установку выходов в желаемое состояние еще во время переднего фронта импульса сигнала синхронизации или в течение времени промежуточного хранения. Это могло бы привести к ложному срабатыванию. Рассмотрим управляемый по одному фронту *JK*-триггер, который управляется прямым фронтом синхроимпульса. При $J = 1$ и $K = 1$ этот триггер будет переключаться, если синхроимпульс достигнет, например, порогового значения напряжения (+1,5 В). Если желаемые выходные состояния ($Q_1 = 1$ и $Q_2 = 0$) очень быстро установятся на выходе, то может произойти повторное срабатывание ($Q_1 = 0$ и $Q_2 = 1$). Динамическая память состоит из одной или нескольких небольших емкостей, которые перезаряжаются. В качестве этих емкостей могут служить емкости запертых *pn*-переходов. Такие триггеры правильно следует называть **триггерами с двойной памятью**.



Логические операции

t_n		t_{n+1}
J	K	Q
L	L	Q_n
L	H	L
H	L	H
H	H	$\bar{Q}_n$

$J = \bar{J}_1 \wedge J_2 \wedge J_3$
$K = \bar{K}_1 \wedge K_2 \wedge K_3$
t_n — момент времени перед синхроимпульсом
t_{n+1} — момент времени после синхроимпульса

L-потенциал на *R* устанавливает на *Q* *L*-уровень
L-потенциал на *S* устанавливает на *Q* *H*-уровень
R и *S* не зависят от *T*

Рис. 7.69. Приложение к таблице данных микросхемы FLJ 101-7470 (Siemens).

7.5.5. *D*-триггеры, управляемые по одному фронту

Управляемый по фронту *D*-триггер по структуре очень похож на тактируемый по уровню *D*-триггер. Эти разновидности триггера отличаются только управлением. Управляемые по одному фронту *D*-триггеры делятся на триггеры, управляемые прямым фронтом и обратным фронтом импульса сигнала синхронизации (рис. 7.70). Таблица истинности изображена на рис. 7.71.

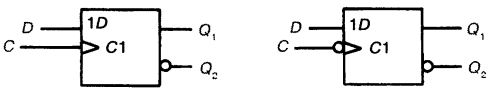


Рис. 7.70. Условное обозначение управляемого по одному фронту *D*-триггера.

	t_n	t_{n+1}
Вар.	D	Q_1
1	0	0
2	0	1

Рис. 7.71. Таблица истинности управляемого по одному фронту *D*-триггера.

После подачи управляющего фронта синхроимпульса сигнал на *D*-входе переходит на выход триггера и действует на выходе Q_1 в прямой и на выходе Q_2 в инверсной форме.

Управляемые по одному фронту *D*-триггеры используются прежде всего для сдвиговых регистров (см. разд. 12). На рис. 7.72 представлена таблица данных микросхемы FLJ 141-7474. Эта микросхема содержит два управляемых прямым фронтом *D*-триггера.

Сдвоенный *D*-триггер

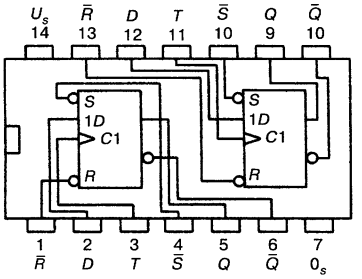
Триггеры FLJ 141 и FLJ 145 имеют нетактируемые входы установки и сброса. Передача информации со входа *D* на выход *Q* происходит во время прямого фронта синхроимпульса, сразу после открытия входного транзистора. После этого *D*-вход снова запирается.

Статические параметры в температурных зонах 1 и 5		Условия испытаний		Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S				4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В		31	2,0			В
Входное напряжение *L*-уровня	U_{IL}						0,8	
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА					1,5	
Выходное напряжение *H*-уровня	U_{QH}	$-I_{QH}$ = 400 мкА	U_S = 4,75 В	31	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	I_{QL} = 16 мА	U_S = 4,75 В	31		0,2	0,4	
Статическая помехоустойчивость	U_{SS}				0,4	1,0		
Входной ток на канал	I_I	U_I = 5,5 В	U_S = 5,25 В	32			1	мА
H-входной ток на *D*	I_{IH}	U_{IH} = 2,4 В	U_S = 5,25 В	32			40	мкА
на $\overline{S}$ или *T*	I_I	U_I = 2,4 В	U_S = 5,25 В	32			80	мкА
на $\overline{R}$	I_{IH}	U_{IH} = 2,4 В	U_S = 5,25 В	32			120	мкА
L-входной ток								
на *D* или $\overline{S}$	$-I_{IL}$	U_{IL} = 0,4 В, U_S = 5,25 В		33			1,6	мА
на $\overline{R}$ или *T*	$-I_{IL}$	U_{IL} = 0,4 В, U_S = 5,25 В		33			3,2	мА
Ток короткого замыкания на канал выхода	$-I_{QH}$	U_S = 5,25 В		34	18		57	мА
Ток питания	I_S	U_I = 5 В		32		17	30	мА
Время переключения, при U_S = 5 В, T_U = 25 °C								
Длительность синхроимпульса	t_{pT}				30			нс
Длительность импульса установки	t_{pS}				30			нс
Длительность импульса сброса	t_{pR}				30			нс
Максимальная тактовая частота	f_Z	C_L = 15 пФ R_L = 400 Ом		30a	15	25		мГц

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Минимальные							
Время подготовки сигнала	t_V	C_L = 15 пФ R_L = 400 Ом	30a		15	20	нс
Минимальное время промежуточного хранения	t_H						
Время прохождения сигнала (быстродействие)							
от *T* к *Q*	t_{PHL}	C_L = 15 пФ R_L = 400 Ом	30	10	20	40	нс
	t_{PLH}		30a	10	14	25	нс
от $\overline{R}$ или *T* к *Q*	t_{PHL}		30			40	нс
	t_{PLH}		30a			25	нс
Логические параметры							
Коэффициент разветвления по выходу на канал	F_Q					10	
Входной нагрузочный коэффициент							
на *D*	F_I					1	
на $\overline{S}$ или *T*	F_I					2	
на $\overline{R}$	F_I					3	

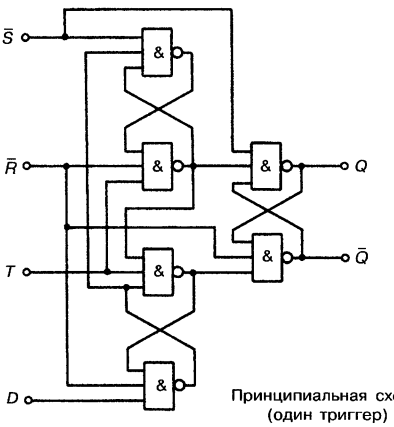
Рис. 7.72. Таблица данных микросхемы FLJ 141-7474 (по данным фирмы Siemens).

Цоколевка и принципиальная схема одного из двух *D*-триггеров показана на рис. 7.73.



Цоколевка корпуса
вид сверху

D — информационный вход
Q, $\overline{Q}$ — выходы
R — вход сброса
S — вход установки
T — вход тактирующего синхроимпульса



Принципиальная схема
(один триггер)

Рис. 7.73. Цоколевка и принципиальная схема FLJ 141-7474 (Siemens).

7.5.6. *RS*-триггеры, управляемые по обоим фронтам

Сигнал на входе поступает в управляемый по двум фронтам триггер во время прямого фронта синхроимпульса. Он не проходит сразу на выход, а временно сохраняется. Только при обратном фронте синхроимпульса выходы триггера устанавливаются.

Для этого процесса необходимы две ячейки памяти, то есть два последовательно включенных триггера. Триггер, который принимает информацию снаружи, называется **ведущим триггером** или **мастером** (master, англ. — господин). Второй триггер, который принимает информацию от мастера, называет **ведомым** или **slave** (slave, англ. — раб) (рис. 7.74). Ведущий триггер переключается при прямом фронте синхроимпульса. Ведомый — при обратном. Триггеры этого вида называются **триггерами «ведущий-ведомый» или *MS*-триггерами**.

MS-триггеры работают очень стабильно. Триггер устанавливается только после возвращения синхросигнала в исходное состояние. На рис. 7.75 изображено условное графическое обозначение *RS*-MASTER-SLAVE-триггера. Подписанный *C*-вход является *C*-входом ведущего триггера (управление прямым фронтом синхроимпульса). *C*-вход ведомого триггера не подписывается. Чтобы обозначить, что данные поступают на выходы только после заднего фронта синхроимпульса, перед выходами ставятся два уголка.

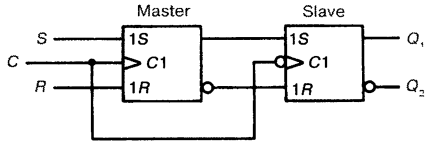


Рис. 7.74. Структура *RS*-триггера типа «ведущий-ведомый» (*MS*-триггера), управляемого по двум фронтам.

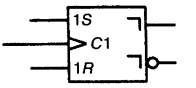


Рис. 7.75. Условное графическое обозначение *RS*-триггера типа «ведущий-ведомый».

На прямом фронте синхроимпульса, как и на обратном фронте, есть две точки включения (рис. 7.76). В этих точках происходит следующее:

t_1: ведомый триггер отделяется от ведущего триггера.

t_2: информация со входа сохраняется в ведущем триггере.

t_3: входы S и R запираются.

t_4: информация из ведущего триггера переходит в ведомый триггер и далее — на выходы Q_1 и Q_2.

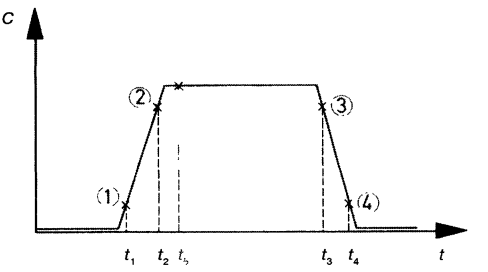


Рис. 7.76. Моменты переключения управляемого по двум фронтам *RS*-триггера типа «ведущий-ведомый». При наличии блокировочной схемы входы могут запираться в момент времени t_5.

Между точками t_2 и t_3 ведущий триггер подвержен влиянию паразитных сигналов и может их запомнить. Тогда они передадутся в ведомый триггер. Период между t_2 и t_3 должен быть как можно короче, чтобы уменьшить влияние паразитных сигналов.

С помощью специальной схемы можно добиться, чтобы запирание входов *S* и *R* происходило в момент времени t_5 (см. рис. 7.76). Этот момент времени находится примерно на 5 нс позднее t_2. *MS*-триггеры, входы которых досрочно запираются в момент времени t_5, называются *MS*-триггерами с блокированием входа.

Для управляемых по двум фронтам *RS*-триггеров имеет силу такая же таблица истинности, как и для управляемых по одному фронту *RS*-триггеров (см. рис. 7.54).

7.5.7. *JK*-триггеры, управляемые по обоим фронтам

Управляемые по двум фронтам *JK*-триггеры, как и управляемые по двум фронтам *RS*-триггеры, являются триггерами типа «ведущий-ведомый». Ведущий триггер должен быть *JK*-триггером, так как он должен переключаться при комбинации $J = 1$ и $K = 1$. В качестве ведомого триггера достаточно взять *RS*-триггер (рис. 7.77), поскольку оба выхода *JK*-триггера одновременно не могут быть равны 1. Условное обозначение этого *JK*-MASTER-SLAVE-триггера изображено на рис. 7.78. Оно отличается от условного обозначения управляемого по одному фронту *JK*-триггера только уголками перед выходами. Таблица истинности является такой же, как для управляемого по одному фронту *JK*-триггера (см. рис. 7.65). Существуют также управляемые по двум фронтам триггеры, которые управляют ведущим триггером по обратному фронту синхроимпульса. Тогда ведомый триггер переключается по прямому фронту синхроимпульса (рис. 7.79).

> В условном обозначении управляемого по двум фронтам триггера (типа «ведущий-ведомый») всегда указывается фронт синхроимпульса, по которому принимается информация. Передача информации на выход происходит по другому фронту синхроимпульса (DIN 40900, часть 12).

На рис. 7.80 представлена таблица данных микросхемы FLJ 131-7476. Эта микросхема содержит два *JK*-MASTER-SLAVE-триггера с нетактиру-

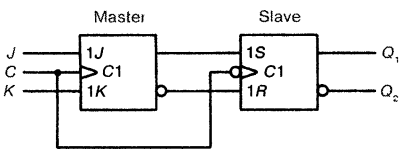


Рис. 7.77. Структура *JK*-триггера типа ведущий-ведомый.

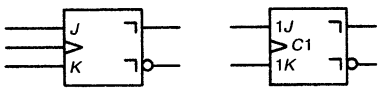
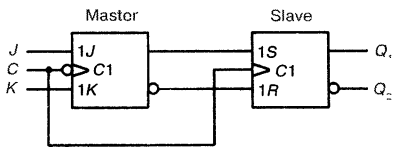
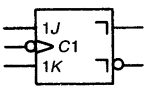


Рис. 7.78. Условное графическое обозначение *JK*-триггера типа «ведущий-ведомый».





Условное обозначение

Рис. 7.79. Структура и условное графическое обозначение *JK*-триггера типа «ведущий-ведомый», передающего информацию по обратному фронту синхроимпульса.

Сдвоенный *JK*-триггер с входами установки и сброса

Статические параметры *B* в температурных зонах 1 и 5		Условия испытаний		Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S	U_S = 4,75 В			4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В		24	2,0			В
Входное напряжение *L*-уровня	U_{IL}						0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА					1,5	В
Выходное напряжение *H*-уровня	U_{QH}	$-I_{QH}$ = 400 мкА	U_S = 4,75 В	24	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	I_{QL} = 16 мА		24		0,2	0,4	В
Статическая помехоустойчивость	U_{SS}				0,4	1,0		В
Входной ток	I_{IH}	U_{IH} = 2,4 В	U_S = 5,25 В	25			40	мкА
на *J* или *K*	I_I	U_I = 5,5 В		25			1	мА
H-входной ток	I_{IH}	U_{IL} = 2,4 В		25			80	мкА
на $\overline{R}$, $\overline{S}$ или *T*	I_I	U_I = 5,5 В		25			1	мА
L-входной ток на *J*, *K*	$-I_{IL}$	U_{IL} = 0,4 В		26			1,6	мА
на $\overline{R}$, $\overline{S}$ или *T*	$-I_{IL}$	U_{IL} = 0,4 В		26			3,2	мА
Ток короткого замыкания на канал выхода	$-I_{QH}$	U_S = 5,25 В		27	18		57	мА
Ток питания	I_S	U_I = 5,0 В		25		20	40	мА
Время переключения, при U_S = 5 В, T_U = 25 °C								
Длительность синхроимпульса	t_{pT}				20			нс
Длительность импульса установки	t_{pS}				25			нс
Длительность импульса сброса	t_{pR}				25			нс
Время подготовки	t_V			29	t_{pT}			
Время промежуточного хранения	t_H				0			

Статические параметры B в температурных зонах 1 и 5		Условия испытаний	Тестовая схема	Нижний предел B	Тип	Верхний предел A	Единица измерения
Максимальная тактовая частота	f_z	C_L = 15 пФ R_L = 400 Ом	29	15	20		мГц
Время прохождения сигнала (быстродействие)	t_{PHL}		29	10	25	40	нс
от T к Q	t_{PLH}		29	10	16	25	нс
Время прохождения сигнала (быстродействие)	t_{PHL}		30		25	40	нс
от $\overline{R}$ или $\overline{S}$ к Q	t_{PLH}		30		16	25	нс
Логические параметры							
Коэффициент разветвления по выходу на канал	F_Q					10	
Входной нагрузочный коэффициент							
на J или J	F_I					1	
на $\overline{R}$, $\overline{S}$ или T	F_I					2	

Рис. 7.80. Таблица данных микросхемы FLJ 131-7476 (Siemens).

емыми входами установки и сброса. Она принадлежит к ТТЛ-семейству. Цоколевка и принципиальная схема триггера показана на рис. 7.81.

Микросхема FLJ 341-74110 содержит *JK*-MASTER-SLAVE-триггер с блокированием входа (рис. 7.82). Функция блокирования входа для управляемых по двум фронтам *RS*-триггеров была рассмотрена выше. Она обеспечивает запирание входов через определенное время после завершения прямого фронта синхроимпульса. Это так называемое время промежуточного хранения составляет для схемы FLJ 341-74110 только 5 нс. Сигналы помехи могут привести к ложному срабатыванию только в течение этого короткого времени. Поэтому триггер очень помехоустойчив.

Три связанные логическим умножением *J*-входа и три также связанные логическим умножением *K*-входа позволяют экономически выгодно производить синхронные счетчики (см. гл. 11).

7.5.8. Дополнительные триггерные схемы

Число возможных триггерных схем чрезвычайно большое. Уже обсужденные *D*-триггеры и *T*-триггеры также производятся как *MS*-триггеры, или как управляемые по двум фронтам триггеры. Их можно также построить на базе *JK*-MASTER-SLAVE-триггеров.

К входам *JK*-MASTER-SLAVE-триггера на рис. 7.83 прикладывается 1-уровень, т. е. напряжение питания. Триггер будет переключаться на каж-

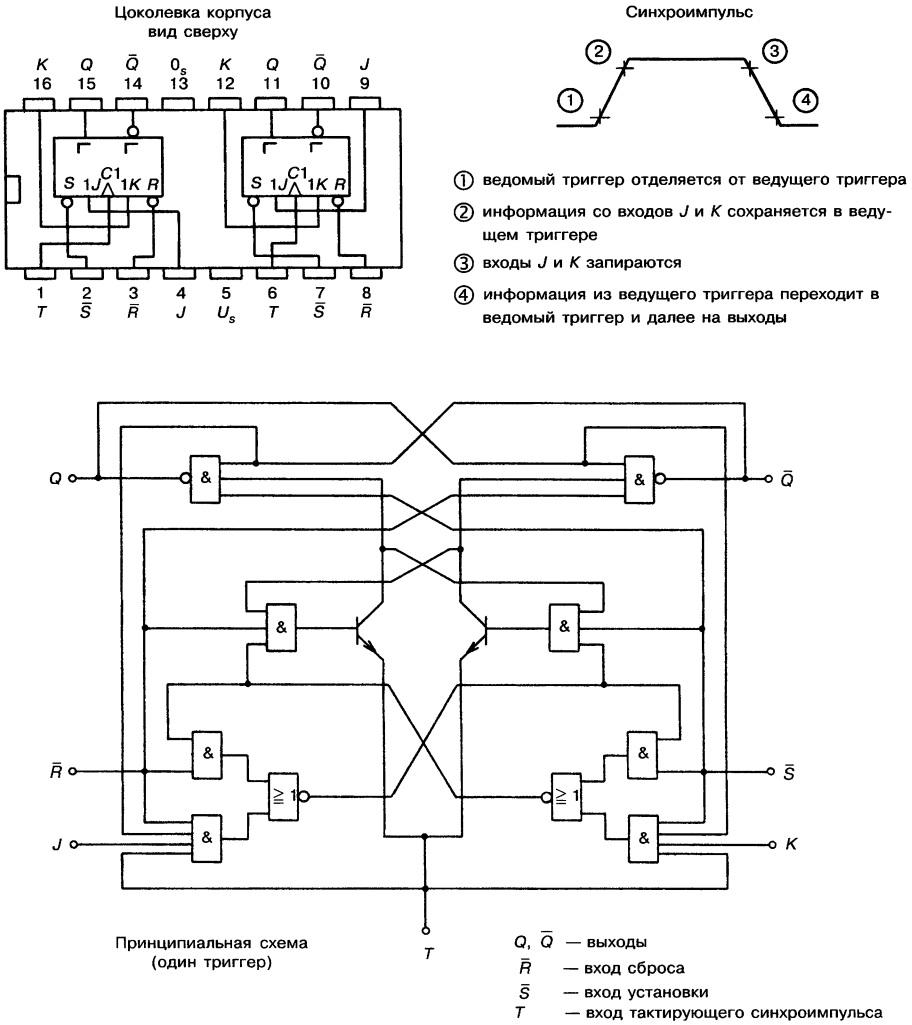


Рис. 7.81. Цоколевка, временная диаграмма и принципиальная схема FLJ 131-7476 (Siemens).

дом такте. Такая схема называется *T*-MASTER-SLAVE-триггер и является идеальной для построения асинхронных счетчиков (гл. 11).

D-MASTER-SLAVE-триггер очень легко построить на базе *JK*-MASTER-SLAVE-триггера (рис. 7.84).

Следующим интересным триггером является *DV*-триггер. Таблица истинности для *DV*-триггера приведена на рис. 7.85. Триггер работает как *D*-триггер, если на подготовительном входе *V* действует 1-сигнал. Он заперт, т. е. состояния выходов не меняются, если на подготовительном входе *V* действует 0-сигнал.

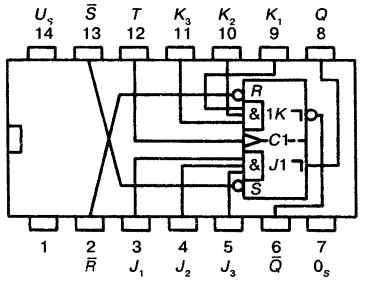
Условное обозначение *DV*-триггера показано на рис. 7.86. *DV*-триггер можно приобрести в виде управляемого по одному фронту триггера или в виде управляемого по двум фронтам *MS*-триггера. Он может быть легко преобразован в *T*-триггер (рис. 7.87).

Ведущий *JK*-триггер с блокированием входа

Модуль FLJ 341/345 имеет время промежуточного хранения всего 5 нс относительно переднего фронта синхроимпульса. Это значит, что *JK*-сигналы могут меняться уже во время синхронизирующего импульса без вызывания ложного срабатывания. FLJ 341/345 взаимозаменяем с FLJ 111/115.

Статические параметры *B* в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S		4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В	2,0			В
Входное напряжение *L*-уровня	U_{IL}				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА			1,5	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В, U_{IH} = 2,0 В, $-I_{QH}$ = 800 мкА	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В, U_{IL} = 0,8 В, I_{QL} = 16 мА		0,2	0,4	В
Входной ток на канал	I_I	U_S =5,25 В, U_{IL} = 5,5 В			1	мА
H-входной ток на *J*, *K* или *T*	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В			40	мкА
на $\overline{R}$ или $\overline{S}$	I_{IH}				160	мкА
L-входной ток на *J*, *K* или *T*	$-I_{IL}$	U_S =5,25 В, U_{IL} = 0,4 В			1,6	мА
на $\overline{R}$ или $\overline{S}$	$-I_{IL}$				3,2	мА
Ток короткого замыкания на канал выхода	$-I_Q$	U_S = 5,25 В	18		57	мА
Ток питания	I_S	U_S = 5,25 В		20	34	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						
Длительность синхроимпульса	t_{pT}		25			нс
Длительность импульса установки	t_{pS}		25			нс
Длительность импульса сброса	t_{pR}		25			нс
Время подготовки	t_V		20			нс
Время промежуточного хранения	t_H		5			нс
Максимальная тактовая частота	f_Z	C_L = 15 пФ R_L = 400 Ом	20	25		мГц
Время прохождения сигнала (быстродействие)	t_{PLH}			12	20	нс
от $\overline{R}$ или $\overline{S}$ к *Q*	t_{PHL}			18	25	нс

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел	Тип	Верхний предел	Единица измерения
Время прохождения сигнала (быстродействие)	t_{PLH}		10	20	30	нс
от T к Q	t_{PHL}		6	13	20	нс
Логические параметры						
Коэффициент разветвления						
по H-выходу на канал	F_{QH}				20	
по L-выходу на канал	F_{QL}				10	
Входной нагрузочный коэффициент						
на J или K	F_I				1	
на $\overline{R}$ и $\overline{S}$	F_I				2	
на T	F_I				3	



Цоколевка корпуса
вид сверху

J, K — информационные входы
$Q, \overline{Q}$ — выходы
$\overline{R}$ — вход сброса
$\overline{S}$ — вход установки
T — вход тактирующего синхроимпульса

Рис. 7.82. Таблица данных микросхемы FLJ 341-74110 (Siemens).

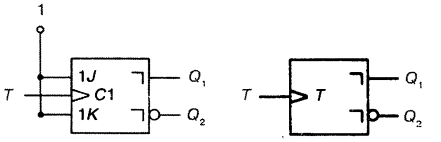


Рис. 7.83. Образование T-Master-Slave-триггера из JK-Master-Slave-триггера.

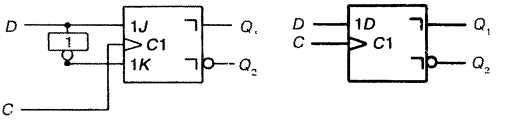


Рис. 7.84. Образование D-Master-Slave-триггера из JK-Master-Slave-триггера.

	t_n		t_{n+1}	
Вар.	D	V	Q_1	
1	0	0	Q_{1n}	хранение
2	0	1	0	установка
3	1	0	Q_{1n}	хранение
4	1	1	1	сброс

Рис. 7.85. Таблица истинности DV-триггера.

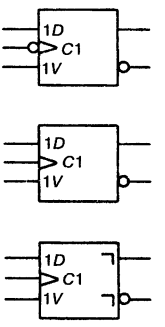


Рис. 7.86. Условные обозначения DV-триггера.

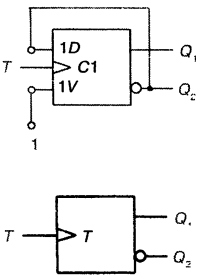


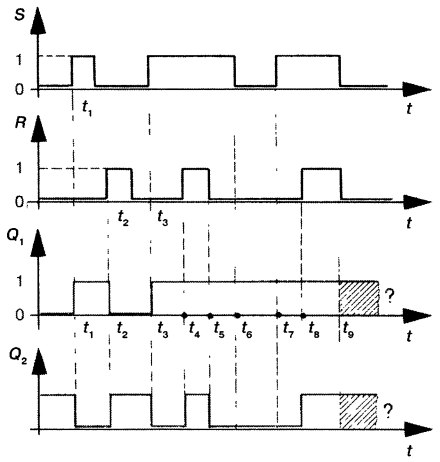
Рис. 7.87. Преобразование DV-триггера в T-триггер.

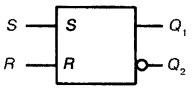
7.6. Временные диаграммы

Временные диаграммы, также называемые **импульсными диаграммами**, являются вспомогательным средством для лучшего понимания работы триггера и всей схемы в целом.

> Входные сигналы временных диаграмм заданы заранее или могут выбираться любыми. Выходные сигналы зависят от входных сигналов.

Поясним сказанное на простом примере. На рис. 7.88 изображен нетактируемый *RS*-триггер, так называемый *RS*-триггер хранения с таблицей истинности и временными диаграммами. Таблица истинности действительна для момента времени t_m.





	t_m			
Вар.	*R*	*S*	Q_1	Q_2
1	0	0	$Q_{1(m-1)}$	$Q_{2(m-1)}$
2	0	1	1	0
3	1	0	0	1
4	1	1	1	1

Рис. 7.88. *RS*-триггер хранения с таблицей истинности для момента времени t_m и временной диаграммой.

В момент времени t_1 триггер устанавливается, так как на *S* действует 1-сигнал. В момент времени t_2 триггер сбрасывается. На выходе *R* сигнал 1. В момент времени t_3 триггер снова устанавливается.

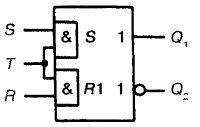
Особенно интересен момент t_4. *C* этого момента времени на обоих входах действует сигнал 1. Эта комбинация запрещена. Оба выхода переключаются в 1. Если в момент времени t_5 на *R*-вход подать 0, то Q_2 также установится в 0.

В точке t_6 *S*-сигнал равен 0. Триггер остается установленным. Сброс возможен только при $R = 1$. В точке t_7 триггер мог бы быть снова установлен. Однако он еще не установлен. Поэтому выходные состояния не изменяются.

В точке t_8 *R*-сигнал становится равным 1. Однако на *S* остается 1. Снова возникает некорректная комбинация $Q_1 = 1$ и $Q_2 = 1$. Особенно критическим является момент t_9, в котором *S* и *R* сигналы одновременно снижаются до 0. Становится неясно, в какое состояние установится триггер. Комбинация $S = R = 1$ является запрещенной и ее следует избегать.

Рассмотрим рис. 7.89. Даны входные сигналы *S* и *R* и сигнал синхронизации *T*. В момент времени t_1 $S = 1$, однако к *T* еще подведен 0-сигнал. Триггер не может установиться. Только к моменту времени t_2 триггер установится. В точке t_3 он сбросится.

В точке t_4 *S* становится равным 1, а в точке t_5 $R = 1$.



Вар.	t_n		t_{n+1}	
	R	S	Q_1	Q_2
1	0	0	Q_{1n}	Q_{2n}
2	0	1	1	0
3	1	0	0	1
4	1	1	0	1

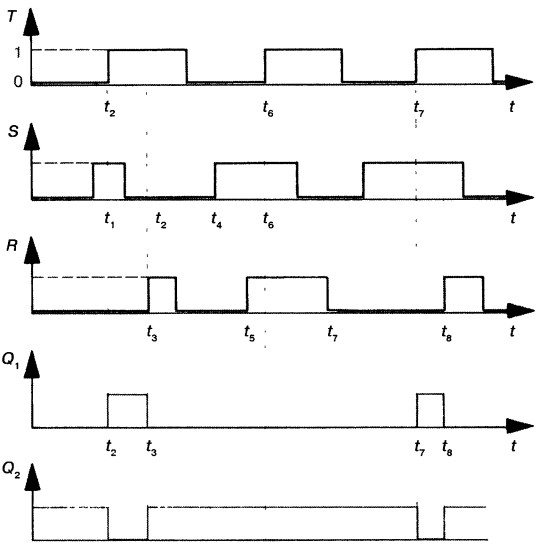
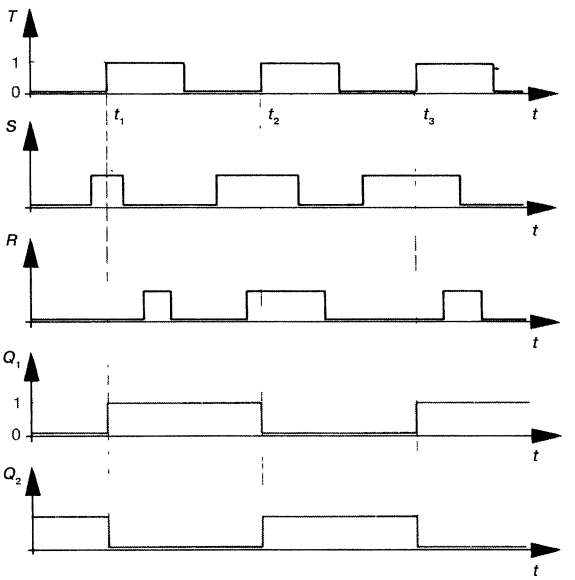


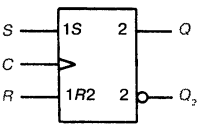
Рис. 7.89. Тактируемый *RS*-триггер с доминирующим *R*-входом, таблицей истинности и временной диаграммой.

Так как тактирующий сигнал не включен, то эти сигналы не оказывают влияния на триггер. Только в точке t_6 они действуют на триггер. Теперь триггер должен был бы сброситься. Однако он уже сброшен и, следовательно, состояние выходов не изменяется.

В точке t_7 триггер устанавливается. В точке t_8 происходит сброс, так как несмотря на единичные состояния S и R входов, R-вход имеет приоритет. Состояние выхода Q_2 находится всегда в противофазе к состоянию выхода Q_1.

Какая временная диаграмма соответствует Q_1 и Q_2, если бы *RS*-триггер с доминирующим R-входом управлялся по переднему фронту синхроимпульса? Соответствующая временная диаграмма представлена на рис. 7.90. Триггер можно переключать только в моменты времени t_1, t_2, t_3. В точке t_1 триггер





Вар.	t_n		t_{n+1}	
	R	S	Q_1	Q_2
1	0	0	Q_{1n}	Q_{2n}
2	0	1	1	0
3	1	0	0	1
4	1	1	0	1

Рис. 7.90. Управляемый по одному фронту *RS*-триггер (по переднему фронту синхроимпульса, с доминирующим R-входом) с таблицей истинности и временной диаграммой.

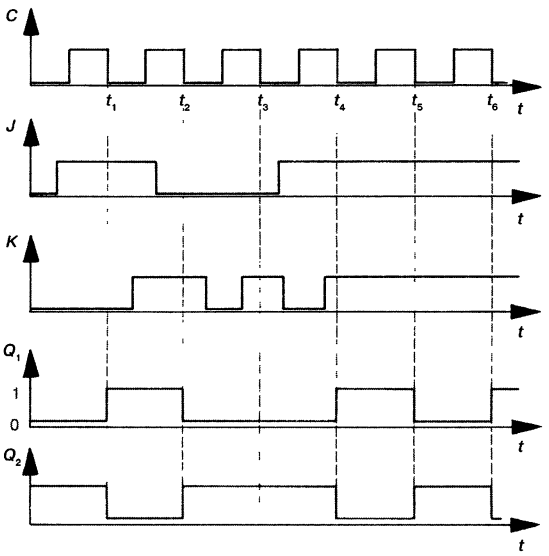
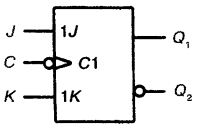


Рис. 7.91. Управляемый по одному фронту *JK*-триггер (по обратному фронту синхроимпульса) с таблицей истинности и временной диаграммой.



	t_n		t_{n+1}	
Вар.	K	J	Q_1	Q_2
1	0	0	Q_{1n}	Q_{2n}
2	0	1	1	0
3	1	0	0	1
4	1	1	$\bar{Q}_{1n}$	$\bar{Q}_{2n}$

будет установлен, так как $S = 1$. В точке t_2 триггер сбрасывается, так как $S = R = 1$. В точке t_3 триггер снова устанавливается ($S = 1$). Для Q_1 и Q_2 получается совершенно другая диаграмма, чем на рис. 7.89.

Для управляемого по одному фронту *JK*-триггера на рис. 7.91 получается следующая временная диаграмма. Триггер может переключаться только в моменты времени t_1, t_2, t_3, t_4, t_5 и t_6. Только в эти моменты времени подаются задние фронты *C*-сигнала.

В точке t_1 триггер установится, так как $J = 1$. В точке t_1 триггер сбросится, так как $K = 1$. В момент времени t_3 триггер должен был бы сброситься, однако так как он уже сброшен, то состояния выходов Q_1 и Q_2 не изменяются.

В точке t_4 $J = 1$ и $K = 1$. Триггер переключается. Так как перед моментом времени t_4 Q_1 был равен 0, то после t_4 $Q_1 = 1$. В точках t_5 и t_6 триггер переключается в противоположное состояние. Состояние выхода Q_2 всегда находится в противофазе к состоянию выхода Q_1.

В качестве следующего примера рассмотрим временные диаграммы управляемого по двум фронтам *JK*-триггера (рис. 7.92). В точке t_1 входной сигнал $J = 1$ поступает в ведущий триггер. И только в момент времени t_2 (с обратным фронтом синхроимпульса) на выходах появляются сигналы, соответствующие режиму установки $Q_1 = 1$ и $Q_2 = 0$.

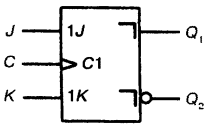
В момент времени t_3 $J = 0$ и $K = 0$. Это режим хранения. Поэтому в точке t_4 состояния выходов не меняются.

В точке t_5 входной сигнал $K = 1$ поступает в ведущий триггер. Но лишь в момент времени t_6 на выходах появляются сигналы, соответствующие режиму сброса $Q_1 = 0$ и $Q_2 = 1$.

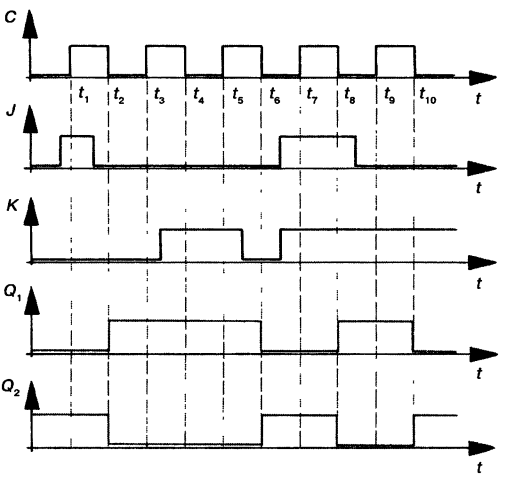
В момент времени t_7 $J = 1$ и $K = 1$. При такой комбинации входных сигналов триггер переключит выходы в состояния, противоположные текущим, но только в момент времени t_8.

В точке t_9 K становится равным 1. Сброс выходных сигналов происходит в точке t_{10}.

Рис. 7.92. Управляемый по двум фронтам *JK*-триггер (*MS*-триггер) с таблицей истинности и временной диаграммой.



	t_n		t_{n+1}	
Вар.	K	J	Q_1	Q_2
1	0	0	Q_{1n}	Q_{2n}
2	0	1	1	0
3	1	0	0	1
4	1	1	$\bar{Q}_{1n}$	$\bar{Q}_{2n}$



7.7. Характеристические уравнения

Принцип действия триггеров уже был описан словесно, таблицами истинности и временными диаграммами. Схемы, включающие в себя триггеры, должны быть удобны в расчетах. Желательно, чтобы триггерами можно было оперировать с помощью алгебры логики. Так как таблицы истинности триггеров известны, то из них можно вывести уравнения алгебры логики. Эти уравнения называются **характеристическими уравнениями**.

> Характеристическое уравнение описывает работу триггера в виде уравнения алгебры логики.

Для каждого вида триггеров можно вывести соответствующее характеристическое уравнение. Оно содержит наряду с входными и выходными переменными величинами указание на два момента времени — t_n и t_{n+1}.

t_n — момент времени перед рассматриваемым тактом.

t_{n+1} — момент времени после рассматриваемого такта.

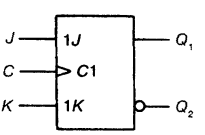
Для начала выведем характеристическое уравнение управляемого по одному фронту *JK*-триггера. Его таблица истинности изображена на рис. 7.93.

Эту таблицу истинности нужно преобразовать в полную таблицу истинности. Полные таблицы истинности рассматривались подробнее в разд. 7.4. Они содержат величину Q_1 в момент времени t_n. Получаются восемь возможных случаев (рис. 7.94), которые мы рассмотрим.

В случае 1 ($J = 0$, $K = 0$) Q_1 перед тактом равен 0. После такта Q_1 снова равен 0. В случае 2 ($J = 0$, $K = 0$) Q_1 перед тактом равен 1 и после такта также равен 1. В случаях 1 и 2 триггер работает в режиме хранения. Выходные состояния не изменяются.

Рис. 7.93. Условное обозначение и таблица истинности управляемого по одному фронту *JK*-триггера.

	t_n		t_{n+1}
Вар.	K	J	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	$\bar{Q}_{1n}$



	t_n			t_{n+1}		
Вар	K	J	Q_1	Q_1		
1	0	0	0	0	Хранение	
2	0	0	1	1		$\Rightarrow Q_1 \wedge \bar{J} \wedge \bar{K}$
3	0	1	0	1	Установка	$\Rightarrow \bar{Q}_1 \wedge J \wedge \bar{K}$
4	0	1	1	1		$\Rightarrow Q_1 \wedge J \wedge \bar{K}$
5	1	0	0	0	Сброс	
6	1	0	1	0		
7	1	1	0	1	Инверсия	$\Rightarrow \bar{Q}_1 \wedge J \wedge K$
8	1	1	1	0		

Рис. 7.94. Полная таблица истинности управляемого по одному фронту *JK*-триггера.

В случае 3 ($J = 1$, $K = 0$) до такта $Q_1 = 0$. Управляющим синхроимпульсом он будет установлен в 1. После такта $Q_1 = 1$. В случае 4 ($J = 1$, $K = 0$) Q_1 перед тактом 1. Итак, триггер уже предустановлен. Управляющий синхроимпульс не вызывает никаких изменений. Q_1 остается на 1. В случаях 3 и 4 триггер работает в режиме установки. Какое бы состояние ни имел Q_1 перед тактом, после такта всегда $Q_1 = 1$. Триггер устанавливается.

В случае 5 ($J = 0$, $K = 1$) до такта $Q_1 = 0$. Триггер должен сброситься. Однако так как он уже сброшен, то состояние выхода Q_1 с приходом такта синхроимпульса не изменится. В случае 6 ($J = 0$, $K = 1$) до такта $Q_1 = 0$. Триггер установлен перед тактом. Управляющим синхроимпульсом он будет сброшен в $Q_1 = 0$. В случаях 5 и 6 триггер работает в режиме сброса. Какое бы состояние ни имел Q_1 перед тактом, после такта всегда $Q_1 = 0$. Триггер сбрасывается.

В случае 7 ($J = 1$, $K = 1$) до такта $Q_1 = 0$. С приходом управляющего синхроимпульса выход должен переключиться в противоположное состояние. Следовательно, после такта $Q_1 = 1$. В случае 8 ($J = 1$, $K = 1$) до такта $Q_1 = 1$. С приходом управляющего синхроимпульса выход переключается в противоположное состояние. Следовательно, после такта $Q_1 = 0$. В случаях 7 и 8 триггер работает в режиме инверсии.

Из полной таблицы истинности получается нормальная форма ИЛИ (см. разд. 5.2.1). В случаях 2, 3, 4 и 7 к моменту времени t_{n+1} $Q_1 = 1$. Получаются четыре полные конъюнкции (см. рис. 7.94). Нормальная форма ИЛИ:

$$Q_{1(n+1)} = \left[\left(Q_1 \wedge \bar{J} \wedge \bar{K}\right) \vee \left(\bar{Q}_1 \wedge J \wedge \bar{K}\right) \vee \left(Q_1 \wedge J \wedge \bar{K}\right) \vee \left(\bar{Q}_1 \wedge J \wedge K\right)\right].$$

Переменные K, J и Q_1 перед рассматриваемым тактом получают индекс n. Переменная Q_1 после рассматриваемого такта получает индекс $n + 1$. Она записывается $Q_{1\,(n+1)}$.

Найденная нормальная форма ИЛИ может быть упрощена с помощью алгебры логики или диаграммы Карно (см. разд. 5.4). Соответствующая диаграмма Карно показана на рис. 7.95. Из диаграммы можно определить упрощенное уравнение:

$$\boxed{Q_{1(n+1)} = \left[\left(J \wedge Q_1\right) \vee \left(\bar{K} \wedge Q_1\right)\right]_n} \quad (JK\text{-триггер})$$

Это уравнение является характеристическим уравнением управляемого по фронту *JK*-триггера. Для уравнения не имеет значения, передний или задний фронт синхроимпульса переключает триггер. Характеристическое уравнение имеет силу для обоих видов управляемых по одному фронту

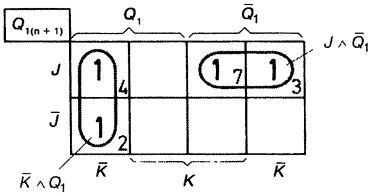
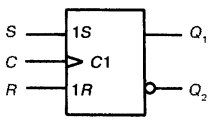


Рис. 7.95. Диаграмма Карно нормальной формы ИЛИ управляемого по фронту *JK*-триггера.



Вар.	t_n R	t_n S	t_{n+1} Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	=

Рис. 7.96. Условное обозначение и таблица истинности управляемого по фронту *RS*-триггера.

JK-триггеров. Оно имеет силу также для управляемых по двум фронтам триггеров, так как время t_n и t_{n+1} является моментом времени до и после такта, а не фронта.

Теперь выведем характеристическое уравнение управляемого по фронту *RS*-триггера. Таблица истинности в обыкновенной форме показана на рис. 7.96. Таблица истинности преобразуется в полную таблицу истинности (рис. 7.97).

Из полной таблицы истинности получается нормальная форма ИЛИ:

$$Q_{1(n+1)} = \left[\left(Q_1 \wedge \bar{S} \wedge \bar{R}\right) \vee \left(\bar{Q}_1 \wedge S \wedge \bar{R}\right) \vee \left(Q_1 \wedge S \wedge \bar{R}\right)\right]_n .$$

Нормальная форма ИЛИ упрощается с помощью диаграммы Карно (рис. 7.98). Получается следующее характеристическое уравнение:

$$Q_{1(n+1)} = \left[\left(S \wedge \bar{R}\right) \vee \left(Q_1 \wedge \bar{R}\right)\right]_n ;$$

$$\boxed{Q_{1(n+1)} = \left[\bar{R} \wedge \left(S \vee Q_1\right)\right]_n} \quad (RS\text{-триггер})$$

Мы пропустили запрещенные случаи 7 и 8 на рис. 7.97 при выводе характеристического уравнения. Эти случаи можно учесть в диаграмме Карно.

В случаях 7 и 8 $Q_{1(n+1)}$ может принимать значения как 0, так и 1.

В клетках комбинаций 7 и 8 вместо соответствующих полных конъюнкций вставлены крестики (рис. 7.99). В случае 7 полная конъюнкция была бы $\bar{Q}_1 \wedge S \wedge R$. Вместо нее стоит крест. Полная конъюнкция в случае 8 была бы $Q_1 \wedge S \wedge R$. И на ее месте стоит крест.

Вар	t_n R	t_n S	t_n Q_1	t_{n+1} Q_1	
1	0	0	0	0	Хранение
2	0	0	1	1	Хранение
3	0	1	0	1	Установка
4	0	1	1	1	Установка
5	1	0	0	0	Сброс
6	1	0	1	0	Сброс
7	1	1	0	=	Запрещенная комбинация
8	1	1	1	=	Запрещенная комбинация

$\Rightarrow Q_1 \wedge \bar{S} \wedge \bar{R}$
$\Rightarrow \bar{Q}_1 \wedge S \wedge \bar{R}$
$\Rightarrow Q_1 \wedge S \wedge \bar{R}$

Рис. 7.97. Полная таблица истинности управляемого по фронту *RS*-триггера.

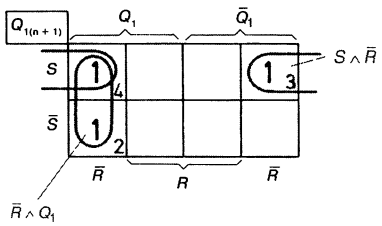


Рис. 7.98. Диаграмма Карно нормальной формы ИЛИ управляемого по фронту *RS*-триггера.

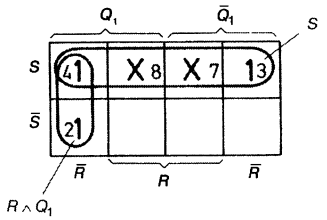


Рис. 7.99. Диаграмма Карно нормальной формы ИЛИ управляемого по фронту *RS*-триггера с обозначением полей, которые могут принимать по желанию значение 0 или 1.

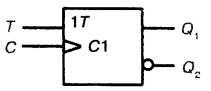
Клетки в диаграмме Карно, которые обозначены крестом, могут быть по желанию рассмотрены как 0 или 1.

Из ячеек с крестами можно образовывать группы. Благодаря этому уравнения упрощаются. Для диаграммы Карно на рис. 7.79 получается характеристическое уравнение:

$$Q_{1(n+1)} = \left[S \wedge \left(\bar{R} \vee Q_1\right)\right]_n \quad (RS\text{-триггер})$$

Как выглядит характеристическое уравнение управляемого по фронту *T*-триггера с *T* и *C* входами? На рис. 7.100 указаны таблица истинности и условное обозначение. По таблице истинности можно построить полную таблицу истинности (рис. 7.101). Нормальная форма ИЛИ:

$$Q_{1(n+1)} = \left[\left(Q_1 \wedge \bar{T}\right) \vee \left(Q_1 \wedge T\right)\right]_n .$$



t_n	t_{n+1}
T	Q_1
0	Q_{1n}
1	$\bar{Q}_{1n}$

Рис. 7.100. Условное обозначение и таблица истинности управляемого по фронту *T*-триггера с *T*-входом и *C*-входом.

	t_n		t_{n+1}	
Вар.	T	Q	Q_1	
1	0	0	0	Хранение $\Rightarrow Q_1 \wedge \bar{T}$
2	0	1	1	
3	1	0	1	Инверсия $\Rightarrow \bar{Q}_1 \wedge T$
4	1	1	0	

Рис. 7.101. Полная таблица истинности управляемого по фронту *T*-триггера.

Карта Карно на рис. 7.102 показывает, что дальнейшее упрощение нормальной формы ИЛИ невозможно. Характеристическое уравнение управляемого по фронту *T*-триггера:

$$Q_{1(n+1)} = \left[\left(Q_1 \wedge \bar{T}\right) \vee \left(\bar{Q}_1 \wedge T\right)\right]_n \quad (T\text{-триггер})$$

Для триггеров, управляемых по уровню сигнала также можно определить характеристические уравнения. Характеристическое уравнение управляемого по фронту *RS*-триггера совпадает с характеристическим уравнени-

$Q_{1(n+1)}$	Q_1	$\bar{Q}_1$
T		1
$\bar{T}$	1	

Рис. 7.102. Диаграмма Карно нормальной формы ИЛИ управляемого по фронту *T*-триггера.

ем управляемого по уровню тактового сигнала *RS*-триггера. Так получается, потому что моменты времени t_n и t_{n+1} являются моментами до и после рассматриваемого такта синхронизации. Собственно сам момент переключения, который виден на временной диаграмме, характеристическое уравнение не рассматривает.

> Полученные характеристические уравнения управляемого по одному фронту триггера действительны для триггера, тактируемого по уровню сигнала того же вида.

Принцип действия нетактируемых триггеров можно описать характеристическим уравнением. Только временные точки нужно определить иначе. Момент времени t_m является рассматриваемым моментом времени, в который на входах и выходах действуют записанные в таблице истинности сигналы. Момент времени t_{m-1} является предыдущим моментом времени с другими входными сигналами.

Условное обозначение и таблица истинности для триггера из двух элементов И-НЕ приведены на рис. 7.103. Полная таблица истинности и диаграмма Карно показаны на рис. 7.104. Получается следующее характеристическое уравнение:

$$A_{1m} = \left[\bar{E}_1 \vee (A_1 \wedge E_2)\right]_{m-1}.$$

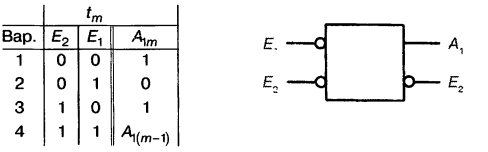


Рис. 7.103. Условное обозначение и таблица истинности нетактируемого И-НЕ-триггера.

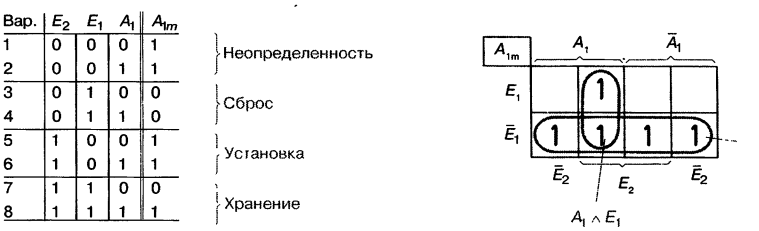


Рис. 7.104. Полная таблица истинности и диаграмма Карно нетактируемого И-НЕ-триггера.

С помощью характеристических уравнений можно рассчитывать схемы, содержащие триггеры и логические элементы (см. гл. 11).

7.8. Моностабильные ячейки

Моностабильные ячейки имеют два состояния. Одно из них называется **стабильным** состоянием.

> В стабильном состоянии на основном выходе *Q* действует 0-сигнал.

Стабильное состояние устанавливается после подачи напряжения питания. Оно сохраняется до тех пор, пока ячейка не перейдет при поступлении управляющего сигнала на вход в нестабильное состояние.

В нестабильном состоянии на основном выходе Q действует 1-сигнал.

Длительность нахождения в нестабильном состоянии определяется внешними элементами. В основном это конденсаторы (C_T) и сопротивления (R_T). Время нахождения в нестабильном состоянии определяется уравнением:

$$t_Q = 0{,}69 \cdot R_T \cdot C_T$$

t_Q — время нахождения в нестабильном состоянии.

Структура бистабильных ячеек рассмотрена в книге «Электроника», ч. 3 (Бойт/Шмуш), разд. 7.2. Бистабильные ячейки производятся в большом количестве в виде интегральных микросхем, в основном ТТЛ-семейства. На рис. 7.105 показано условное обозначение и временная диаграмма моностабильной ячейки. В момент времени t_x на входе появляется 1-сигнал. Ячейка переключается в $Q = 1$. По истечении времени t_Q она самостоятельно возвращается обратно в стабильное состояние.

Изменение входного сигнала в течение времени нахождения в нестабильном состоянии t_Q не оказывает влияния на состояние моностабильной ячейки.

Если в течение времени t_Q входной сигнал изменяет свое состояние с 0 на 1, то время нахождения в нестабильном состоянии не увеличивается.

Моностабильные ячейки управляются фронтально-тактовым способом. Они меняют состояние либо по переднему, либо по заднему фронту импульса входного сигнала.

На рис. 7.106 представлены условное обозначение и временная диаграмма моностабильной ячейки, которая переключается передним фронтом импульса входного сигнала. Ячейка переключается в момент времени t_{x1}. В течение времени t_Q она находится в нестабильном состоянии. Изменения входного сигнала в это время не влияют на состояние выхода. Возврат в

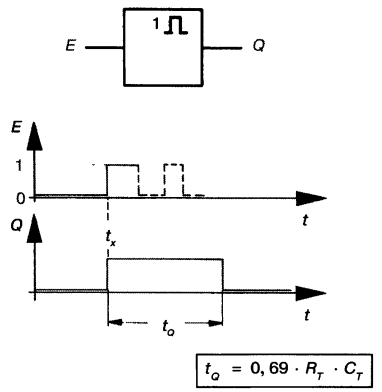


Рис. 7.105. Условное обозначение и таблица истинности моностабильной ячейки (управление состоянием).

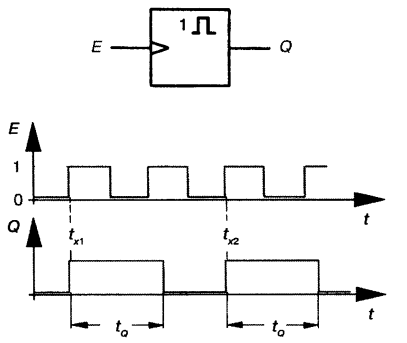


Рис. 7.106. Условное обозначение и временная диаграмма управляемой по фронту моностабильной ячейки (управление по переднему фронту).

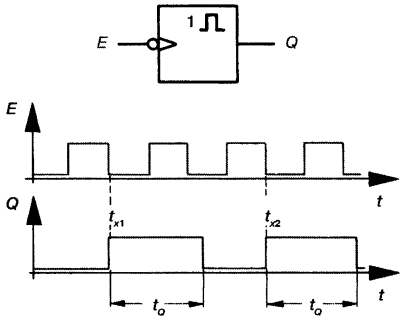


Рис. 7.107. Условное обозначение и временная диаграмма управляемой по фронту моностабильной ячейки (управление по обратному фронту).

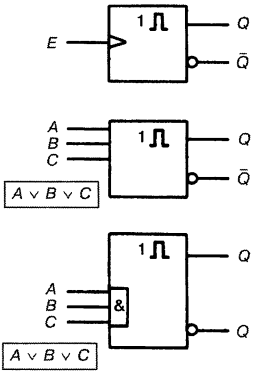


Рис. 7.108. Условные обозначения различных типов моностабильных ячеек.

стабильное состояние происходит и тогда, когда на входе присутствует 1-сигнал. Моностабильная ячейка переключается опять в точке t_{x2}.

На рис. 7.107 показаны условное обозначение и временная диаграмма управляемой по фронту моностабильной ячейки, переключаемой задним фронтом импульса.

Моностабильные ячейки могут иметь несколько выходов. Они обозначаются, как показано на рис. 7.108. Наряду с основным выходом Q имеется выход $\bar{Q}$, состояние которого всегда противоположно состоянию выхода Q. Также возможны несколько входов. Эти входы связаны друг с другом логическим сложением ИЛИ, если нет указания на какую-либо другую связь. При логическом умножении И-входы объединяются знаком логического умножения, как показано на рис. 7.108.

Время нахождения в нестабильном состоянии t_Q может указываться на условном обозначении моностабильной ячейки. Согласно DIN 40700, ч. 14, могут использоваться заглавные буквы для единиц времени, то есть *S* для секунд, *MS* для миллисекунд и *NS* для наносекунд (рис. 7.109).

Время t_Q определяется в основном внешними элементами. Входы микросхемы, к которым подключаются эти внешние элементы, обозначаются крестами. Рис. 7.110 показывает условное обозначение моностабильной ячейки с управляющим входом *E*, входом сброса *R* и входами *A* и *B*, к которым подключаются внешние элементы. 0-сигнал на *R* возвращает ячейку в $Q = 0$ (возврат в стабильное состояние). Управление происходит передним фронтом синхроимпульса. Моностабильные ячейки могут быть построены так, чтобы срабатывать с задержкой. Для таких ячеек используется условное обозначение (рис. 7.111). Время задержки может указываться в

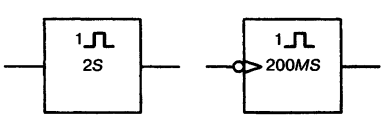


Рис. 7.109. Условное обозначение моностабильной ячейки с указанием времени нахождения в нестабильном состоянии t_Q.

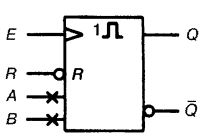


Рис. 7.110. Условное обозначение моностабильной ячейки с входами *A* и *B* и входом сброса *R*.

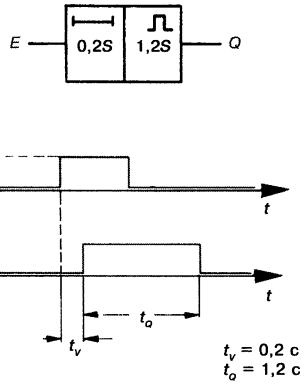


Рис. 7.111. Условное обозначение и временная диаграмма моностабильной ячейки с временем задержки $t_V = 0{,}2$ секунды и временем нахождения в нестабильном состоянии $t_Q = 1{,}2$ с.

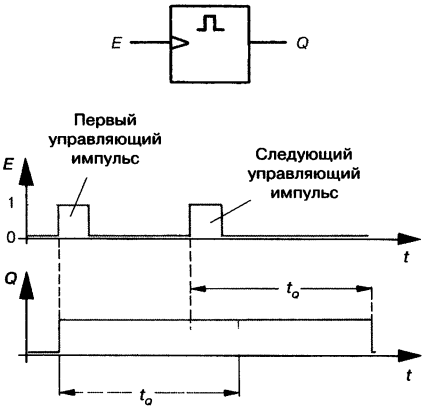


Рис. 7.112. Условное обозначение и временная диаграмма моностабильной ячейки с тактируемым t_Q (управление по переднему фронту).

условном обозначении. На рис. 7.111 оно составляет $t_V = 0{,}2$ с. Наряду с рассмотренной моностабильной ячейкой существует как частный случай моностабильная ячейка с тактируемым временем нахождения в нестабильном состоянии.

> В моностабильных ячейках с тактируемым временем нахождения в нестабильном состоянии t_Q регулируется и может быть увеличено посредством управляющих импульсов.

После переключения такой ячейки в нестабильное состояние начинается отсчет t_Q. Если в течение этого времени поступает новый управляющий импульс, то отсчет времени t_Q начинается сначала. Следующий управляющий импульс снова начинает отсчет t_Q сначала и так далее. На рис. 7.112 представлена временная диаграмма моностабильной ячейки с тактируемым временем нахождения в нестабильном состоянии.

Условное обозначение для моностабильной ячейки с тактируемым временем нахождения в нестабильном состоянии введено в DIN 40900, ч. 12. Оно совпадает с условным обозначением для обычной моностабильной ячейки, только 1 перед знаком импульса убирается (рис. 7.112). Ранее употребляемая буква *N* больше не пишется.

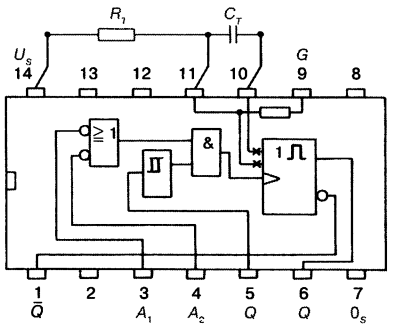


Рис. 7.113. Цоколевка микросхемы FLK 101-74121.

Производители интегральных микросхем предлагают различные моностабильные ячейки. В качестве примера приведем только одну схему. Интегральная микросхема FLK 101-74 121 построена согласно рис. 7.113. Она содержит элемент ИЛИ с инверсными входами. Через элемент И управляется моностабильная ячейка, которая переключается пе-

редним фронтом синхроимпульса. Вход *B* является входом триггера Шмитта. Через этот вход можно работать с медленно возрастающими сигналами (примерно до 1 В/с).

Если на одном из *A*-входов сигнал меняется с 1 на 0, то выходной сигнал элемента ИЛИ переходит с 0 на 1. Если ко входу *B* приложен сигнал 1, то выходной сигнал элемента И переключается с 0 на 1. Моностабильная ячейка переходит в нестабильное состояние. Если $B = 1$, то моностабильная ячейка управляется задним фронтом импульса на входах *A*. Если $B = 0$, то изменения сигнала на A-входах не оказывают воздействия.

Если управление осуществляется с *B*-входа, то на одном из входов *A* должен быть нулевой сигнал. На выходе элемента ИЛИ тогда действует 1. Начинаем медленно увеличивать сигнал на входе *B*. Когда его величина будет больше порогового значения триггера Шмитта, выходной сигнал триггера изменится с 0 на 1. Тогда выходной сигнал элемента И изменится с 0 на 1, и моностабильная ячейка переключится в нестабильное состояние.

Элементами, определяющими время нахождения в нестабильном состоянии t_Q, являются сопротивление R_T и конденсатор C_T. Конденсатор подключен к выводам 10 и 11 (положительный полюс на 11). Сопротивление подключено к выводам 11 и 14 (рис. 7.113). Возможна работа и без внешних элементов. Если подключить внутреннее сопротивление микросхемы 2 кОм (выход 9) к выходу 14 и оставить неподключенными выводы 10 и 11, то получится время нахождения в нестабильном состоянии $t_Q = 30$ нс.

Микросхема FLK 101-74121 принадлежит к схемам ТТЛ-семейства. Она предназначена для напряжения питания 5 В и имеет параметры, типичные для схем этого семейства, которые приводятся в каталогах фирм-производителей.

7.9. Элементы задержки

Логические элементы задержки, как следует из названия, задерживают прохождение сигнала. Если на входе элемента задержки происходит переход сигнала с 0 на 1, то через определенное время выход элемента меняет состояние с 0 на 1. Изменение сигнала с 1 на 0 на входе вызывает через время t_2 изменение сигнала с 1 на 0 на выходе. Условное обозначение элемента задержки приведено на рис. 7.114. Верхнее условное обозначение применяется для всех элементов задержки вообще. Нижнее условное обозначение содержит времена t_1 и t_2. Вместо t_1 и t_2 могут стоять фактические времена задержки.

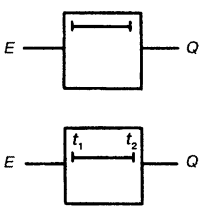
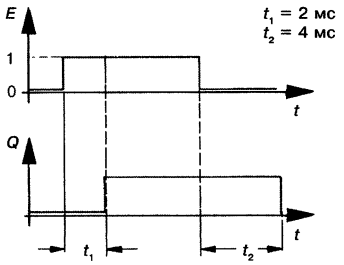


Рис. 7.114. Условное обозначение элемента задержки.

Время задержки t_1 показывает, на какое время задерживаются входные сигналы $0 \Rightarrow 1$.

Время задержки t_2 показывает, на какое время задерживаются входные сигналы $1 \Rightarrow 0$.



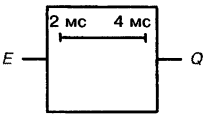


Рис. 7.115. Временная диаграмма элемента задержки.

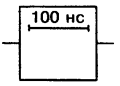


Рис. 7.116. Элемент задержки с задержкой $t_1 = t_2 = 100$ нс.

Элемент задержки на рис. 7.115 имеет время задержки $t_1 = 2$ мс и время задержки $t_2 = 4$ мс. Соответствующие временные диаграммы показывают запаздывание сигналов. Если времена t_1 и t_2 одинаковы, то в условном обозначении можно поставить только одно время (рис. 7.116).

Часто необходимы так называемые элементы задержки на включение. Эти элементы задерживают на определенное время t_1 передний фронт сигнала. Обратный фронт проходит без запаздывания $t_2 = 0$ (рис. 7.117).

Кроме элементов задержки на включение имеются также элементы задержки на выключение. Они задерживают на определенное время t_2 обратный фронт сигнала. Передний фронт проходит без запаздывания $t_1 = 0$ (рис. 7.118).

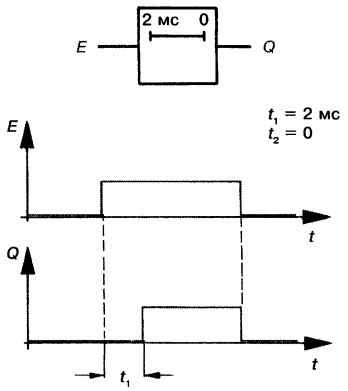


Рис. 7.117. Логический элемент задержки на включение и его временная диаграмма.

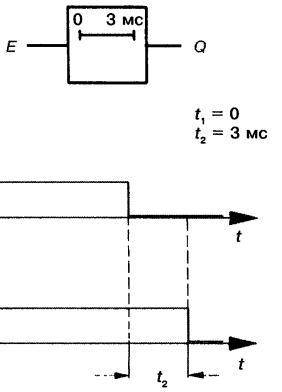


Рис. 7.118. Логический элемент задержки на выключение и его временная диаграмма.

Также производятся элементы задержки с несколькими различными временами задержки. Такие элементы называются **отводными элементами задержки**. На рис. 7.119 показана структура и условное обозначение такого элемента задержки. Отводной элемент задержки построен на нескольких простых элементах задержки.

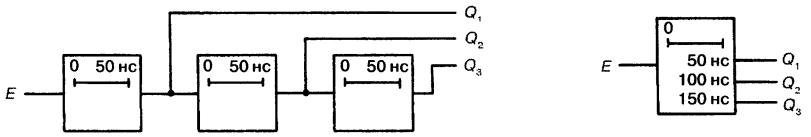


Рис. 7.119. Отводной элемент задержки на выключение.

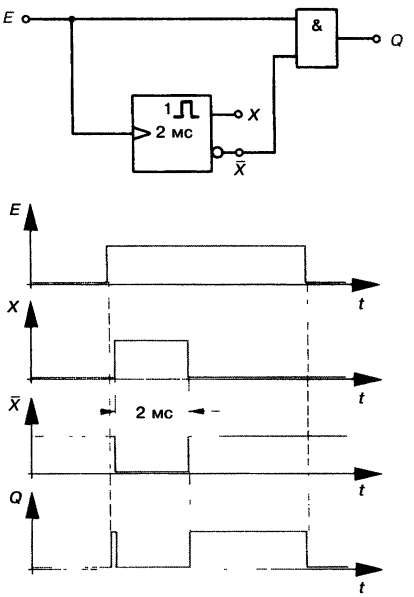


Рис. 7.120. Структура элемента задержки на включение.

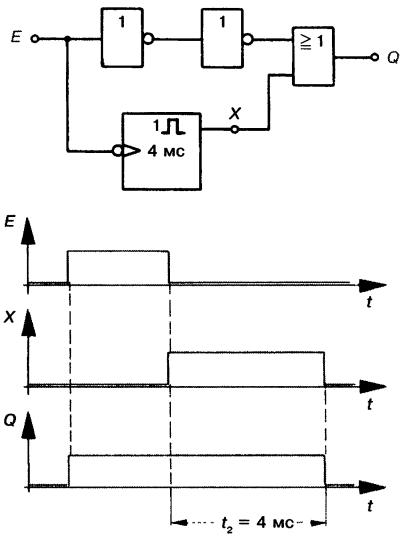


Рис. 7.122. Структура элемента задержки на выключение.

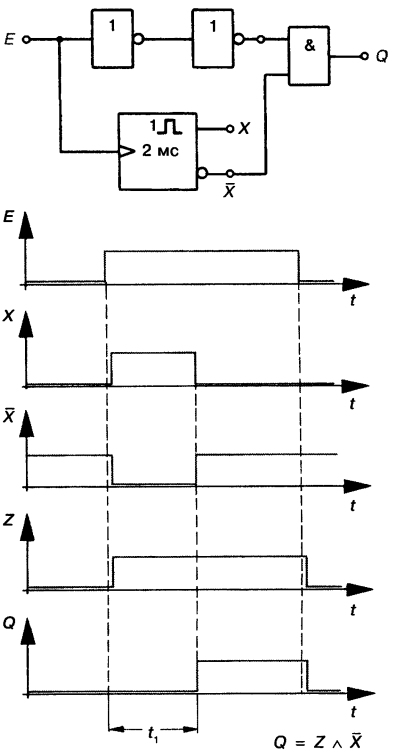
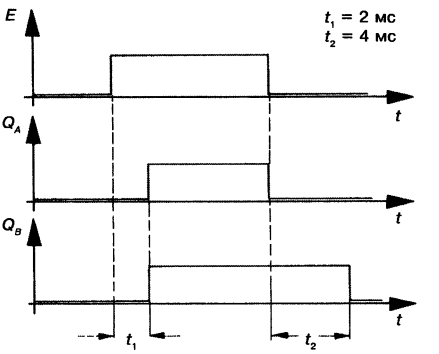
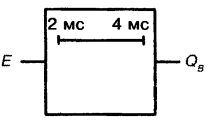
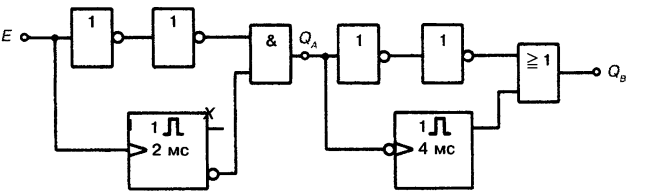


Рис. 7.121. Структура элемента задержки на включение.

Рис. 7.23. Элемент задержки на включение—выключение.







Элементы задержки производятся в виде интегральных микросхем. Они также могут строиться на моностабильных ячейках и связующих логических элементах. На рис. 7.120 показана структура элемента задержки на включение и его временная диаграмма. Проходит определенное, очень короткое время (примерно 10 нс), прежде чем включится моностабильная ячейка. Сигнал $\bar{X}$ еще равен 1, когда поступает передний фронт импульса входного сигнала. Из-за этого возникает нежелательный импульс на выходе Q. Этого импульса можно избежать, если включить последовательно два элемента НЕ согласно рис. 7.121. Каждый элемент НЕ имеет время прохождения сигнала примерно 10 нс, так что передний фронт импульса входного сигнала появится на входе элемента И на 20 нс позднее.

Элемент задержки на выключение построен согласно рис. 7.122. Здесь также необходимы два элемента НЕ для задержки времени распространения сигнала.

Если необходима задержка на включение и выключение, то можно соединить последовательно элементы задержки на включение и выключение (рис. 7.123). Желаемых времен задержки можно достигнуть применением моностабильных ячеек с внешними пассивными элементами (см. разд. 7.8).

Контрольный тест

1. Что означает условное графическое обозначение на рис. 7.124?

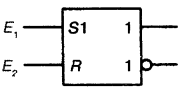


Рис. 7.124. Условное графическое обозначение.

2. Чем отличается триггер, управляемый по уровню сигнала от управляемого по фронту триггера?

3. Нарисуйте схему преобразования *RS*-триггера, управляемого по обратному фронту, в *JK*-триггер, управляемый по переднему фронту. Можно применять любые дополнительные логические элементы.

4. Как работает тактируемый *RS*-триггер с доминирующим *S*-входом? Изобразите таблицу истинности и условное графическое обозначение.

5. Объясните принцип действия моностабильной ячейки.

6. Нарисуйте временную диаграмму моностабильной ячейки, которая переключается обратным фронтом управляющего сигнала и имеет время нахождения в нестабильном состоянии 4 мс.

7. Что означает следующее уравнение:

$$Q_{1(n+1)} = \left[\left(J \wedge \bar{Q}_1\right) \vee \left(\bar{K} \wedge Q_1\right)\right]_n ?$$

8. На рис. 7.125 изображена таблица истинности триггера. Составьте полную таблицу истинности и характеристическое уравнение для этого триггера. Как называется такой триггер?

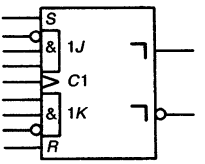
9. Как работает триггер типа «ведущий-ведомый» (*MS*-триггер)?

10. Объясните назначение входов и принцип действия изображенного на рис. 7.126 триггера.

	t_n		t_{n+1}
Вар.	E_2	E_1	Q_1
1	0	0	Q_{1n}
2	0	1	Q_{1n}
3	1	0	0
4	1	1	1

Рис. 7.125. Таблица истинности триггера.

Рис. 7.126. Условное графическое обозначение триггера.



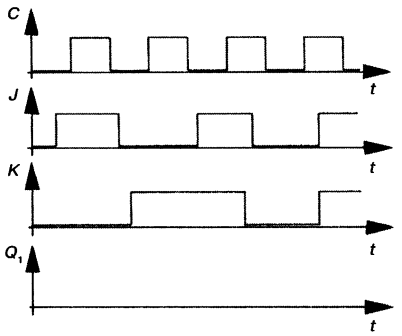
11. Изобразите таблицу истинности управляемого по одному фронту *D*-триггера.

12. Нарисуйте схему *T-MASTER-SLAVE*-триггера на базе двух *JK*-триггеров, переключающихся передним фронтом импульса.

13. Нарисуйте выходные сигналы Q_1 для временных диаграмм на рис. 7.127, если:

а) триггер переключается передним фронтом синхроимпульса,

б) триггер переключается задним фронтом синхроимпульса.



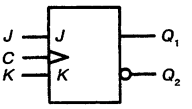


Рис. 7.127. Временная диаграмма.

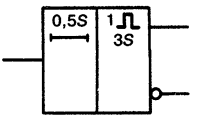


Рис. 7.128. Условное графическое обозначение.

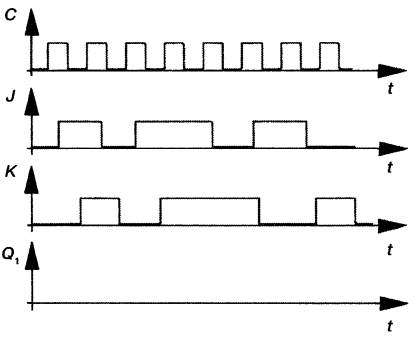
14. Какая схема представлена условным обозначением на рис. 7.128? Как работает эта схема? Нарисуйте временные диаграммы для этой схемы.

15. Разработайте схему управляемого по уровню сигнала *RS*-триггера на базе элементов И-НЕ.

16. Что понимают под записью зависимости для схем на триггерах? Приведите пример.

17. Какие сигналы Q_1 получаются при входных сигналах согласно рис. 7.129 для триггеров *I* и *II*? Изобразите временные диаграммы.

18. Элемент задержки на рис. 7.130 должен быть построен на базе любой комбинации моностабильных ячеек, переключаемых передним фронтом. Изобразите возможную схему. Какое время нахождения в нестабильном состоянии должны иметь ячейки?



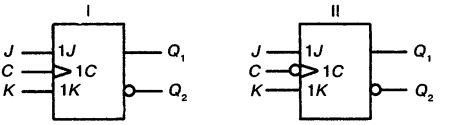


Рис. 7.129. Временная диаграмма.

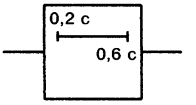


Рис. 7.130. Условное графическое обозначение элемента задержки.

ГЛАВА 8

ДВОИЧНЫЕ КОДЫ И СИСТЕМЫ СЧИСЛЕНИЯ

8.1. Введение

С помощью цифровых схем производятся различные вычисления. Поэтому необходимо представлять все десятичные цифры и все необходимые числа в виде комбинации 0 и 1. Представление числа только двумя знаками называется **бинарным представлением**.

> Коды, которые используют только два знака, называют бинарными (двоичными) кодами.

Существует множество бинарных кодов. Но на практике применяются только некоторые из них. Бинарные коды имеют регламентированную разрядность. Каждая десятичная цифра в определенном коде представляется определенным количеством так называемых бинарных разрядов. Бинарный разряд может принимать значение 0 или 1. Бинарный разряд, или базовая единица данных называется бит (от англ.: *binary digit* — двоичная цифра).

> Бит означает один бинарный разряд. Он может быть равен 0 или 1.

В двоичном коде представляются прежде всего десятичные числа. Однако оказалось целесообразным использовать также другие системы счисления. Особенное значение имеет шестнадцатеричная система счисления. Наряду с этим часто используется восьмеричная система счисления. Особенное значение имеет двоичная система счисления. Двоичная система счисления одновременно является бинарным кодом, так как состоит только из цифр 0 и 1.

8.2. Двоичная система счисления

8.2.1. Структура двоичной системы счисления

Все используемые системы счисления являются так называемыми позиционными системами счисления. В позиционной системе счисления позиция цифры однозначно связана со значением числа посредством особого фактора увеличения в виде степенного числа.

В десятичной системе счисления каждый разряд числа умножается на 10 в соответствующей степени (рис. 8.1). Чтобы посчитать от нуля до 9, нужен ноль и девять цифр в колонке единиц. Число десять запишется как 1 в колонке десятков и 0 в колонке единиц.

Если в распоряжении имеются только цифры 0 и 1, то каждый разряд числа умножается на степень числа два (рис. 8.2). В первом столбце справа

Рис. 8.1. Структура десятичной системы счисления.

Тысячи		Сотни		Десятки		Единицы
$\cdot 10^3$		$\cdot 10^2$		$\cdot 10^1$		$\cdot 10^0$
2		3		7		1
↓		↓		↓		↓
$2 \cdot 10^3$	+	$3 \cdot 10^2$	+	$7 \cdot 10^1$	+	$1 \cdot 10^0$
2000	+	300	+	70	+	1

$\cdot 16$		$\cdot 8$		$\cdot 4$		$\cdot 2$		1
2^4		$\cdot 2^3$		$\cdot 2^2$		$\cdot 2^1$		$\cdot 2^0$
1		0		1		1		0
↓		↓		↓		↓		↓
1 16	+	0 8	+	1 4	+	1 2	+	0 2
16	+	0	+	4	+	2	+	0

Рис. 8.2. Структура двоичной системы счисления.

Рис. 8.3. Десятичные числа и соответствующие им двоичные числа.

Десятичное число	Двоичное число				
	$\cdot 16$	$\cdot 8$	$\cdot 4$	$\cdot 2$	$\cdot 1$
0					0
1					1
2				1	0
3				1	1
4			1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14		1	1	1	0
15		1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	1
23	1	0	1	1	1
24	1	1	0	0	0

могут быть записаны только числа от 0 до 1. Для записи числа необходимо использовать 2 столбца справа. Число 2 получается путем записи 0 в первом столбце справа и 1 во втором столбце справа (рис. 8.3). Для представления числа 7 требуется записать 111. Первая 1 справа представляет значение 1, вторая 1 — значение 2, и третья 1 — значение 4. В итоге получается 4 + 2 + 1 = 7.

8.2.2. Перевод двоичных чисел в десятичную систему счисления

Перевести двоичное число в десятичное очень просто. Для этого используют таблицу согласно рис. 8.4. Эта таблица может быть по желанию продолжена влево.

Двоичное число заносится в таблицу (рис. 8.4). Столбцы, в которых должны быть 0, не представляют интереса. Важными являются столбцы, в которых стоит 1. Первое двоичное число на рис. 8.4 имеет 1 в столбце 2^5. Эта 1 представляет значение 32. Следующая 1 стоит в столбце 2^2. Эта 1 представляет значение 4. Общее значение двоичного числа составляет 32 + 4 = 36.

Второе двоичное число имеет 1 в столбце 2^7. Эта 1 представляет значение 128. Следующая 1 стоит в столбце 2^5. Эта 1 представляет значение 32. Обе следующие единицы представляют значения 4 и 2. Общее значение двоичного числа в итоге 128 + 32 + 4 + 2 = 166.

Десятичное число	Двоичное число										
	2^{10} 1024	2^9 512	2^8 256	2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
36						1 ↓ 32	0	0	1 ↓ 4	0	0
166				1	0	1	0	0	1	1	0
?	1	1	0	0	1	1	0	0	0	0	1
?		1	0	1	1	1	1	0	0	0	0

Рис. 8.4. Таблица для перевода двоичных чисел в десятичные числа.

Определим значения третьего и четвертого двоичных чисел на рис. 8.4. Для третьего двоичного числа должно получиться значение 1633. Четвертое двоичное число имеет значение 752.

8.2.3. Перевод десятичных чисел в двоичную систему счисления

Преобразование десятичных чисел в двоичные может также быть проведено при помощи таблицы. Таблица должна иметь достаточно большое количество столбцов. При преобразовании определяют прежде всего единицу с наибольшим значением, затем единицы с меньшими значениями. Общее значение десятичного числа разделяется на столбцы. Рассмотрим пример.

Переведем десятичное число 900 в двоичное число. 1 со значением 1024 не подходит, так как десятичное число имеет значение только 900. Наибольшая единица для этого примера 2^9. Она имеет значение 512. Итак, мы

Десятичное число	Двоичное число										
	2^{10} 1024	2^9 512	2^8 256	2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
		1	1	1	0	0	0	0	1	0	0
1300	1	0	1	0	0	0	1	0	1	0	0
1877	1	1	1	0	1	0	1	0	1	0	1

900	1300	1877	85
− 512	−1024	−1024	−64
388	276	853	21
− 256	− 276	− 512	−16
132	20	341	5
− 128	− 16	− 256	− 4
4	4	85	1
− 4	− 4		−1
0	0		0
=	=		=

Рис. 8.5. Таблица для перевода десятичных чисел в двоичные.

задействовали 512 из 900. Остается еще 388. Следующая 1 в столбце 2^8 имеет значение 256. Теперь остаток составляет только 388 – 256 = 132. 1 в столбце 2^7 имеет значение 128, так что остается 4. Остаток 4 дает 1 в столбце 2^2. В другие столбцы записывается 0. В итоге десятичное число 900 преобразовано в двоичное число 1110000100. Теперь попробуем преобразовать двоичное число в десятичное.

Преобразуйте десятичные числа 1300 и 1877 в двоичные. Получают следующие результаты:

1300 = 10100010100;
1877 = 11101010101.

8.2.4. Вещественные двоичные числа (правильные дроби)

Двоичные числа бывают, как и десятичные, с цифрами после запятой. Первому разряду справа от запятой ставится в соответствие 2^{-1}. Второму разряду справа от запятой — 2^{-2}. На рис. 8.6 показано распределение степеней числа 2 по разрядам справа от запятой.

Десятичное число	Двоичное число							
	2^3 8	2^2 4	2^1 2	2^0 1	2^{-1} 0,5	2^{-2} 0,25	2^{-3} 0,125	2^{-4} 0,0625
4,25		1	0	0,	0	1		
11,5625	1	0	1	1,	1	0	0	1

Рис. 8.6. Разложение вещественного двоичного числа по разрядам.

Двоичные числа с запятой (дробные) пересчитываются в десятичные числа таким же способом, как и двоичные числа без запятой. Соответственно можно и дробные десятичные числа перевести в двоичную систему.

Пример

Десятичное число	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1	2^{-1} 0,5	2^{-2} 0,25	2^{-3} 0,125	2^{-4} 0,0625
22,6875		1	0	1	1	0	1	0	1	1

Может так случиться, что десятичное число с запятой не может быть преобразовано в двоичное число без остатка. Тогда нужно решить, на сколько разрядов после запятой следует производить преобразование и по достижении этой разрядности закончить пересчет. Для облегчения пересчета можно использовать таблицу на рис. 8.7.

```
 22,6875          0,6875
- 16              - 0,5
 6,6875           0,1875
- 4               - 0,125
 2,6875           0,0625
- 2               - 0,0625
 0,6875           0,0
```

2^0	=	1			
2^1	=	2	2^{-1}	=	0,5
2^2	=	4	2^{-2}	=	0,25
2^3	=	8	2^{-3}	=	0,125
2^4	=	16	2^{-4}	=	0,0625
2^5	=	32	2^{-5}	=	0,03125
2^6	=	64	2^{-6}	=	0,015625
2^7	=	128	2^{-7}	=	0,0078125
2^8	=	256	2^{-8}	=	0,00390625
2^9	=	512	2^{-9}	=	0,001953125
2^{10}	=	1024	2^{-10}	=	0,0009765625
2^{11}	=	2048	2^{-11}	=	0,00048828125
2^{12}	=	4096	2^{-12}	=	0,000244140625
2^{13}	=	8192	2^{-13}	=	0,0001220703125
2^{14}	=	16384	2^{-14}	=	0,00006103515625
2^{15}	=	32768	2^{-15}	=	0,000030517578125
2^{16}	=	65536			
2^{17}	=	131072			
2^{18}	=	262144			
2^{19}	=	524288			
2^{20}	=	1048576			
2^{21}	=	2097152			
2^{22}	=	4194304			
2^{23}	=	8388608			
2^{24}	=	16777216			
2^{25}	=	33554432			

Рис. 8.7. Таблица степеней числа 2.

8.2.5. Сложение двоичных чисел

Двоичные числа складываются так же, как и десятичные числа. При сложении действуют следующие правила:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10$$
$$1 + 1 + 1 = 11$$

За одно действие складываются всегда только два числа. Если нужно сложить сумму нескольких чисел, то сначала складывают первое и второе число. Затем к результату прибавляется третье число. Далее к найденному результату прибавляется четвертое число и так далее, пока не будут сложены все числа. Сложение в столбик сразу нескольких чисел, как для десятичных чисел, в двоичном счислении не принято. Принципиально это возможно, однако в процессе вычисления возникают сложности с переносом старших разрядов.

Оба складываемых числа пишутся поразрядно друг над другом. Затем складываются две цифры столбца с наименьшим разрядом. В случае переноса единица записывается в следующий столбец и учитывается при его сложении. То есть в случае переноса нужно складывать три двоичных числа. По такому алгоритму складываются столбец за столбцом справа налево, пока не будут сложены все цифры.

Пример

2^4	2^3	2^2	2^1	2^0	
16	8	4	2	1	
		1	1		Перенос
	1	0	1	1	1. Число
1	0	0	1	1	2. Число
1	1	1	1	0	

Если преобразовать двоичные числа в десятичные, можно легко проверить правильность проведенного сложения.

$$\begin{array}{rrrrr} 1 & 0 & 1 & 1_{(2)} & \Rightarrow 11_{(10)} \\ 1 & 0 & 0 & 1 & 1_{(2)} \Rightarrow 19_{(10)} \\ \hline 1 & 1 & 1 & 1 & 0_{(2)} \Rightarrow 30_{(10)} \end{array}$$

Маленькое 2 в скобках внизу обозначает двоичное число. Число с 10 в скобках внизу является десятичным числом. Эта идентификация применяется только в случае возможных недоразумений.

8.2.6. Вычитание двоичных чисел

8.2.6.1. Непосредственное вычитание

Двоичное число может, как и в десятичной системе, вычитаться из другого двоичного числа. Такое вычитание называется **нормальным**. Для него действуют следующие правила.

Правила:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Вычитание 0 – 1 приводит к отрицательному результату. Здесь возникает несколько трудностей.

При непосредственном вычитании вычитаемое число пишется прямо под числом, от которого оно отнимается (уменьшаемым).

Пример

1	1	0	1	1	Уменьшаемое
–1	0	0	0	1	Вычитаемое
	1	0	1	0	Разность

Вычитание начинают со столбца с наименьшей степенью, с самого правого. Цифра вычитаемого вычитается из цифры уменьшаемого (1 – 1 = 0 в

примере). Затем происходит вычитание во втором столбце справа (1 – 0 = 1), затем в 3 столбце справа и т. д. Этот пример не представляет никакой трудности, так как не возникает ситуации 0 – 1. В следующем примере иначе.

Пример

```
      10
      ←
  1  (1) 0  1  1        27
-        1  1  1      - 7
  1  0  1  0  0        20
```

Чтобы провести вычитание в третьем столбце справа, «одолжим» 1 из 4-го столбца. Получается: 10 – 1 = 1. Единица в сером круге, таким образом, становится 0.

8.2.6.2. Вычитание в дополнительном коде

В компьютерной технике вычитание производится преимущественно добавлением дополнения к вычитаемому числу.

Вычитание с дополнением также возможно в десятичной системе. Предположим, что пятиразрядный спидометр машины показывает 95 000 (рис. 8.8). Если машина проедет еще 15 000 км, то спидометр покажет 10 000. Такое же число получится, если от 95 000 отнять 85 000. Число 15 000 называется дополнением к числу 85 000. Конечно, этот способ функционирует только при выполнении условия, что при прибавлении дополнения результат не отображается в шестом разряде. То есть спидометр на рис. 8.8 не может быть шестиразрядным. В компьютерной технике можно просто осуществлять запрет переносов.

При пятиразрядном представлении в десятичной системе дополнение и вычитаемое в сумме дают число 100000, т. е. 10^5. При шестиразрядном представлении в десятичной системе дополнение и вычитаемое в сумме дают 10^6. Общий принцип гласит:

> В десятичной системе дополнение и вычитаемое число дополняют друг друга при n-разрядном представлении до 10^n.

Найденное дополнение называется *B*-дополнением.

В двоичной системе вычитание в дополнительном коде производится аналогичным образом.

Рис. 8.8. Вычитание в дополнительном коде.

Пример

$$\begin{array}{rrrrr} & & & & 1\ 1\ 1\ 1 \text{ Перенос} \\ & 1 & 1 & 1 & 1 \\ - & & 1 & 1 & 1 \\ \hline & 1 & 0 & 0 & 0 \end{array} \qquad \begin{array}{rr} & 15 \\ - & 7 \\ \hline & 8 \end{array} \qquad \begin{array}{rrrrr} & 1 & 1 & 1 & 1 \\ + & \multicolumn{4}{c}{\boxed{\ ?\ }} \\ \hline & 1 & 0 & 0 & 0 \end{array} \qquad \begin{array}{rrrrr} & 1 & 1 & 1 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

В приведенном примере нужно вычесть из числа 15 число 7. Результат будет 8. Какое число должно быть прибавлено к $15_{(10)} = 1111_{(2)}$, чтобы в результате получилось $8_{(10)} = 1000_{(2)}$ при условии, что перенос в 5 разряде запрещен? Методом подбора находим число $1001_{(2)} = 9_{(10)}$. Это число является дополнением к $111_{(2)} = 7_{(10)}$.

При четырехзначном представлении дополняют вычитаемое число до $16 = 2^4$. При пятизначном представлении — до $2^5 = 32$. В следующем примере показано, как дополнение (25) и вычитаемое (7) дополняют друг друга до 32.

Пример

$$\begin{array}{rrrrrrr} & & & & & & 1\ 1\ 1\ 1 \\ & 1 & 0 & 1 & 1 & 1 & = 23 \\ - & & & 1 & 1 & 1 & = 7 \\ \hline & 1 & 0 & 0 & 0 & 0 & = 16 \end{array} \qquad \begin{array}{rrrrrrrrr} & 1 & 0 & 1 & 1 & 1 & & & \\ + & 1 & 1 & 0 & 0 & 1 & = & 25 & \\ \hline & 1 & 0 & 0 & 0 & 0 & & & \end{array}$$

Итак, можно сформулировать следующее правило.

> В двоичной системе дополнение и вычитаемое число дополняют друг друга при n-разрядном представлении до 2^n.

Если требуется найти дополнение вычитаемого числа, то прежде всего нужно знать, с каким количеством разрядов предстоит работать. В компьютерной технике разрядность известна заранее. Для наших примеров мы принимаем разрядность 6. Если вычитаемое число имеет меньше значащих цифр, чем 6, то оно расширяется ведущими нулями до достижения заданного 6-разрядного формата.

Пример

$$\underbrace{0\ 0\ 0}_{\text{Расширение}} \quad 1\ 1\ 1$$

При 6 разрядах сумма дополнения и вычитаемого должна быть равна $2^6 = 64$. Если вычитаемое число 7, то дополнение должно быть 57.

Если теперь проинвертировать расширенное число, т. е. вместо всех 0 записать 1 и вместо всех 1 записать 0, то получится число, которое только на 1 меньше искомого дополнения. Получается число 56.

Пример

32	16	8	4	2	1		
0	0	0	1	1	1	=	7
↓	↓	↓	↓	↓	↓		
1	1	1	0	0	0	=	56

Это не случайность, а закономерность, которую можно проверить на следующих примерах.

После инвертирования вычитаемого числа, расширенного до полного формата, получается число на 1 меньше дополнения к вычитаемому.

Если к полученному коду добавить 1, то получается искомое дополнение.

Для получения дополнения в двоичной системе следует воспользоваться следующим алгоритмом:

1. Расширить вычитаемое число до полноразрядного формата добавлением ведущих нулей
2. Инвертировать вычитаемое.
3. К инверсии добавить число 1.

Справедливость этого метода доказана на следующих примерах.

Пример (6-разрядное представление)

	1	0	1	1	1	1	= 47
–		1	1	0	1	1	= 27
			?				

0	1	1	0	1	1	Уменьшаемое число
↓	↓	↓	↓	↓	↓	
1	0	0	1	0	0	Инвертированное уменьшаемое число
				+	1	
1	0	0	1	0	1	Дополнение

Перенос		✗	1	1	1	1		
		1	0	1	1	1	1	= 47
Дополнение	+	1	0	0	1	0	1	
Результат:		0	1	0	1	0	0	= 20

Пример (8-разрядное представление)

	1	0	1	1	1	1	= 47
–		1	1	0	1	1	= 27
			?				

```
0 0 0 1 1 0 1 1   Уменьшаемое число
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
1 1 1 0 0 1 0 0   Инвертированное уменьшаемое число
            + 1
---------------
1 1 1 0 0 1 0 1   Дополнение
```

```
Перенос          1 1     1 1 1 1
                 | 0 0 1 0 1 1 1 1 |
Дополнение   +   | 1 1 1 0 0 1 0 1 |
Результат:       | 0 0 0 1 0 1 0 0 | = 20
```

8.2.7. Отрицательные двоичные числа

Что делать, если вычитаемое число больше, чем уменьшаемое? В результате получается отрицательное число.

Пример

```
  27          1 1 0 1 1
- 47        -1 0 1 1 1 1
-----
- 20
```

```
Образование дополнения:   1 0 1 1 1 1
                          ↓ ↓ ↓ ↓ ↓ ↓
6 разрядов                0 1 0 0 0 0
                                  + 1
                          -----------
                          0 1 0 0 0 1   Дополнение к 47
```

```
      1       1 1
      0 1 1 0 1 1   Число, из которого вычитается
+     0 1 0 0 0 1   Дополнение
-------------------
  1 0 1 1 0 0       Результат
I
```

нет переноса в 7-й разряд

Результат является отрицательным числом. Это следует из отсутствия переноса в 7-й разряд.

> Если при добавлении дополнения в n-разрядном представлении отсутствует перенос в $n + 1$ разряд, то результат является отрицательным числом.

Чтобы узнать величину отрицательного числа, нужно получить его дополнение в двоичном коде:

Пример

1	0	1	1	0	0	Результат
↓	↓	↓	↓	↓	↓	
0	1	0	0	1	1	
				+	1	
0	1	0	1	0	0	Дополнение результата

Дополнение результата равно 20.

Можно отнять определенное число от числа 0. В результате получается отрицательное вычитаемое число.

Пример

0		0	0	0	0	0
− 9		−0	1	0	0	1
− 9				?		

Образование дополнения:

0	1	0	0	1	
↓	↓	↓	↓	↓	
1	0	1	1	0	
			+	1	
1	0	1	1	1	Дополнение к 9

(5-разрядный формат)

Число 10111 равно −9. Если образовывать дополнение от этого числа, то получается число 9:

1	0	1	1	1	
↓	↓	↓	↓	↓	
0	1	0	0	0	
			+	1	
0	1	0	0	1	= 9

> Дополнение двоичного числа равно его отрицательному значению.

С помощью образования дополнения можно преобразовывать положительные двоичные числа в отрицательные. Отрицательные числа сразу распознать трудно. Найденное число −9 = 10111 может также восприниматься как положительное число 23. Нужно производить распознавание числа.

Возможный способ распознавания числа показан на рис. 8.9. Бросается в глаза, что столбец с разрядом 2^4 при положительных числах всегда содержит 0, а при отрицательных числах всегда содержит 1.

Десятичная цифра	(2^4) (16)	2^3 8	2^2 4	2^1 2	2^0 1	
+9	0	1	0	0	1	Положительный диапазон
+8	0	1	0	0	0	
+7	0	0	1	1	1	
+6	0	0	1	1	0	
+5	0	0	1	0	1	
+4	0	0	1	0	0	
+3	0	0	0	1	1	
+2	0	0	0	1	0	
+1	0	0	0	0	1	
0	0	0	0	0	0	
–1	1	1	1	1	1	Отрицательный диапазон
–2	1	1	1	1	0	
–3	1	1	1	0	1	
–4	1	1	1	0	0	
–5	1	1	0	1	1	
–6	1	1	0	1	0	
–7	1	1	0	0	1	
–8	1	1	0	0	0	
–9	1	0	1	1	1	

Рис. 8.9. Распознавание положительных и отрицательных двоичных чисел.

Десятичная цифра	2^3 8	2^2 4	2^1 2	2^0 2	
0	0	0	0	0	Тетрады
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
	1	0	1	0	Псевдотетрады
	1	0	1	1	
	1	1	0	0	
	1	1	0	1	
	1	1	1	0	
	1	1	1	1	

Рис. 8.10. Двоично-десятичный код.

> У отрицательных чисел наибольший разряд всегда равен 1.

Наибольший разряд можно назвать знаковым разрядом.

> Крайний левый разряд слева у положительных двоичных чисел равен 0, у отрицательных двоичных чисел он равен 1.

Компьютеры работают всегда с установленной разрядностью, например с 6, 8,16 или 32 разрядами. Так что наибольший разряд всегда известен и может рассматриваться безошибочно как знаковый разряд.

8.3. Двоично-десятичный код (*BCD*-числа)

Двоично-десятичный код тесно связан с двоичной системой счисления. Сокращение *BCD* происходит от английского обозначения «Binary Coded Decimals».

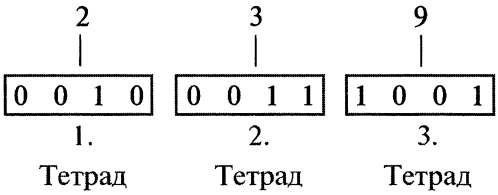
8.3.1. Представление чисел в двоично-десятичном коде

В двоично-десятичном коде каждое десятичное число представлено четырьмя бинарными разрядами, или 4 битами. Совокупность четырех битов называется **тетрадой** (по-гречески «группа из четырех»).

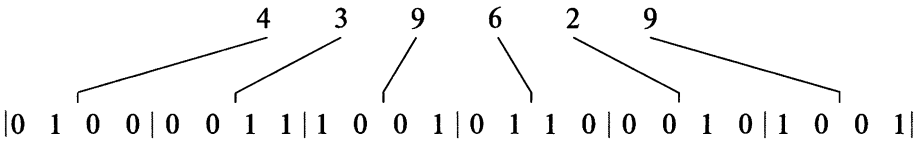
Двоично-десятичный код представлен на рис. 8.10. Каждая десятичная цифра представлена двоичным числом. Из 16 возможных тетрад используются только 10. Шесть тетрад могут быть не задействованы в двоично-десятичном коде. Они называются **псевдотетрады**. Для каждой цифры десятичного числа используется одна тетрада.

> *N*-разрядное десятичное число представляется в двоично-десятичном коде *n*-тетрадами.

Пример



Пример



8.3.2. Сложение в *BCD*-формате

Сложение происходит по тому же принципу, как и для двоичных чисел. Оно не представляет проблем до тех пор, пока результат не попадает в область псевдотетрад.

Пример

$$\begin{array}{rr} & 1\ 1 \\ & 0\ 0\ 1\ 1 \quad 3 \\ +\ & 0\ 1\ 1\ 0 \quad +6 \\ \hline & 1\ 0\ 0\ 1 \quad 9 \end{array}$$

Если при сложении возникает псевдотетрада, значит, сумма больше, чем 9 и должна представляться двумя тетрадами. В этом случае следует применить корректирующее сложение. К псевдотетраде следует прибавить число $6_{(10)} = 0110_{(2)}$. Тогда получают две тетрады.

Пример

$$\begin{array}{rr} & 1\ 1 \\ & 1\ 0\ 0\ 1 \quad 9 \\ +\ & 0\ 0\ 1\ 1 \quad +3 \\ \hline & 1\ 1\ 0\ 0 \quad 12 \end{array}$$

$$\begin{array}{rr|l} & & 1 \\ & & 1\ 1\ 0\ 0 \\ & + & 0\ 1\ 1\ 0 \quad = 6 \\ \hline 0\ 0\ 0\ 1 & & 0\ 0\ 1\ 1 \\ \underbrace{}_{1} & & \underbrace{}_{2} \end{array}$$

Пример

```
  1 1 1
  0 1 1 1      7
  1 0 0 1    + 9
---------   ----
1 0 0 0 0     16

        1 | 0 0 0 0
    +     | 0 1 1 0   = 6
-------------------
0 0 0 1 | 0 1 1 0
  1       6
```

В общем, действует правило:

> Если при сложении двух *BCD*-чисел получается результат, равный или больший $10_{(10)}$, то для коррекции к результату нужно прибавить число $0110_{(2)}$.

Сложение *BCD*-чисел, которые состоят из нескольких тетрад, происходит потетрадно справа налево. Если при сложении двух тетрад возникает переход в 5-ом разряде, то он передается к наименьшему разряду следующей тетрады. Прибавление коррекции 0110 нужно предпринимать всегда, если результат сложения двух тетрад равен или больше 10.

Пример

```
  1 1 1 [1]  ←  [1]
  0 1 1 1          1 0 0 0        78
+ 0 1 1 0        + 1 0 0 1      + 69
---------        ---------      ----
  1 1 1 0          0 0 0 1       147

        1 1 1
    1 1 1 0   0 0 0 1
  | 0 1 1 0 |+0 1 1 0
  | 0 1 0 0 | 0 1 1 1
  |  1   4  |  7
```

Если при добавлении коррекции 0110 к псевдотетраде происходит переход в 5-ом разряде, то он также добавляется к наименьшему разряду следующей тетрады.

Пример

```
1 1 1 [1]
0 1 0 1        0 1 1 1
0 0 1 1      + 1 0 0 0
---------      ---------
1 0 0 1        1 1 1 1
       ↓
         [1]   1 1
               0 1 1 1        57
             + 0 1 1 0      + 38
               ---------      ----
1 0 0 1        0 1 0 1        95
9              5
```

8.3.3. Вычитание в *BCD*-формате

Вычитание в двоично-десятичном коде производится прибавлением дополнения.

Различают дополнение до девяти и дополнение до десяти.

> Дополнение до девяти $K_{(9)}$ *BCD*-тетрады является дополнением тетрады до $1001_{(2)} = 9_{(10)}$.

Пример

Найдем дополнение до девяти от 0010.

$$\begin{array}{rrrr|r} 1 & 0 & 0 & 1 & 9 \\ -0 & 0 & 1 & 0 & -\,2 \\ \hline 0 & 1 & 1 & 1 & 7 \end{array}$$

Дополнением до девяти $K_{(9)}$ к $2_{(10)}$ будет $7_{(10)} = 0111_{(2)}$.

> Дополнение до десяти $K_{(10)}$ *BCD*-тетрады является дополнением тетрады до $1010_{(2)} = 10_{(10)}$.

Дополнение до десяти на 1 больше, чем дополнение до девяти.

Пример

Найдем дополнение до десяти от 0010.

$$\begin{array}{rrrr|r} 1 & 0 & 1 & 0 & 10 \\ -0 & 0 & 1 & 0 & -\,2 \\ \hline 1 & 0 & 0 & 0 & 8 \end{array}$$

Дополнением до десяти $K_{(10)}$ к $2_{(10)}$ будет $8_{(10)} = 1000_{(2)}$.

Если от *BCD*-тетрады *A* надо отнять *BCD*-тетраду *B*, то следует образовать дополнение до десяти *BCD*-тетрады *B* и прибавить его к *BCD*-тетраде *A*.

> Вычитание в двоично-десятичном коде сводится к сложению дополнения до десяти вычитаемого с уменьшаемым числом.

Если получается псевдотетрада, то производится коррекция прибавлением 0110. Перенос в 5-ом разряде указывает на то, что результат является положительным числом. Перенос при определении значения результата не учитывается.

Пример

$$\begin{array}{rrrrr|r} \text{A} & 1 & 0 & 0 & 1 & 9 \\ \text{B} & -0 & 1 & 1 & 1 & -\,2 \\ \hline & & ? & & & 7 \end{array}$$

$K_{(10)}$ от $7_{(10)} = 0111_{(2)}$ равен $3_{(10)} = 0011_{(2)}$.

```
   1 1
 1 0 0 1
+0 0 1 1
 1 1 0 0   (Псевдотетрада)
```

```
                1 1
Коррекция     + 1 1 0 0
              + 0 1 1 0
Результат:  1   0 0 1 0
                  2
```

Отрицательные *BCD*-числа должны быть распознаваемы как отрицательные. Если при вычитании получается отрицательное число, это должно быть сразу понятно. Нужен особенный признак.

> Если при сложении точного дополнения до десяти с *BCD*-тетрадой не образуется переноса в 5-й разряд, то результат является отрицательным числом.

Пример

```
A   0 1 1 1      7
B  -1 0 0 1     -9
                -2
     ?
```

Дополнением до десяти $K_{(10)}$ к $9_{(10)}$ будет $1_{(10)} = 0001_{(2)}$.

```
 0 1 1 1   B
+0 0 0 1   K10
 1 0 0 0   Отрицательное число
```

Нет переноса в 5-й разряд.

Результат 1000 является отрицательным числом. Чтобы найти значение этого отрицательного числа, требуется обратное дополнение. Значение отрицательного числа является его дополнением до десяти. Значит, надо искать дополнение до десяти к 1000.

Дополнением до десяти $K_{(10)}$ к $1000_{(2)} = 8_{(10)}$ будет $0010_{(2)} = 2_{(10)}$.

Получается число 2. Следовательно, результатом вычитания является -2.

8.4. Другие тетрадные системы счисления

Из множества тетрадных систем счисления, существующих наряду с двоично-десятичным кодом, рассмотрим только три самых важных. Другие тетрадные системы играют второстепенную роль и применяются только для специальных заданий.

8.4.1. Код с избытком три

Десятичная цифра	D	C	B	A	
	0	0	0	0	Псевдотетрада
	0	0	0	1	
	0	0	1	0	
0	0	0	1	1	Симметрия
1	0	1	0	0	
2	0	1	0	1	
3	0	1	1	0	
4	0	1	1	1	
5	1	0	0	0	
6	1	0	0	1	
7	1	0	1	0	
8	1	0	1	1	
9	1	1	0	0	
	1	1	0	1	Псевдотетрада
	1	1	1	0	
	1	1	1	1	

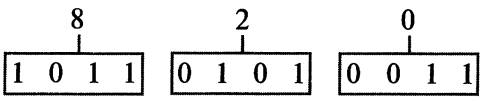
Рис. 8.11. Код с избытком 3.

В коде с избытком 3 не используются первые и последние три из 16 возможных тетрад. Эти шесть тетрад являются псевдотетрадами (рис. 8.11).

Если рассматривать тетрады кода с избытком 3 в виде двоичных чисел, то видно, что их значения все время на три больше, чем значение соответствующей десятичной цифры. Десятичное число 4 представляется как 0111, то есть двоичным числом семь. Получается симметричный код (см. рис. 8.11).

Как и в двоично-десятичном коде каждая десятичная цифра представляется тетрадой.

Пример



Преимуществом кода с избытком 3 является отсутствие тетрады 0000. Так как часто принимаются соотношения 0 ≙ 0 В и 1 ≙ U_S, то тетрада 0000 получается при исчезновении напряжения. Еще одним преимуществом является простое образование дополнения до 9.

> Дополнение до 9 в коде с избытком 3 получается простым инвертированием.

Пример

Найти дополнение до 9 к $0111 = 4_{(10)}$.

$$\begin{array}{cccc} 0 & 1 & 1 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 0 & 0 \end{array} \quad \text{Инверсия} \quad = 5_{(10)}$$

Если прибавить к дополнению до 9 единицу, то получается дополнение до десяти. Для сложения в коде с избытком 3 действуют следующие правила коррекции.

> Если при сложении двух тетрад не возникает переноса в 5-й разряд, то от результата следует отнять число 0011.

> Если при сложении двух тетрад возникает перенос в 5-й разряд, то к каждой тетраде результата следует прибавить число 0011.

Пример

$$\begin{array}{cccc} 1 & & & \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 \end{array} \qquad \begin{array}{c} 1 \\ +\ 3 \\ \hline 4 \end{array}$$

$$\begin{array}{cccc} & 1 & 0 & 1 & 0 \\ - & 0 & 0 & 1 & 1 \\ \hline & & 1 & 1 & 1 \end{array} \quad =4$$

Пример

$$\begin{array}{cccc} 1 & & & \\ & 1 & 0 & 0 & 1 \\ & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 1 \end{array} \qquad \begin{array}{c} 6 \\ +\ 9 \\ \hline 15 \end{array}$$

$$\begin{array}{cccc|cccc} & & & & 1 & 1 & 1 & \\ & & & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \underbrace{}_{1} & & & & \underbrace{}_{5} & & & \end{array}$$

Коррекция осуществляется посредством переноса в 5-й разряд. Эта операция легко осуществима в современной цифровой технике.

8.4.2. Код Айкена

В коде Айкена используются первые и последние пять из 16 возможных тетрад. Псевдотетрады находятся в середине (рис. 8.12). Получается симметричный код, в котором легко получить дополнение до 9.

> Дополнение до 9 $K_{(9)}$ получается в коде Айкена простым инвертированием.

Для отдельных позиций в пределах тетрады действуют следующие соотношения:

D	*C*	*B*	*A*
2	4	2	1

Каждая десятичная цифра выражается тетрадой.

Пример

$$\begin{array}{cccc} 1 & 9 & 0 & 1 \\ \hline 0\ 0\ 0\ 1 & 1\ 1\ 1\ 1 & 1\ 1\ 1\ 0 & 0\ 0\ 0\ 1 \end{array}$$

Десятичная цифра	② *D*	④ *C*	② *B*	① *A*	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
Псевдотетрада	0	1	0	1	
	0	1	1	0	
	0	1	1	1	
	1	0	0	0	
	1	0	1	0	
5	1	0	1	1	
6	1	1	0	0	
7	1	1	0	1	
8	1	1	1	0	
9	1	1	1	1	

Симметрия

Рис. 8.12. Код Айкена.

При сложении коррекция требуется только тогда, когда образуется псевдотетрада. Действуют следующие правила.

> Если при сложении двух тетрад в коде Айкена возникает псевдотетрада с переносом в 5-й разряд, то от результата следует отнять число 0110.

> Если при сложении двух тетрад в коде Айкена возникает псевдотетрада без переноса в 5-й разряд, то к результату следует прибавить число 0110.

Пример

$$\begin{array}{r} 1 \\ 0\ 1\ 0\ 0 \\ +1\ 1\ 1\ 1 \\ \hline 0\ 0\ 0\ 1 \mid 0\ 0\ 1\ 1 \end{array} \qquad \begin{array}{r} 4 \\ +\ 9 \\ \hline 13 \end{array}$$

Результат

Так как не образуется псевдотетрада, то коррекция не нужна.

Пример

$$\begin{array}{r} 0\ 1\ 0\ 0 \\ +\ 0\ 0\ 1\ 1 \\ \hline 0\ 1\ 1\ 1 \end{array} \qquad \begin{array}{r} 4 \\ +\ 3 \\ \hline 7 \end{array}$$

Псевдотетрада: (без переноса в 5-ом разряде)

$$\begin{array}{rr} & 0\ 1\ 1\ 1 \\ \text{Коррекция} & +\ 0\ 1\ 1\ 0 \\ & \hline & 1\ 1\ 0\ 1 \end{array} = 7 \text{ (Результат)}$$

Пример

$$\begin{array}{r} 1\ 1\ 1 \\ \boxed{1}\ 1\ 0\ 1\ 1 \\ +1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 0 \end{array} \qquad \begin{array}{r} 5 \\ +\ 7 \\ \hline 12 \end{array}$$

Псевдотетрада: (с переносом в 5-ом разряде)

$$\begin{array}{r} 1\ 0\ 0\ 0 \\ -\ 0\ 1\ 1\ 0 \\ \hline 0\ 0\ 0\ \boxed{1} \mid 0\ 0\ 1\ 0 \end{array}$$

(Результат)

8.4.3. Код Грея

Код Грея не разрабатывался с точки зрения удобства для кодирования. Его особенностью является то, что при переходе от тетрады к тетраде всегда только один разряд меняется с 0 на 1 или от 1 на 0. То есть изменяется всегда только один бит тетрады (рис. 8.13).

> Коды, в которых при переходе от одной тетрады к следующей всегда меняется только один бит, называются одношаговыми. Код Грея является одношаговым кодом.

Одношаговые коды также называют прогрессивными кодами. *BCD*-код, код с избытком 3 и код Айкена являются, напротив, многошаговыми кодами. В многошаговом коде при переходе от одной тетрады к следующей может появиться ошибка, если все биты, которые должны измениться, изменяются не одновременно. Если, например, изменился один бит, а два других бита еще нет, то вплоть до изменения других битов в наличии имеется ошибочная тетрада.

Код Грея используется прежде всего для задач управления и особенно часто — при считывании кодирующих дисков. При таком считывании нельзя обеспечить, чтобы изменение сигнала происходило одновременно по всем битам. Многошаговые коды для решения подобных задач проблематичны.

Представленный на рис. 8.13 код Грея имеет недостаток: при переходе от $9_{(10)} = 1101$ к $0_{(10)} = 0000$ должны изменяться три бинарных разряда. Говорят, код Грея не является циклическим.

Код Грея может расширяться на все 16 возможных тетрад (рис. 8.14). При расширенном коде Грея отдельные тетрады следуют в такой последовательности, что при переходе от 15 на 0 также изменяется только один бит. Расширенный код Грея является циклическим.

Десятичная цифра	*G*	*R*	*A*	*Y*
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1

Рис. 8.13. Код Грея (не циклический).

Десятичная цифра	*G*	*R*	*A*	*Y*
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Рис. 8.14. Расширенный код Грея (циклический).

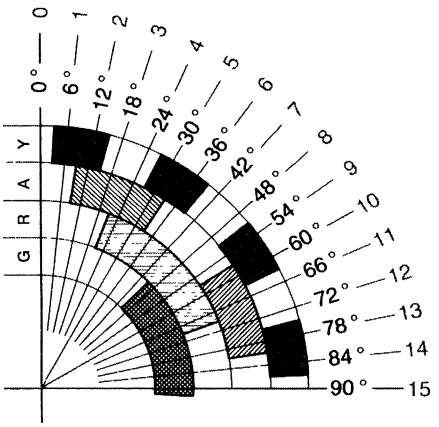


Рис. 8.15. Поворотный кодирующий диск со считыванием расширенным кодом Грея.

Прикладной областью применения кода Грея является кодирование угла. Каждой величине угла ставится в соответствие определенная тетрада кода Грея. Чаще всего используется расширенный код Грея.

На рис. 8.15 изображен поворотный кодирующий диск. Шестнадцать тетрад расширенного кода Грея поделены на 90°. Сегменты выдают 1-сигнал.

Кодирующий диск считывается электрически четырьмя щетками. Диск прикреплен к валу и вращается под фиксированными щетками. Примерно каждые 6° на четырех щетках оказывается следующая тетрада. Более тонкое разрешение получают, если 16 тетрадам ставится в соответствие 16 знаковых градусов. Однозначное кодирование возможно в этом случае только в угловом диапазоне от 0 до 15°.

8.5. Шестнадцатеричная система счисления (Hexadecimal)

8.5.1. Структура шестнадцатеричной системы счисления

Шестнадцатеричная система является позиционной системой счисления. В качестве основания используются степени числа 16.

> Каждому разряду шестнадцатеричного числа поставлена в соответствие степень числа 16.

Структура шестнадцатеричной системы счисления показана на рис. 8.16. В разряде, которому поставлена в соответствие $16^0 = 1$, можно считать до 15. Только начиная с 16 можно задействовать второй разряд. Итак, вместе с нулем в каждом разряде задействованы 16 цифр.

> В шестнадцатеричной системе счисления задействованы 16 цифр.

Прежде всего применяются известные 10 цифр десятичной системы от 0 до 9.

Десятеричная цифра	Шестнадцатеричное число				
	16^4 65 536	16^3 4096	16^2 256	16^1 16	16^0 1
520 ⇐			2	0	8
			↓	↓	↓
			$2 \cdot 256$ +	$0 \cdot 16$ +	$8 \cdot 1$

Рис. 8.16. Структура шестнадцатеричной системы.

Рис. 8.17. Шестнадцатеричная цифра.

Десятичная цифра	Шестнадцатеричная цифра
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	*A* (A)
11	*B* (B)
12	*C* (C)
13	*D* (D)
14	*E* (E)
15	*F* (F)

Для цифр от 10 до 15 можно было бы придумать какие-нибудь новые цифры. Но их не было бы на пишущих машинках и в типографиях. Поэтому вместо шестнадцатеричных цифр используют буквы *A*, *B*, *C*, *D*, *E* и *F* (рис. 8.17). Двойная функция буквы и цифры не приводит к путанице. Из контекста можно понять, обычная ли это буква или буква, выполняющая функцию цифры. Чтобы предотвратить путаницу, можно писать буквы вверх ногами, если они выполняют функцию цифры.

8.5.2. Перевод шестнадцатеричных чисел в десятичную систему счисления

Преобразование шестнадцатеричных чисел в десятичную систему счисления происходит по известному алгоритму. Целесообразно составить таблицу согласно рис. 8.17а. Количество столбцов этой таблицы определяет наибольшее возможное шестнадцатеричное число.

Десятеричное число	Шестнадцатеричное число					
	16^5 1 048 576	16^4 65 536	16^3 4096	16^2 256	16^1 16	16^0 1
41 551 ⟸			A	2	4	F
			10 · 4096 +	2 · 256 +	4 · 16 +	15 · 1
68 651 ⟸		1	0	C	2	B

Рис. 8.17а. Таблица пересчета шестнадцатеричного числа в десятичное.

Пересчет шестнадцатеричного числа *A*24*F* в десятичное производится следующим образом:

$$\begin{aligned} A &\Rightarrow 10 \cdot 4096 = 40960 \\ 2 &\Rightarrow 2 \cdot 256 = 512 \\ 4 &\Rightarrow 4 \cdot 16 = 64 \\ F &\Rightarrow 15 \cdot 1 = 15 \\ & \qquad\qquad \overline{41551} \end{aligned}$$

Для $10C2B_{(16)}$ получается:

$$\begin{aligned} 1 &\Rightarrow 1 \cdot 65536 = 65536 \\ 0 &\Rightarrow 0 \cdot 4096 = 0 \\ C &\Rightarrow 12 \cdot 256 = 3072 \\ 2 &\Rightarrow 2 \cdot 16 = 32 \\ B &\Rightarrow 11 \cdot 1 = 11 \\ & \qquad\qquad \overline{68651} \end{aligned}$$

Карманный калькулятор очень облегчает пересчет.

8.5.3. Перевод десятичных чисел в шестнадцатеричную систему счисления

При преобразовании десятичных чисел в шестнадцатеричные возникают небольшие сложности. Предлагается использовать таблицу на рис. 8.17а. Содержимое столбцов известно, и содержимое каждого столбца может быть равно от 0 до 15. Нужно записать в таблицу цифры от 0 до *F*. Составим справочную таблицу, в которой содержимое всех ячеек умножено на соответствующую степень числа 16. Такая таблица до 16^4 представлена на рис. 8.18.

		Возведение в степень				
Десятичная цифра	Шестнадцатеричная цифра	16^4	16^3	16^2	16^1	16^0
		65536	4096	256	16	1
1	1	65536	4096	256	16	1
2	2	131072	8192	512	32	2
3	3	196608	12288	768	48	3
4	4	262144	16384	1024	64	4
5	5	327680	20480	1280	80	5
6	6	393216	24576	1536	96	6
7	7	458752	28672	1792	112	7
8	8	524288	32768	2048	128	8
9	9	589824	36864	2304	144	09
10	*A*	655360	40960	2560	160	10
11	*B*	720896	45056	2816	176	11
12	*C*	786432	49152	3072	192	12
13	*D*	851968	53248	3328	208	13
14	*E*	917504	57344	3584	224	14
15	*F*	983040	61440	3840	240	15

Рис. 8.18. Таблица для пересчета десятичных чисел в шестнадцатеричные.

Если нужно представить десятичное число 1982 как шестнадцатеричное, то следует составить таблицу согласно рис. 8.19. Столбец 16^3 не требуется. Теперь в таблице на рис. 8.18 ищется самое большое число, равное или меньше 1982. Это число $1792 = 7 \cdot 256 = 7 \cdot 16^2$. В столбец 16^2 записывается цифра 7.

Число 1792 уже задействовано. Остается еще остаток 190:

$$\begin{array}{r} 1982 \\ -\ 1792 \\ \hline 190 \end{array}$$

Теперь ищется наибольшее число из таблицы на рис. 8.18, которое равно или меньше 190. Это число $176 = 11 \cdot 16^1$. В столбец 16^1 заносится цифра *B*. Остается остаток 14:

$$\begin{array}{r} 190 \\ -\ 176 \\ \hline 14 \end{array}$$

Остаток от 14 — это $14 = 14 \cdot 16^0$. В столбец 16^0 записывается цифра *E*. И остаток израсходован:

$$\begin{array}{r} 14 \\ -\ 14 \\ \hline 0 \end{array}$$

Искомое шестнадцатеричное число: 7*BE*.

Десятичное число	Шестнадцатеричное число			
	16^3	16^2	16^1	16^0
1982 ⟹		7	B	E
50 860 ⟹	C	6	A	C

Рис. 8.19. Преобразование десятичных чисел в шестнадцатеричные.

В качестве следующего примера преобразуем десятичное число 50 860 в шестнадцатеричное. Самое большое число в таблице на рис. 8.18, которое равно или меньше, чем 50 860 — это 49 152 = 12 · 4096 = 12 · 16^3. Итак, в столбец 16^3 заносится шестнадцатеричная цифра двенадцать, то есть *C*.

В столбец 16^2 заносится шестнадцатеричная цифра 6, так как 6 · 16^2 = 6 · 256, или 1536 (табл. на рис. 8.18). Остается остаток от 1708 – 1536 = 172. В столбец 16^1 записывается шестнадцатеричная цифра *A*, так как 10 · 16 = 160. Остается остаток 12. В столбец 16^0 заносится шестнадцатеричная цифра 12 = *C*. Десятичное число 50860 преобразовано в шестнадцатеричное число *C*6*AC*.

Правильность преобразования проверяется обратным преобразованием шестнадцатеричного числа в десятичное.

Проверка:

$$\begin{aligned} C &\Rightarrow 12 \cdot 4096 = 49152 \\ 2 &\Rightarrow 6 \cdot 256 = 1536 \\ A &\Rightarrow 10 \cdot 16 = 160 \\ C &\Rightarrow 12 \cdot 1 = 12 \\ & \qquad\qquad\quad \overline{50860} \end{aligned}$$

8.5.4. Перевод двоичных чисел в шестнадцатеричную систему счисления

Если нужно преобразовать двоичное число в шестнадцатеричное, то сначала оно переводится в десятичную систему счисления. Затем десятичное число преобразовывается в шестнадцатеричное так, как описано в разд. 8.5.3. Этот метод надежен, однако трудоемок. Имеется значительно более простой способ преобразования.

Между двоичной и шестнадцатеричной системами счисления имеется тесная связь. Все числа с основанием 16 могут также быть записаны как числа с основанием 2 ($16^0 = 2^0$, $16^1 = 2^4$, $16^2 = 2^8$ и т. д.). Если составить уже известную таблицу пересчета для двоичных чисел, то окажется, что содержимое каждого четвертого столбца в двоичной системе соответствует по величине содержимому столбца шестнадцатеричной системы (рис. 8.20).

Любое четырехразрядное двоичное число может быть представлено 1 шестнадцатеричным числом.

Десятичная цифра	16^2 256 2^8	128 2^7	64 2^6	32 2^5	16^1 16 2^4	8 2^3	4 2^2	2 2^1	16^0 1 2^0
		1	1	1	1	1	0	0	0

Рис. 8.20. Таблица пересчета.

Шестнадцатеричная цифра	2^3 8	2^2 4	2^1 2	2^0 1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Рис. 8.21. Шестнадцатеричные цифры, выраженные четырехразрядными двоичными числами.

С одним четырехразрядным двоичным числом можно вести счет от 0 до 15, значит, всего существуют 16 тетрад. Каждая тетрада соответствует шестнадцатеричной цифре (рис. 8.21).

Двоичные числа с разрядностью больше четырех представляются несколькими шестнадцатеричными цифрами, каждая из которых представляет четыре двоичных разряда. Если последняя группа слева содержит меньше, чем четыре разряда, то ее нужно дополнить нулями до четырех разрядов.

Группа из четырех двоичных цифр представляет одно шестнадцатеричное число.

Пример

Двоичное число $\Rightarrow$ |0 0 1 1|0 1 1 1|0 1 0 1|

Шестнадцатеричное число $\Rightarrow$ 3 7 5

$1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1_{(2)} = 375_{(16)}$

Посредством таблиц пересчета (рис. 8.22) можно проверить результат. Результат верен.

В вещественных двоичных числах с запятой нужно образовывать тетрады справа и слева от запятой.

Пример

| 0 1 1 0 | 1 1 1 1, | 1 0 1 0 | 1 0 0 0 |

$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$

6 *F* , *A* 8

$1\ 0\ 1\ 1\ 1\ 1\ 1\ ,\ 1\ 0\ 1\ 0\ 1_{(2)} = 6\ F\ ,\ A\ 8_{(16)}$

Десятичное число	Двоичное число										
	2^{10} 1024	2^9 512	2^8 256	2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
885	0	1	1	0	1	1	1	0	1	0	1
		$3_{(16)}$			$7_{(16)}$				$5_{(16)}$		

Десятичное число	Шестнадцатеричное число		
	16^2 256	16^1 16	16^0 1
885	3	7	5

$3 \cdot 256 = 768$
$7 \cdot 16 = 112$
$5 \cdot 1 = 5$
$885_{(10)}$

512
+ 256
+ 64
+ 32
+ 16
+ 4
+ 1
$885_{(10)}$

Рис. 8.22. Проверка результата.

Десятичное число	Двоичное число												
	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1	2^{-1} 0.5	2^{-2} 0,25	2^{-3} 0,125	2^{-4} 0,0625	2^{-5} 0,03125	2^{-6} 0,015625
111,65625	1	1	0	1	1	1	1,	1	0	1	0	1	

Десятичное число	Шестнадцатеричное число				
	16^6 256	16^1 16	16^0 1	16^{-1} 0,0625	16^{-2} 0.00390625
111,65625		6	F,	A	8

$6 \cdot 16 = 96$
$15 \cdot 1 = 15$
$10 \cdot 0{,}0625 = 0{,}625$
$8 \cdot 0{,}00390625 = 0{,}03125$
$111{,}65625_{(10)}$

64
+ 32
+ 8
+ 4
+ 2
+ 1
+ 0,5
+ 0,125
+ 0,03125
$111{,}65625_{(10)}$

Рис. 8.23. Проверка результата.

Проверка с помощью таблиц пересчета на рис. 8.23 показывает, что найденный результат верен.

Шестнадцатеричная система счисления часто используется, чтобы более наглядно представить длинное двоичное число.

Например, 32-разрядное двоичное число можно записать восемью шестнадцатеричными цифрами.

Пример

$$|1001|0110|1110|1111|0001|1111|0100|0111|_{(2)} = 96EF1F47_{(16)}$$

⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓

9 6 E F 1 F 4 7

8.5.5. Перевод шестнадцатеричных чисел в двоичную систему счисления

Так как нам уже знакомо преобразование двоичных чисел в шестнадцатеричные, то обратное преобразование не представляет трудности.

> Каждая шестнадцатеричная цифра представляется 4 двоичными разрядами.

При помощи таблицы на рис. 8.21 преобразование шестнадцатеричных чисел в двоичные производится очень быстро. Для каждой шестнадцатеричной цифры пишут соответствующие ей четыре двоичных разряда.

Пример

```
   E      6      0      5
   |      |      |      |
|1 1 1 0|0 1 1 0|0 0 0 0|0 1 0 1|
```

8.6. Восьмеричная система счисления (Octal)

8.6.1. Структура восьмеричной системы счисления

Восьмеричная система счисления, как и шестнадцатеричная, является позиционной системой счисления.

> Каждый разряд восьмеричного числа является множителем степени 8.

Структура восьмеричной системы счисления показана на рис. 8.24. В разряде, который умножается на $8^0 = 1$, можно считать до 7. Только начиная с 8 можно использовать второй разряд. Значит, вместе с нулем нужны 8 цифр. В восьмеричной системе счисления используются цифры, известные из десятичной системы.

> В восьмеричной системе счисления используют 8 цифр.

На рис. 8.25 показано соответствие восьмеричных цифр десятичным числам от 0 до 7. Чтобы избежать путаницы между десятичными и восьмеричными числами, записывают индекс в скобках. Индекс 8 обозначает восьмеричную систему счисления, индекс 10 — десятичную систему.

Десятичное число	Восьмеричная цифра					
	8^5 32 768	8^4 4096	8^3 512	8^2 64	8^1 8	8^0 1
2583 ⟸			5	0	2	7
			5·512	+ 0·64	+ 2·8	+ 7·1

Рис. 8.24. Структура восьмеричной системы счисления.

Рис. 8.25. Восьмеричные цифры.

Восьмеричная цифра	Десятичное число
1	1
2	2
3	3
4	4
5	5
6	6
7	7
(8)	(10)

Пример

$$2583_{(10)} = 5027_{(8)}$$

8.6.2. Преобразование восьмеричных чисел

Преобразование восьмеричных чисел в десятичные происходит по тому же принципу, как и преобразование шестнадцатеричных чисел в десятичные (разд. 8.5.2).

Алгоритм преобразования десятичного числа в восьмеричное аналогичен рассмотренному в разд. 8.5.3. Только основание степени другое.

Пример

Десятичное число 1983 нужно преобразовывать в восьмеричное число. Предлагается использовать таблицу согласно рис. 8.26. В столбце 8^3 может стоять восьмеричная цифра 3, так как $3 \cdot 512$ равно 1536. Восьмеричная цифра 3 в столбце 8^3 имеет значение 1536. Остается еще остаток 447.

Десятичное число	Восьмеричное число			
	8^3 512	8^2 64	8^1 8	8^0 1
1983	3	6	7	7

$3 \cdot 512 = 1536$
$6 \cdot 64 = 384$
$7 \cdot 8 = 56$
$7 \cdot 1 = 7$
1983

Рис. 8.26. Преобразование десятичных чисел в восьмеричные.

В столбце 8^2 может стоять восьмеричная цифра 6, так как $6 \cdot 64$ равно 384. Такое значение имеет восьмеричная цифра 6 в этом столбце. Остается еще 63:

$$\begin{array}{rl} 1983 & \\ -1536 = & 3 \cdot 512 \\ \hline 447 & \\ -\ 384 = & 6 \cdot 64 \\ \hline 63 & \end{array}$$

Для столбца 8^1 получается восьмеричная цифра 7. Она представляет значение $7 \cdot 8 = 56$. Вычитая из 63 число 56, получаем остаток 7. В столбец 8^0 записывается восьмеричная цифра 7, так как $7 \cdot 1$ равно 7:

$$\begin{array}{rl} 63 & \\ -56 = & 7 \cdot 8 \\ \hline 7 & \\ -\ 7 = & 7 \cdot 1 \\ \hline 0 & \end{array}$$

Результат преобразования:

$$1983_{(10)} = 3677_{(8)}$$

Как и для шестнадцатеричной системы, между двоичной и восьмеричной системами счисления имеется тесная связь. Все числа с основанием 8 также могут быть записаны как числа с основанием 2 ($8^0 = 2^0$, $8^1 = 2^3$, $8^2 = 2^6$ и т. д.). Если составить уже известную таблицу пересчета для двоичных чисел, то окажется, что содержимое каждого третьего столбца в двоичной системе соответствует по величине содержимому столбца восьмеричной системы (рис. 8.27).

Любое трехразрядное двоичное число может быть представлено одним восьмеричным числом.

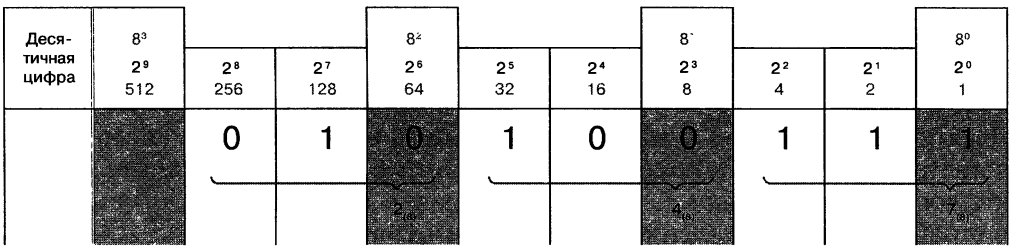


Рис. 8.27. Таблица пересчета.

С одним трехразрядным числом можно вести счет от 0 до 7, значит, всего существуют 8 триад (троек бинарных разрядов). Каждая триада соответствует восьмеричной цифре (рис. 8.28).

Восьмеричное число	Двоичное число		
	2^2	2^1	2^0
	4	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	0	1	1

Рис. 8.28. Шестнадцатеричные цифры, выраженные с помощью 3-разрядных двоичных чисел.

Двоичные числа с разрядностью больше чем три представляются несколькими восьмеричными цифрами, каждая из которых представляет три двоичных разряда. Если последняя группа слева содержит меньше, чем три разряда, то ее нужно дополнить нулями до трех разрядов.

Группа из трех двоичных цифр представляет одно восьмеричное число.

Пример

Двоичное число ⇒ | 0 0 1 | 1 0 1 | 1 1 0 | 1 0 1 |

Восьмеричное число ⇒ 1 5 6 5

Итак, преобразовывать двоичные числа в восьмеричные очень легко.

Если нужно преобразовать двоичное число в восьмеричное, то для каждой восьмеричной цифры записывают соответствующие три двоичных разряда.

Каждая восьмеричная цифра представляется тремя двоичными разрядами.

Пример

$$\begin{array}{cccc} 3 & 6 & 7 & 7 \\ | & | & | & | \\ |111|&110|&111|&111| \end{array}$$

$$3677_{(8)} = 111\,110\,111\,111_{(2)} = 1983_{(10)}$$

Согласно рис. 8.26 является $3677_{(8)} = 1983_{(10)}$.

Если нужно преобразовать восьмеричное число в шестнадцатеричное, то это очень удобно сделать через двоичную систему счисления. Восьмеричное число расписывается как двоичное, группируется по тетрадам, и затем каждая тетрада заменяется соответствующей шестнадцатеричной цифрой.

Пример

$$3677_{(8)} = \underbrace{0111}_{7} \mid \underbrace{1011}_{B} \mid \underbrace{1111}_{F}$$

$$3677_{(8)} = 7BF_{(16)}$$

8.7. Коды, распознающие ошибки

8.7.1. Понятие избыточности

Распознавание ошибок возможно только при наличии дополнительной избыточной информации, то есть передаваемая информация превышает необходимый минимум. Если оратор очень кратко о чем-то рассказывает, то часто по его речи нельзя оценить правильность высказывания. Если же оратор во время выступления дает дополнительные сведения, то возможна проверка правильности сказанного. Такая дополнительная информация называется избыточной (redundans, лат. — в изобилии).

Наш язык и письменность содержат довольно большую избыточность. Только благодаря избыточности можно идентифицировать орфографические ошибки и опечатки. Это становится ясно, если мы рассмотрим пример информации без избыточности. Цифра 7 представляется в двоично-десятичном коде как 0111. Если в процессе передачи данных 1 ошибочно передастся как 0, то получится 0101. Это уже цифра 5. Без дополнительной информации мы не узнаем, что полученная 5 ошибочна.

Если цифра 7 будет передаваться в словесной форме как «семь» и в процессе передачи будет изменена одна буква, то ошибка будет сразу очевидна (например, «семв» вместо «семь»). Словесная форма содержит дополнительную информацию, избыточность.

> Избыточность существует всегда в том случае, если кроме собственно информации передаются дополнительные сведения, которые помогают при узнавании или исправлении ошибки.

Чтобы определить наличие ошибки, во многих случаях хватает незначительной избыточности. Если ошибка должна быть не только идентифицирована, но и исправлена, требуется больше дополнительных сведений — большая избыточность.

> Для исправления ошибок необходима большая избыточность, чем только для их идентификации.

Потребность в идентификации ошибок и их исправлении привела к появлению специализированных кодов.

8.7.2. Дополнительный двоичный код

Лучшим примером избыточного кодирования для первого знакомства является расширенный двоичный код. На рис. 8.29 показан уже рассмотренный двоично-десятичный код. Он содержит дополнительный разряд, дополнительный 5-й бит. Столбец 5-го бита обозначен на рис. 8.29 как *E*.

5-ым битом двоично-десятичный код дополняется на «четность». Это значит, что он дополняется таким образом, чтобы количество битов, имеющих значение 1, было четным.

Для десятичной цифры 0 дополнение не требуется. Десятичная цифра 1 записывается как 0001. Количество битов, которые имеют значение 1, равно 1, т. е. нечетно. Таким образом, 5-й бит получает значение 1. В десятичной цифре 2 (0010) также только один бит равен 1. Следовательно, *E* получает значение 1. В десятичной цифре 3 (0011) два бита имеют значение 1. Количество битов, равных 1, является четным. *E* получает значение 0 и т. д.

Десятичная цифра	2^3 8	2^2 4	2^1 2	2^0 1	E
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0

Рис. 8.29. Образование двоичного дополнительного кода из *BCD*-кода.

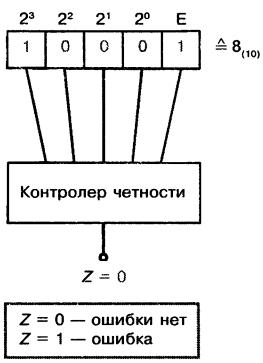


Рис. 8.30. Распознавание ошибки при помощи контроля четности.

Каждая десятичная цифра представляется 5-битовой кодовой комбинацией. 5-й бит является дополнительной информацией, т. е. избыточным. Он называется контрольным разрядом или битом.

Каждая 5-битовая кодовая комбинация проверяется особенной схемой, так называемым контролером четности, на четность единиц (рис. 8.30). Если комбинация четная, то $Z = 0$. Если нечетная, то $Z = 1$. При $Z = 1$ появляется сообщение об ошибке.

Если при передаче данных ошибочно вместо 0 передан 1 или вместо 1 передан 0, то выводится сообщение об ошибке. Определяется только то, что переданная десятичная цифра ошибочна. Неизвестно, какая она должна быть на самом деле. Значит, она не может быть исправлена.

Если в 5-битовой кодовой комбинации два бита ошибочны, то сообщение об ошибке не выдается, так как число единичных битов снова четное. Такие ошибки не распознаются в расширенном двоичном коде.

Вероятность возникновения такой ошибки очень мала. Если она все же возникнет, то скорее всего в данном сеансе связи имеют место много распознаваемых ошибок с одним неверным битом, и будет выведено сообщение об ошибке в передаче данных.

8.7.3. Код «2 из 5»

Кроме расширенного двоичного кода существует множество 5-битовых кодов, из которых так называемые коды «2 из 5» имеют особенное значение. В этом коде распознавание ошибки происходит так же, как и при двоичном расширенном коде, при помощи проверки четности.

На рис. 8.31 показаны кодировочные таблицы для лексикографического кода, кода Волкинга, кода 7-4-2-1-0 и кода 8-4-2-1-0.

	Лексикографический код					Код Волкинга					7-4-2-1-0-код					8-4-2-0-1-код				
Номер бита	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
Вес	Нет					Нет					7	4	2	1	0	8	4	2	1	0
Десятичная цифра																				
0	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0
1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1
2	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	1	0	1
3	1	0	0	1	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1	0
4	1	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1
5	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0
6	0	1	0	1	0	1	1	0	0	0	0	1	1	0	0	0	1	1	0	0
7	0	1	0	0	1	0	1	0	0	1	1	0	0	0	1	1	1	0	0	0
8	0	0	1	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	1
9	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0

Рис. 8.31. Кодировочные таблицы важнейших кодов 2 из 5.

Лексикографический код и код Волкинга не различают «вес» двоичных разрядов. В 7-4-2-1-0-коде двоичным разрядам присвоены веса 7, 4, 2, 1 и 0. *Вес не имеет значения для десятичной цифры 0, т. е. для первой строки* таблицы кода.

В 8-4-2-1-0-коде двоичным разрядам присвоены веса 8, 4, 2, 1 и 0. Это различие действует ограниченно, т. е. недействительно для десятичных цифр 0 и 7.

		Лексикографический код					Код Волкинга					7-4-2-1-0-код					8-4-2-0-1-код				
Номер бита		5	4	3	2	1	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
Вес		Нет					Нет					7	4	2	1	0	8	4	2	1	0
Десятичная цифра	0				■	■				■	■	■	■				■		■		
	1	■	■						■		■				■	■				■	■
	2	■		■					■	■				■		■			■		■
	3	■			■			■		■				■	■				■	■	
	4	■				■		■	■				■			■		■			■
	5		■	■			■		■				■		■			■		■	
	6		■		■		■	■					■	■				■	■		
	7		■			■		■			■	■				■	■	■			
	8			■	■						■	■			■		■				■
	9			■		■	■			■		■		■			■			■	

Рис. 8.32. Таблицы перевода важнейших кодов 2 из 5.

Кроме кодовых таблиц кода с 0 и 1 также распространены так называемые таблицы перевода. В таблицах перевода каждая 1 обозначена заштрихованным полем, а каждый 0 — пустым (рис. 8.32). Это представление очень наглядно.

8.7.4. Код «3 из 5»

Из 5-битовых комбинаций также можно построить код 3 из 5. Каждая 5-битовая комбинация содержит три состояния 1 и два 0. Часто используется код Лоренца и шифровальный код номер 3. Таблицы перевода представлены на рис. 8.33.

Для распознавания ошибки проводится проверка на нечетность. 5-битовая комбинация безошибочна только при условии, что три ее бита имеют состоя-

		Код Лоренца					Шифровальный код № 3				
Номер бита		5	4	3	2	1	5	4	3	2	1
Вес		4	3	2	1	0	Нет				
Десятичная цифра	0	■			■	■		■		■	■
	1	■		■			■	■			
	2	■	■				■	■		■	
	3			■	■		■	■			■
	4		■		■		■		■	■	
	5		■	■		■	■		■		■
	6		■	■	■		■			■	■
	7	■		■	■				■	■	■
	8	■	■		■			■	■	■	
	9	■		■		■		■	■		■

Рис. 8.33. Таблицы перевода важнейших кодов 3 из 5.

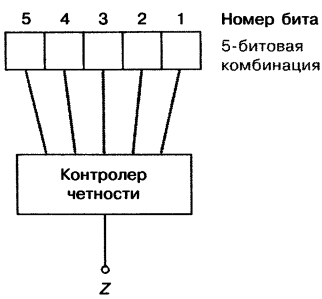


Рис. 8.34. Распознавание ошибки при помощи контроля нечетности.

ние 1 и два бита состояние 0. Если это не так, то контролер нечетности на выходе показывает состояние 1 и выдает сообщение об ошибке (рис. 8.34).

Коды «3 из 5» используются прежде всего для гарантированной передачи чисел на большие расстояния.

8.7.5. Код «2 из 7»

Коды «2 из 7» состоят из 7-битовых комбинаций. 7-битовые комбинации также называются 7-битовым словом. Каждая десятичная цифра представляется 7 битами. Получаемая при этом избыточность больше, чем при представлении цифры только 5 битами.

Из 7 битов 2 бита всегда имеют состояние 1 и 5 битов состояние 0. Два часто применяемые кода «2 из 7» показаны на рис. 8.35. Это двоично-пятеричный код и отраженный двоично-пятеричный код. Биты 6 и 7 образуют код «1 из 2». Биты номер 5, 4, 3, 2 и 0 образуют код 1 из 5. Такая структура кода позволяет относительно простую обработку 7-битового слова. Отраженный двоично-пятеричный код получается простым образованием дополнения. Дополнение образуется путем замены 1 на 0 в битах номер 6 и 7.

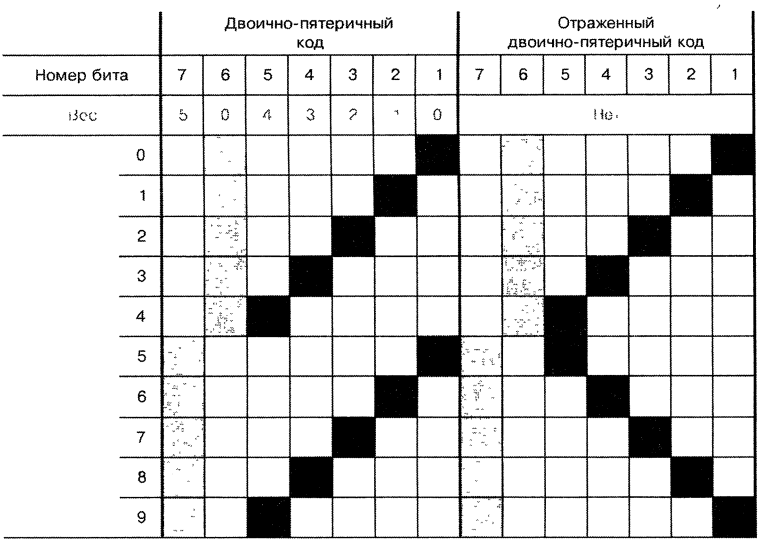


Рис. 8.35. Таблица перевода двоично-пятеричного кода и отраженного двоично-пятеричного кода.

8.8. Коды, исправляющие ошибки

8.8.1. Принцип действия

Прежде чем ошибка может быть исправлена, она должна быть обнаружена.

> Код, исправляющий ошибки, также является кодом, распознающим ошибки.

По сравнению с кодом, распознающим ошибки, код с исправлением ошибок обладает большей избыточностью. Для передачи одного знака требуется на несколько бит больше. Десятичная цифра представляется, например, 7 битами, а не как в коде, распознающим ошибки, 5 битами.

Большая избыточность позволяет выявить ошибочный бит. Если известен ошибочный бит, то самостоятельная коррекция относительно проста. Если ошибочный бит содержит 1, то правильное значение 0. Если ошибочный бит содержит 0, то правильное значение 1. То есть ошибочный бит должен быть инвертирован.

> Код с исправлением ошибок производит самостоятельную коррекцию ошибочного бита.

Сообщение об ошибке может выдаваться независимо от самокоррекции. Во многих случаях желательна регистрация выявленной ошибки.

В основном коды с исправлением ошибок могут исправлять только одну ошибку в слове. Если в слове ошибочны два бита, то в большинстве случаев выдается сообщение только об одной ошибке. То есть самостоятельное исправление ошибки не произойдет. Вероятность того, что в слове будут одновременно ошибочны два бита, чрезвычайно мала. Если, тем не менее, такая ошибка встретилась, то после сообщения об ошибке нужно остановить передачу данных и устранить ее причину.

В настоящее время разработаны коды, которые могут исправлять два и больше битов в слове. Но им нужно еще больше битов для передачи одного слова, и они так сложно устроены, что их применение экономически нецелесообразно.

8.8.2. Код Хемминга

Среди кодов, распознающих ошибки, чаще всего используется код Хемминга, также называемый Хемминг-дополненным двоично-десятичным кодом. Для передачи одной десятичной цифры в коде Хемминга необходимы 7 бит (рис. 8.36).

По схеме кодирования Хемминга после каждых четырех бит данных добавляются три контрольных бита.

		Код Хемминга						
Номер бита		7	6	5	4	3	2	1
Вес		K	K	2^3	K_2	2^2	2^1	2^0
Десятичная цифра	0							
	1							
	2							
	3							
	4							
	5							
	6							
	7							
	8							
	9							

Номер бита		1	2	3	4	5	6	7
Вес		K	K_1	2^3	K_2	2^2	2^1	2^0
Десятичная цифра	0	0	0	0	0	0	0	0
	1	1	1	0	1	0	0	1
	2	0	1	0	1	0	1	0
	3	1	0	0	0	0	1	1
	4	1	0	0	1	1	0	0
	5	0	1	0	0	1	0	1
	6	1	1	0	0	1	1	0
	7	0	0	0	1	1	1	1
	8	1	1	1	0	0	0	0
	9	0	0	1	1	0	0	1

Рис. 8.36. Код Хемминга.

Каждая контрольная группа кода Хемминга состоит из трех информационных и одного контрольного бита.

С помощью контрольного бита три информационных бита контрольной группы дополняются до четного числа 1-состояний.

Структура контрольной группы K_2 показана на рис. 8.37. Информационные биты — номер 5, номер 6 и номер 7. Контрольный бит — номер 4. При изображении десятичной цифры 0 информационные биты не имеют 1-состояний. Поэтому контрольный бит также не получает 1-состояния.

При передаче десятичной цифры 1 информационные биты содержат одно 1-состояние. Контрольный бит принимает в этом случае 1-состояние. Тогда количество 1-состояний контрольной группы станет четным. Такая же ситуация имеет место для представления десятичной цифры 2. При изображении десятичной цифры 3 информационные биты содержат два 1-состояния. Число 1-состояний четно. Контрольный бит принимает в этом случае 0-состояние. Для десятичных цифр от 4 до 9 контрольный бит всегда равен 1, если три информационных бита содержат нечетное число 1-состояний. Контрольный бит всегда равен 0, если три информационных бита содержат четное число 1-состояний.

Структура контрольной группы K_1 состоит из информационных битов номер 3, номер 6 и номер 7 и контрольного бита 2 (рис. 8.38). С помощью контрольного бита K_1 три информационных бита контрольной группы дополняются до четного числа 1-состояний. Алгоритм — как у контрольной группы K_2.

Номер бита		1	2	3	4	5	6	7
Вес					K_2	2^2	2^1	2^0
Десятичная цифра	0				0	0	0	0
	1				1	0	0	1
	2				1	0	1	0
	3				0	0	1	1
	4				1	1	0	0
	5				0	1	0	1
	6				0	1	1	0
	7				1	1	1	1
	8				0	0	0	0
	9				1	0	0	1

Контрольная группа K_2

Рис. 8.37. Структура контрольной группы K_2.

Номер бита		1	2	3	4	5	6	7
Вес			K_1	2^3			2^1	2^0
Десятичная цифра	0		0	0			0	0
	1		1	0			0	1
	2		1	0			1	0
	3		0	0			1	1
	4		0	0			0	0
	5		1	0			0	1
	6		1	0			1	0
	7		0	0			1	1
	8		1	1			0	0
	9		0	1			0	1

Контрольная группа K_1

Рис. 8.38. Структура контрольной группы K_1.

Третья контрольная группа — K_0. Она состоит из информационных битов номер 3, номер 5 и номер 7. Контрольный бит K_0 имеет номер 1 (рис. 8.39).

С помощью контрольного бита три информационных бита контрольной группы дополняются до четного числа 1-состояний. K_0 всегда равен 1, если три информационных бита содержат нечетное число 1-состояний.

Установление ошибки происходит посредством проверки четности контрольных групп.

В коде Хемминга каждая контрольная группа проверяется на четность.

Номер бита		1	2	3	4	5	6	7
Вес		K_0		2^3		2^2		2^0
Десятичная цифра	0	0		0		0		0
	1	1		0		0		1
	2	0		0		0		0
	3	1		0		0		1
	4	1		0		1		0
	5	0		0		1		1
	6	1		0		1		0
	7	0		0		1		1
	8	1		1		0		0
	9	0		1		0		1

Контрольная группа K_0

Рис. 8.39. Структура контрольной группы K_0.

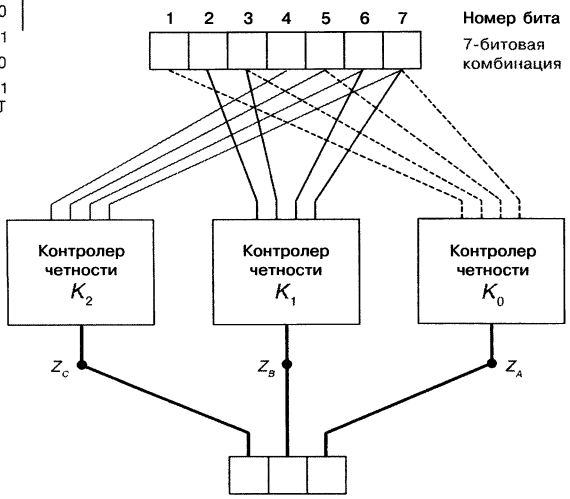


Рис. 8.40. Подключение контролеров четности.

Для проверки 7-битовой комбинации требуются три контролера четности. Они подключаются согласно рис. 8.40. При нечетности контрольной группы на выходе соответствующего контролера появляется 1-состояние. Это 1-состояние означает ошибку.

> 7-битовая комбинация кода Хемминга считается содержащей ошибку, если по крайней мере один контролер четности сообщает об ошибке.

Итак, распознавание ошибки происходит без проблем. Как происходит ее исправление? Если ошибка встречается в бите номер 1, то контролер четности группы K_0 сообщает об ошибке. Выход Z_A принимает состояние 1. Ошибка в бите номер 2 вызывает сообщение об ошибке контролера четности группы K_1 ($Z_B = 1$). В случае ошибки в бите номер 3 ошибку выдают контролеры четности K_0 и K_1. На рис. 8.41 показано, какие контролеры четности выдают сообщение об ошибке, и состояния выходов Z_A, Z_B и Z_C в случае ошибок в отдельных битах.

Ошибка в номере бита	Сообщение об ошибке от контролера четности	Выходные состояния		
		K_2 Z_C	K_1 Z_B	K_0 Z_A
1	K_0	0	0	1
2	K_1	0	1	0
3	K_0 и K_1	0	1	1
4	K_2	1	0	0
5	K_0 и K_2	1	0	1
6	K_1 и K_2	1	1	0
7	K_0, K_1 и K_2	1	1	1
		2^2	2^1	2^0

Рис. 8.41. Соответствие сообщений об ошибках выходным состояниям контролеров четности.

При рассмотрении рис. 8.41 ясно, что выходные состояния Z_A, Z_B и Z_C образуют двоичное число, которое соответствует номеру ошибочного бита. Выход Z_A соответствует 2^0, выход Z_B соответствует 2^1, выход Z_C соответствует 2^2.

> Выходные состояния контролеров четности в коде Хемминга указывают номер ошибочного бита.

Таким образом, однозначно идентифицируется ошибочный бит. Теперь его можно исправлять. Коррекция происходит самостоятельно с помощью цифровой микросхемы, которая инвертирует бит, помеченный как ошибочный. Больше делать ничего не надо, так как если ошибочный бит равен 1, то его истинное значение 0. Если ошибочный бит равен 0, то его истинное значение 1.

В схемах, которые работают с кодом Хемминга, каждая 7-битовая комбинация проверяется в определенных точках схемы и в случае необходимости исправляется. Такая проверка и коррекция целесообразна прежде всего после передачи информации через длинные дорожки на плате, так как длинные дорожки больше подвержены воздействию помех.

Контрольный тест

1. Чем отличаются понятия бинарный и двоичный?
2. Преобразуйте двоичные числа таблицы 8.42 в десятичные.

Десятичная цифра	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
								1	1	0	0	1	0
						1	1	0	1	0	1	1	1
						1	0	1	0	1	1	0	0
				1	0	1	0	1	1	0	0	0	0
				1	1	1	0	0	0	1	1	0	1
?			1	1	1	0	0	0	1	1	1	0	0
		1	1	0	0	1	1	0	0	1	1	0	0
		1	0	0	1	0	1	1	1	0	1	1	1
	1	0	1	1	1	1	0	1	0	0	1	0	0
	1	1	0	0	0	1	1	1	0	1	1	0	1
	1	0	1	0	1	1	0	0	0	0	1	1	1
	1	1	1	1	0	0	0	1	1	0	1	0	0

Рис. 8.42. Преобразование двоичных чисел в десятичные.

3. Преобразуйте следующие десятичные числа в двоичные:
   58
   512
   1298
   1983

20000
17750
2730
9990
11000
32000.

4. Преобразуйте следующие вещественные двоичные числа в десятичные:
а) 110110,101;
б) 100101,1101;
в) 1010,11101;
г) 0,10101;
д) 0,011101.

5. Сложение в двоичной системе счисления. Решите следующие примеры:

а) 1101 + 100; ?
б) 111 101 + 1001; ?
в) 11 011 + 100 100; ?
г) 110 001 + 11 101; ?
д) 111 100 + 1 100 111; ?
е) 110 011 + 1 010 100; ?
ж) 1000,11 + 111,11; ?
з) 1100,11 + 111,01. ?

Проверьте результат переводом в десятичную систему счисления.

6. Вычитание в двоичной системе счисления. Решите следующие примеры в дополнительном коде:

а) 1101 − 100; ?
б) 111 101 − 1001; ?
в) 11 011 − 1111; ?
г) 1 001 100 − 101 010; ?
д) 100 111 − 10 111; ?
е) 110 011 − 11 010; ?
ж) 111 000 − 10 011; ?
з) 1101 − 10 100. ?

7. Преобразуйте следующие десятичные числа в двоично-десятичный код:
а) 10 941;
б) 3 890;
в) 7 863;
г) 98 001;
д) 7 989.

8. Сложение в двоично-десятичном коде:

а) 0100 + 0011; ?
б) 1000 + 0110; ?
в) 0111 + 1001; ?
г) 0011 + 110; ?
д) 1001 + 1000; ?
е) 1001 + 0001; ?
ж) 0110 + 0110; ?
з) 1001 + 0110. ?

9. Сложение в двоично-десятичном коде:

а) 1000 − 0111; ?
б) 1001 − 1000; ?
в) 0111 − 0110; ?
г) 1001 − 0111; ?
д) 0111 − 0011; ?
е) 0111 − 1001; ?
ж) 1000 − 0011; ?
з) 0011 − 1000. ?

10. Переведите шестнадцатеричные числа в десятичные и двоичные
а) *AB*1;
б) 87*F*2;
в) *E*605;
г) *BCD*4;
д) 12*B*31;
е) *BA*1*A*;
ж) 31 459;
з) 1*A*1*B*.

11. Переведите десятичные числа в шестнадцатеричные и двоичные
а) 100;
б) 259;
в) 1 020;
г) 1 983;
д) 10 000;
е) 126;
ж) 18 020;
з) 999.

12. Перекодируйте числа в таблице на рис. 8.43. В каждом свободном поле нужно записать результат. Десятичное число 2560, например, должно преобразоваться в двоичное число, в шестнадцатеричное число, в восьмеричное число и в *BCD*-число.

Десятичное число	Двоичное число	Шестнадцатеричное число	Восьмеричное число	*BCD*-число				
2560								
	100 1111 0110							
		*AF*36						
			1772					
				11	1001	0111	0001	1000
		1*A*2*BC*						

Рис. 8.43. Задача на кодирование.

13. Объясните структуру кода с избытком 3.
14. Что понимают под избыточностью?
15. Как построен одношаговый код?
16. Назовите три кода с распознаванием ошибки и объясните на примере, как производится распознавание ошибки.

17. Что такое проверка четности?

18. Объясните, как представляются отрицательные числа в двоичной системе счисления.

19. Чем различаются код с распознаванием ошибки и код с исправлением ошибки?

20. Как устроен код Хемминга и как происходит исправление ошибок?

ГЛАВА 9

ПРЕОБРАЗОВАТЕЛИ КОДОВ И УРОВНЕЙ

9.1. Преобразователи кодов

Преобразователи кодов служат для преобразования информации из одного кода в другой.

9.1.1. Расчет преобразователей кода

Количество входов преобразователя кода всегда соответствует количеству битов преобразуемого кода. Для 4-битового кода (тетрадного кода) необходимы 4 входа. Количество выходов соответствует количеству битов кода, в который нужно преобразовать. Если производится преобразование из кода Айкена в код Хемминга, то необходимы 4 входа и 7 выходов (рис. 9.1).

Преобразователи кодов рассчитываются по правилам синтеза схем.

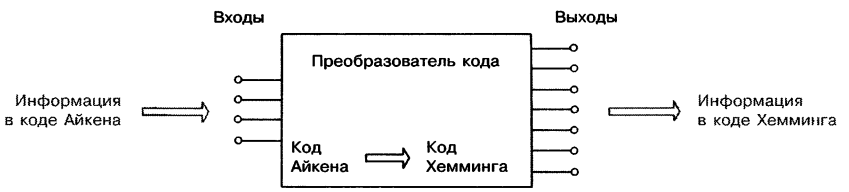


Рис. 9.1. Преобразователь из кода Айкена в код Хемминга.

Желаемый преобразователь кода нужно представить в виде таблицы истинности. По таблице истинности следует записать для каждого выхода нормальную форму ИЛИ. Затем ее следует максимально упростить. По упрощенным уравнениям синтезируют схему.

Пример

Постройте преобразователь из кода Айкена в двоично-десятичный код.

Нужна схема с 4 входами и 4 выходами согласно рис. 9.2. Операции, которые должна производить эта схема, представлены в таблице истинности на рис. 9.3.

Для выходов Q_1, Q_2, Q_3 и Q_4 получаются следующие нормальные формы ИЛИ:

$$Q_1 = \left(A \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}\right) \vee \left(A \wedge B \wedge \overline{C} \wedge \overline{D}\right) \vee \left(A \wedge B \wedge \overline{C} \wedge D\right) \vee$$
$$\vee \left(A \wedge \overline{B} \wedge C \wedge D\right) \vee \left(A \wedge B \wedge C \wedge D\right);$$

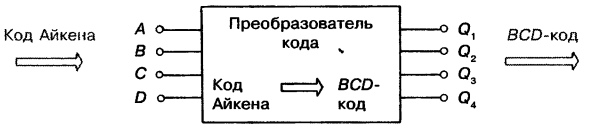


Рис. 9.2. Преобразователь из кода Айкена в *BCD*-код (двоично-десятичный).

	Входы Код Айкена				Выходы *BCD*-код			
	D	*C*	*B*	*A*	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	1	0	1	1	0	1	0	1
6	1	1	0	0	0	1	1	0
7	1	1	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0
9	1	1	1	1	1	0	0	1

Рис. 9.3. Таблица истинности преобразователя из кода Айкена в *BCD*-код (двоично-десятичный).

$$Q_2 = \left(\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D}\right) \vee \left(A \wedge B \wedge \bar{C} \wedge \bar{D}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C \wedge D\right) \vee \left(A \wedge \bar{B} \wedge C \wedge D\right);$$

$$Q_3 = \left(\bar{A} \wedge \bar{B} \wedge C \wedge \bar{D}\right) \vee \left(A \wedge B \wedge \bar{C} \wedge D\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C \wedge D\right) \vee \left(A \wedge \bar{B} \wedge C \wedge D\right);$$

$$Q_4 = \left(\bar{A} \wedge B \wedge C \wedge D\right) \vee \left(A \wedge B \wedge C \wedge D\right).$$

Нормальные формы ИЛИ упрощаются с помощью диаграмм Карно (рис. 9.4).

$$Q_1 = \left(A \wedge B \wedge \bar{C}\right) \vee \left(A \wedge C \wedge D\right) \vee \left(A \wedge \bar{C} \wedge \bar{D}\right);$$

$$Q_2 = \left(B \wedge \bar{C} \wedge \bar{D}\right) \vee \left(\bar{B} \wedge C \wedge D\right);$$

$$Q_3 = \left(\bar{A} \wedge \bar{B} \wedge C\right) \vee \left(\bar{B} \wedge C \wedge D\right) \vee \left(A \wedge B \wedge \bar{C} \wedge D\right);$$

$$Q_4 = B \wedge C \wedge D.$$

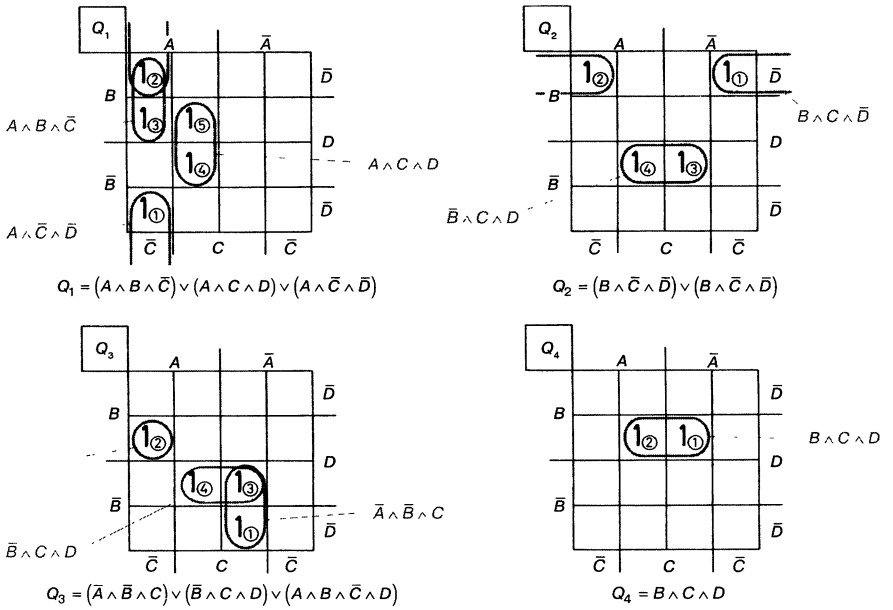


Рис. 9.4. Упрощение нормальной формы ИЛИ.

По упрощенным уравнениям можно строить схему. Если в наличии имеются только элементы И-НЕ, то уравнения нужно соответствующим образом пересчитать. На рис. 9.5 показана схема в базисе И-НЕ.

$$Q_1 = \overline{\overline{A \wedge B \wedge \bar{C}} \wedge \overline{A \wedge C \wedge D} \wedge \overline{A \wedge \bar{C} \wedge \bar{D}}};$$

$$Q_2 = \overline{\overline{B \wedge \bar{C} \wedge \bar{D}} \wedge \overline{\bar{B} \wedge C \wedge D}};$$

$$Q_3 = \overline{\overline{\bar{A} \wedge \bar{B} \wedge C} \wedge \overline{\bar{B} \wedge C \wedge D} \wedge \overline{A \wedge B \wedge \bar{C} \wedge D}};$$

$$Q_4 = \overline{\overline{B \wedge C \wedge D}}.$$

По этому алгоритму могут рассчитываться преобразователи кодов для любого типа преобразования. Для преобразований между часто применяемыми кодами в распоряжении имеются готовые микросхемы.

9.1.2. Преобразователи десятичного кода в *BCD*-код

Во многих цифровых схемах — прежде всего в вычислителях — вводятся десятичные цифры. Требуется преобразование из десятичного кода в двоичный или в двоично-десятичный код.

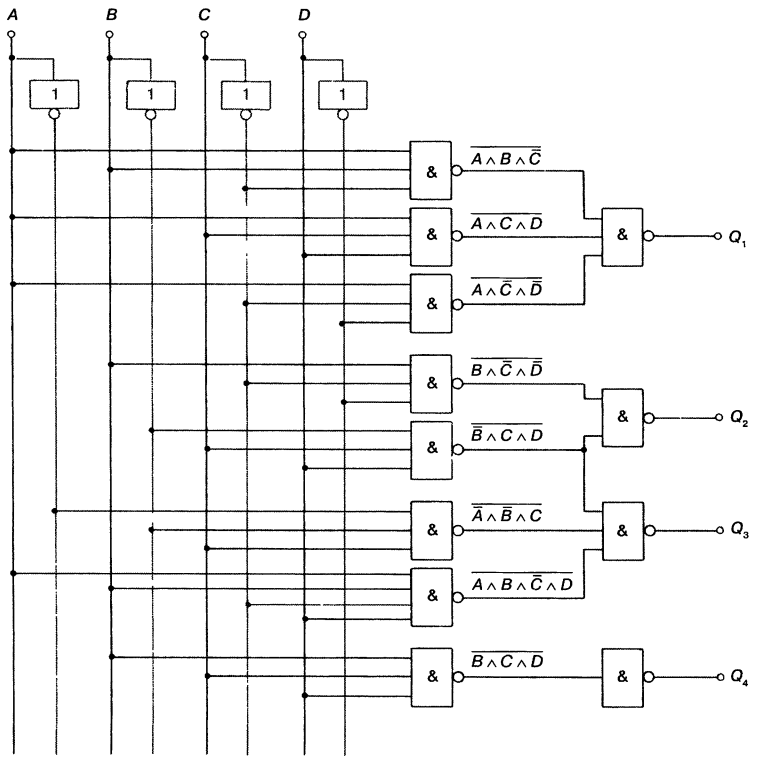


Рис. 9.5. Преобразователь из кода Айкена в *BCD*-код (двоично-десятичный).

Преобразователи из десятичного кода в *BCD*-код выполняют преобразование десятичных цифр в двоичные числа.

Десятичный код является кодом 1 из 10. Преобразователи кодов, преобразующие десятичные цифры в *BCD*-код, должны иметь 10 входов и 4 выхода. Так как десятичная цифра 0 в *BCD*-коде выражается как 0000, то вход для десятичной цифры 0 не нужен. Итак, необходимы только 9 входов (рис. 9.6).

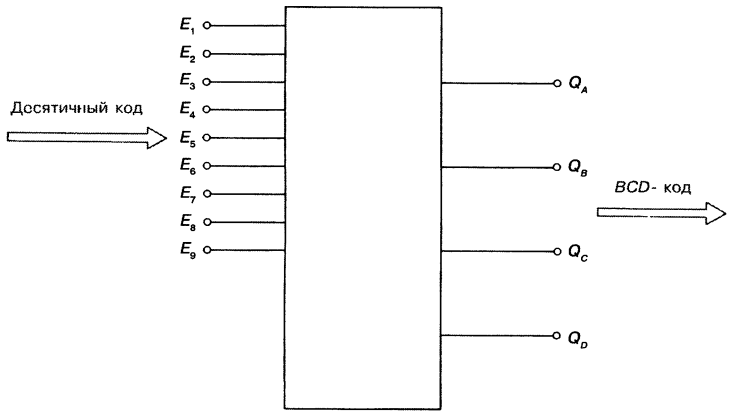


Рис. 9.6. Преобразователь из десятичного кода в *BCD*-код (двоично-десятичный).

Схема может быть рассчитана по алгоритму разд. 9.1.1. Однако ее можно синтезировать исходя из простых рассуждений. Таблица истинности преобразователя кодов, преобразующего десятичные цифры в *BCD*-код, представлена на рис. 9.7. Каждый 1-сигнал на одном из входов должен вызывать 1-сигнал на определенных выходах.

Входной сигнал должен распределяться на соответствующие выходы. Для этой задачи можно применить матричный переключатель (рис. 9.8). Каждый выход снабжается 1-сигналом через элемент ИЛИ.

Преобразователи кодов, преобразующие десятичные цифры в *BCD*-код, производятся в виде микросхем. В ТТЛ-исполнении это, например, микросхемы 74147 и 84147.

Десятичная цифра	Входы десятичный код (код 1 из 10)									Выходы *BCD*-код			
										2^3	2^2	2^1	2^0
	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	Q_D	Q_C	Q_B	Q_A
1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	1	1	0	0	1

Рис. 9.7. Таблица истинности преобразователя из десятичного кода в *BCD*-код (двоично-десятичный).

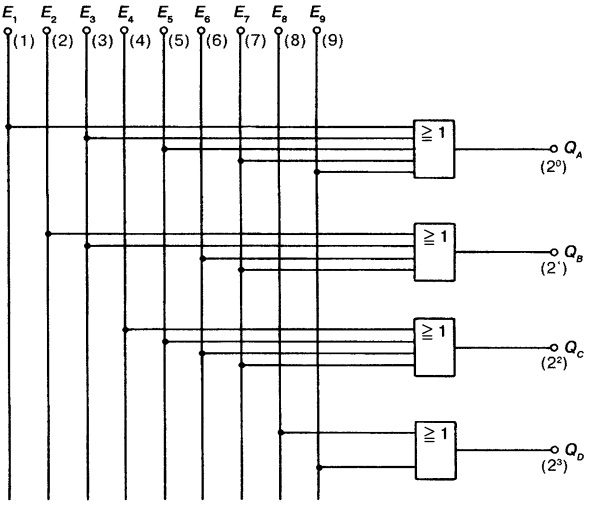


Рис. 9.8. Схема преобразователя из десятичного кода в *BCD*-код (двоично-десятичный).

9.1.3. Преобразователи *BCD*-кода в десятичный код

Для преобразования *BCD*-чисел в десятичные нужны преобразователи из *BCD*-кода в десятичный. Такой преобразователь кода должен иметь 4 входа для приема *BCD*-цифр. Для каждой десятичной цифры требуется свой выход. Единичный сигнал на выходе десятичной цифры 3 означает, что должна быть отображена цифра 3. Такой вид изображения цифровой информации применяется в цифровых указательных трубках (см. Бойт, Электроника, ч. 2). Однако экономически целесообразным является визуализация информации на 7-сегментных индикаторах (см. разд. 9.1.9).

> Преобразователи из *BCD*-кода в десятичный преобразуют *BCD*-числа в десятичные.

Расчет преобразователя из *BCD*-кода в десятичный код очень прост. Как следует из таблицы истинности на рис. 9.9, для каждого выхода получается только одна полная конъюнкция.

Десятичная цифра	Входы *BCD*-код 2^3 *D*	2^2 *C*	2^1 *B*	2^0 *A*	Выходы десятичный код (код 1 из 10) Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	1	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1

Рис. 9.9. Таблица истинности преобразователя из *BCD*-кода в десятичный.

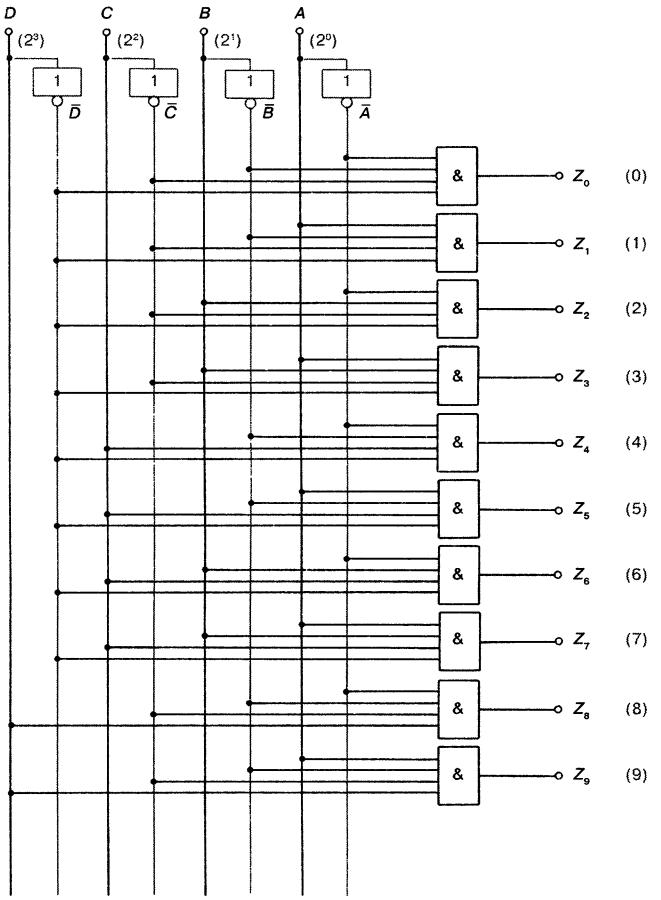


Рис. 9.10. Принципиальная схема преобразователя из *BCD*-кода в десятичный.

Логические параметры

BCD-входы				Десятичные выходы									
D	*C*	*B*	*A*	0	1	2	3	4	5	6	7	8	9
L	*L*	*L*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
L	*L*	*L*	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
L	*L*	*H*	*L*	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
L	*L*	*H*	*H*	*H*	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*
L	*H*	*L*	*L*	*H*	*H*	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*
L	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*H*	*H*	*H*	*H*
L	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*H*	*H*	*H*
L	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*H*	*H*
H	*L*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*H*
H	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*
H	*L*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
H	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
H	*H*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
H	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
H	*H*	*H*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
H	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*

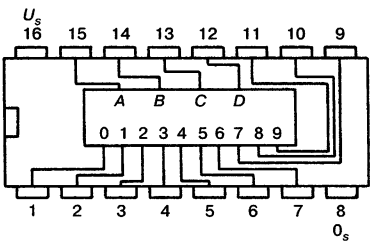


Рис. 9.11. Цоколевка, параметры и таблица уровней микросхемы FLH281-7442A (Siemens).

FLH281-7442A
FLH285-8442A

Модуль FLH281/285 преобразует *BCD*-код. Входы можно напрямую подключать к выходам всех десятичных счетчиков, при этом *A* соединяется с Q_A, *B* с Q_B, *C* с Q_C и *D* с Q_D.

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S	U_S = 4,75 В	4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}		2,0			В
Входное напряжение *L*-уровня	U_{IL}				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА			1,5	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В, U_{IH} = 2 В, U_{IL} = 0,8 В, $-I_{QH}$ = 800 мкА	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В, U_{IH} = 2 В, U_{IL} = 0,8 В, I_{QL} = 16 мА		0,2	0,4	В
Входной ток на канал	I_I	U_S = 5,25 В, U_I = 5,5 В			1	мА
H-входной ток на канал	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В	18		55	мА
L-входной ток на канал	$-I_{IL}$	U_{IL} = 0,4 В, U_S = 5,25 В			40	мкА
Ток короткого замыкания на канал выхода	$-I_Q$	U_S = 5,25 В			1,6	мА
Ток питания	I_S	U_S = 5,25 В		28	56	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						
Время прохождения сигнала (быстродействие)						
к выходу 0	t_{PHL}	R_L = 400 Ом C_L = 15 пФ		14	25	В
к выходам от 1 до 9	t_{PHL}			17	30	мкА
к выходу 0	t_{PLH}			10	25	мА
к выходам от 1 до 9	t_{PLH}			17	30	мА
Логические параметры						мА
Коэффициент разветвления по выходу на канал						мА
H-сигнал	F_{QH}				20	
L-сигнал	F_{QL}				10	
Входной нагрузочный коэффициент на канал	F_I				1	

Входные переменные A, B, C и D должны быть доступны в прямой и инверсной формах. Для реализации полных конъюнкций необходимы 10 элементов И, каждый с четырьмя входами (рис. 9.10):

$$Z_0 = \overline{A} \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}; \qquad Z_5 = A \wedge \overline{B} \wedge C \wedge \overline{D};$$

$$Z_1 = A \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}; \qquad Z_6 = \overline{A} \wedge B \wedge C \wedge \overline{D};$$

$$Z_2 = \overline{A} \wedge B \wedge \overline{C} \wedge \overline{D}; \qquad Z_7 = A \wedge B \wedge C \wedge \overline{D};$$

$$Z_3 = A \wedge B \wedge \overline{C} \wedge \overline{D}; \qquad Z_8 = \overline{A} \wedge \overline{B} \wedge \overline{C} \wedge D;$$

$$Z_4 = \overline{A} \wedge \overline{B} \wedge C \wedge \overline{D}; \qquad Z_9 = A \wedge \overline{B} \wedge \overline{C} \wedge D.$$

Схема на рис. 9.10 может быть получена путем логического рассуждения. Каждой 4-битовой комбинации *BCD*-кода должен однозначно соответствовать один выход. Такое соответствие можно получить, производя над битами операции И или НЕ.

Преобразователи кодов, преобразующие *BCD*-код в десятичные цифры, производятся в виде микросхем. В ТТЛ-исполнении это, например, микросхема FLH281-7442A.

На рис. 9.11 изображена цоколевка микросхемы вместе с ее параметрами и таблицей уровней. При наличии десятичной цифры на выходе на нем действует L-уровень. Это целесообразно, если цифровые индикаторы управляются через драйверы.

Если на входы A, B, C и D схемы FLH281-7442A подаются 4-битовые комбинации, которые не принадлежат к двоично-десятичному коду (так называемые псевдотетрады), то такие комбинации не приводят к выводу цифры. Таблица уровней на рис. 9.11 показывает, что псевдотетрады не выводятся.

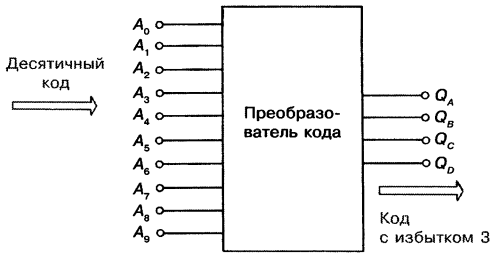
9.1.4. Преобразователи десятичного кода в код с избытком 3

Преобразователи десятичного кода в код с избытком 3 кода преобразуют десятичные цифры в 4-битовые комбинации кода с избытком 3.

Преобразователь кода может быть рассчитан по методике разд. 9.1.1. Однако проще рассчитать его по принципу матричного переключателя. 1-состояния на десятичных входах «распределяются» на выходы с избытком 3 посредством элементов ИЛИ (рис. 9.12).

9.1.5. Преобразователи кода с избытком 3 в десятичный код

Преобразователи кода с избытком 3 в десятичный код преобразуют 4-битовые комбинации кода с избытком 3 в десятичные цифры.



Десятичная цифра	Q_D	Q_C	Q_B	Q_A
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

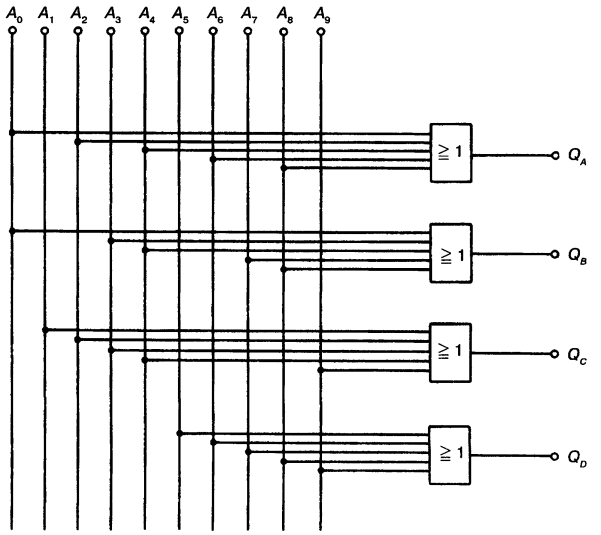


Рис. 9.12. Преобразователи десятичного кода в код с избытком 3.

Расчет преобразователя кода не требуется. Он может быть построен как преобразователь *BCD*-кода в десятичный — с соответствующей для кода с избытком 3 разводкой. Необходимы входные переменные в прямой и инверсных формах. 4-битовые комбинации кода с избытком 3 приводятся в соответствие десятичному коду посредством элементов И (рис. 9.13).

9.1.6. Преобразователи десятичного кода в 7-сегментный

Десятичные цифры представляются в основном посредством 7-сегментных индикаторов. Эти индикаторы построены на светодиодных или жидкокристаллических сегментах (см. Бойт, Электроника, ч. 2).

Для управления 7-сегментными индикаторами нужен особенный код, который называется **7-сегментный код**. Этот код указывает, какие сегменты должны использоваться для изображения отдельных десятичных цифр. Например, для изображения десятичной цифры 3 должны использоваться сегменты *a*, *b*, *c*, *d* и *g* (рис. 9.14). Для отображения десятичной цифры 8 необходимы все сегменты. На рис. 9.15 представлен 7-сегментный код.

Преобразователи десятичного кода в семисегментный преобразуют десятичный код в семисегментный.

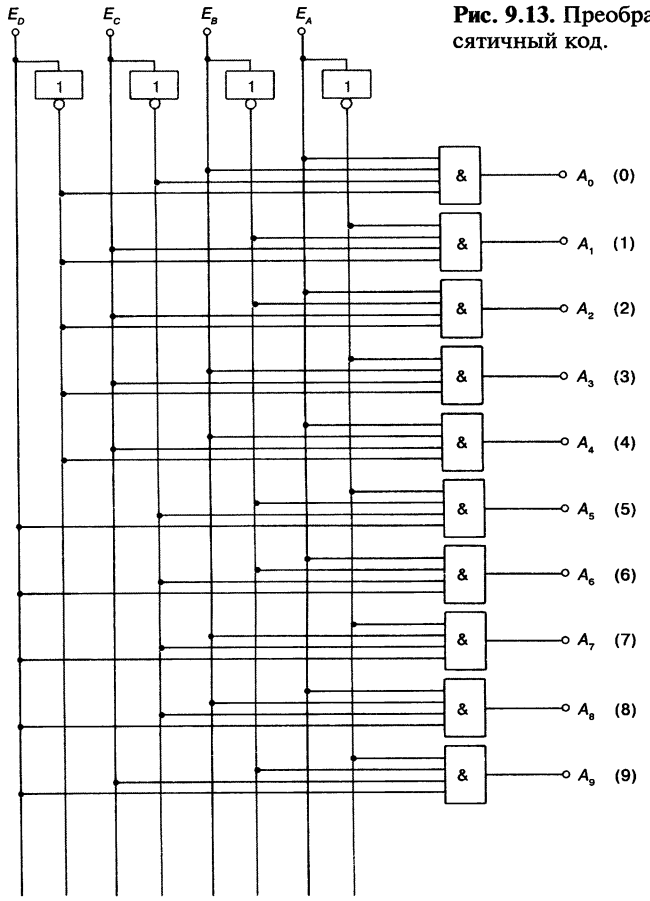


Рис. 9.13. Преобразователь кода с избытком 3 в десятичный код.

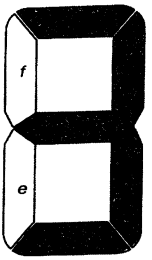


Рис. 9.14. 7-сегментный индикатор.

Десятичная цифра	7-сегментный код						
	a	*b*	*c*	*d*	*e*	*f*	*g*
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	0	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	0	0	1	1

Рис. 9.15. 7-сегментный код.

Преобразователи десятичного кода нет необходимости рассчитывать. Они могут быть построены по принципу матричного переключателя. 1-состояния на десятичных входах «распределяются» на 7-сегментные выходы посредством элементов ИЛИ (рис. 9.16).

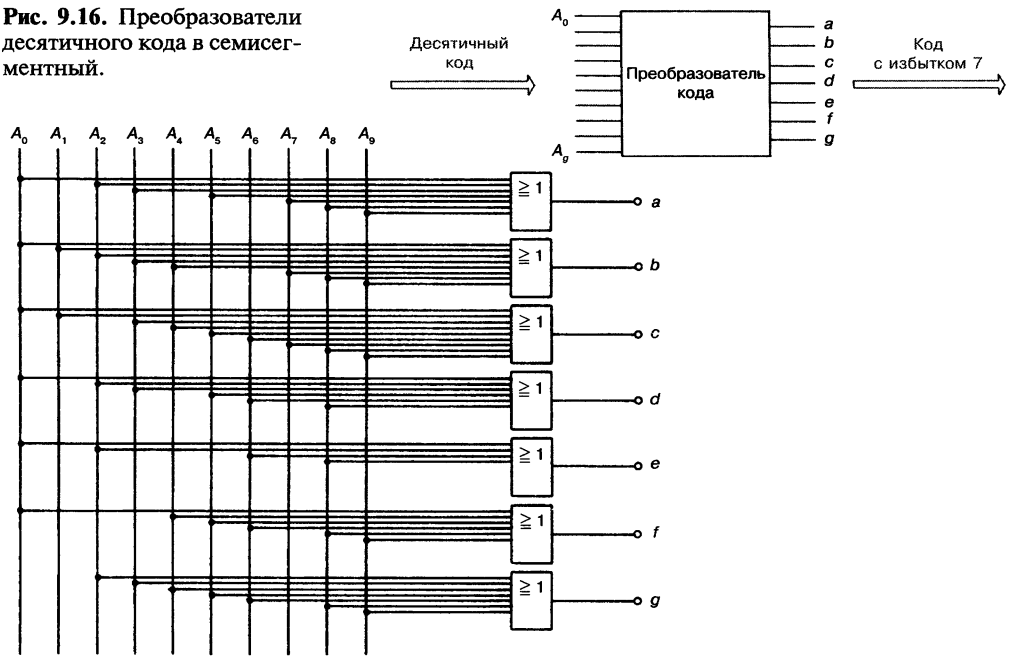
9.1.7. Преобразователи *BCD*-кода в семисегментный

BCD-код (двоично-десятичный код) применяется очень часто. Так же часто закодированная в двоично-десятичном коде информация должна отображаться на 7-сегментном дисплее. Поэтому коды, осуществляющие преобразование двоично-десятичного кода в 7-сегментный код, имеют большое значение.

> Преобразователи двоично-десятичного кода в 7-сегментный преобразуют двоично-десятичный код в 7-сегментный.

Такой преобразователь кода можно реализовать двумя видами из рассмотренных до сих пор преобразователей кода. Если соединить *BCD*-десятичный-преобразователь кода и десятично-7-сегментный преобразователи кода вместе, то получится преобразователь кода, который преобразует дво-

Рис. 9.16. Преобразователи десятичного кода в семисегментный.



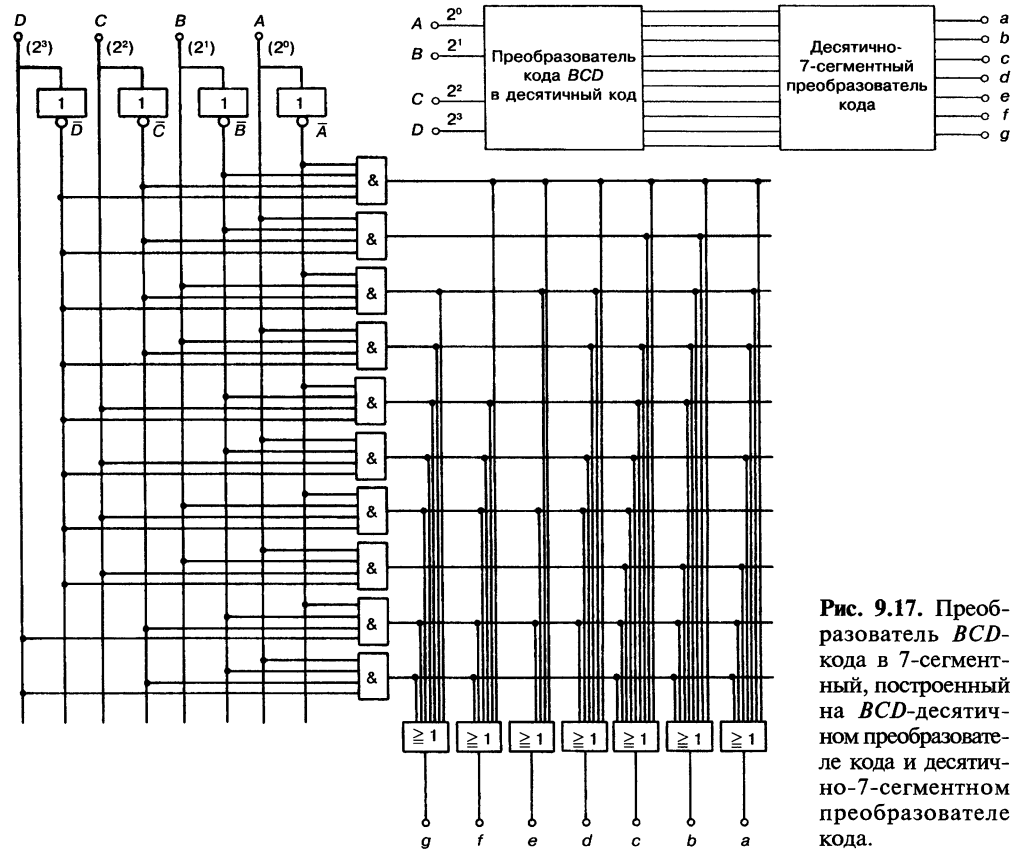


Рис. 9.17. Преобразователь *BCD*-кода в 7-сегментный, построенный на *BCD*-десятичном преобразователе кода и десятично-7-сегментном преобразователе кода.

Десятичная цифра	BCD-код				7-сегментный код						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Рис. 9.18. Таблица истинности *BCD*-7-сегментного преобразователя.

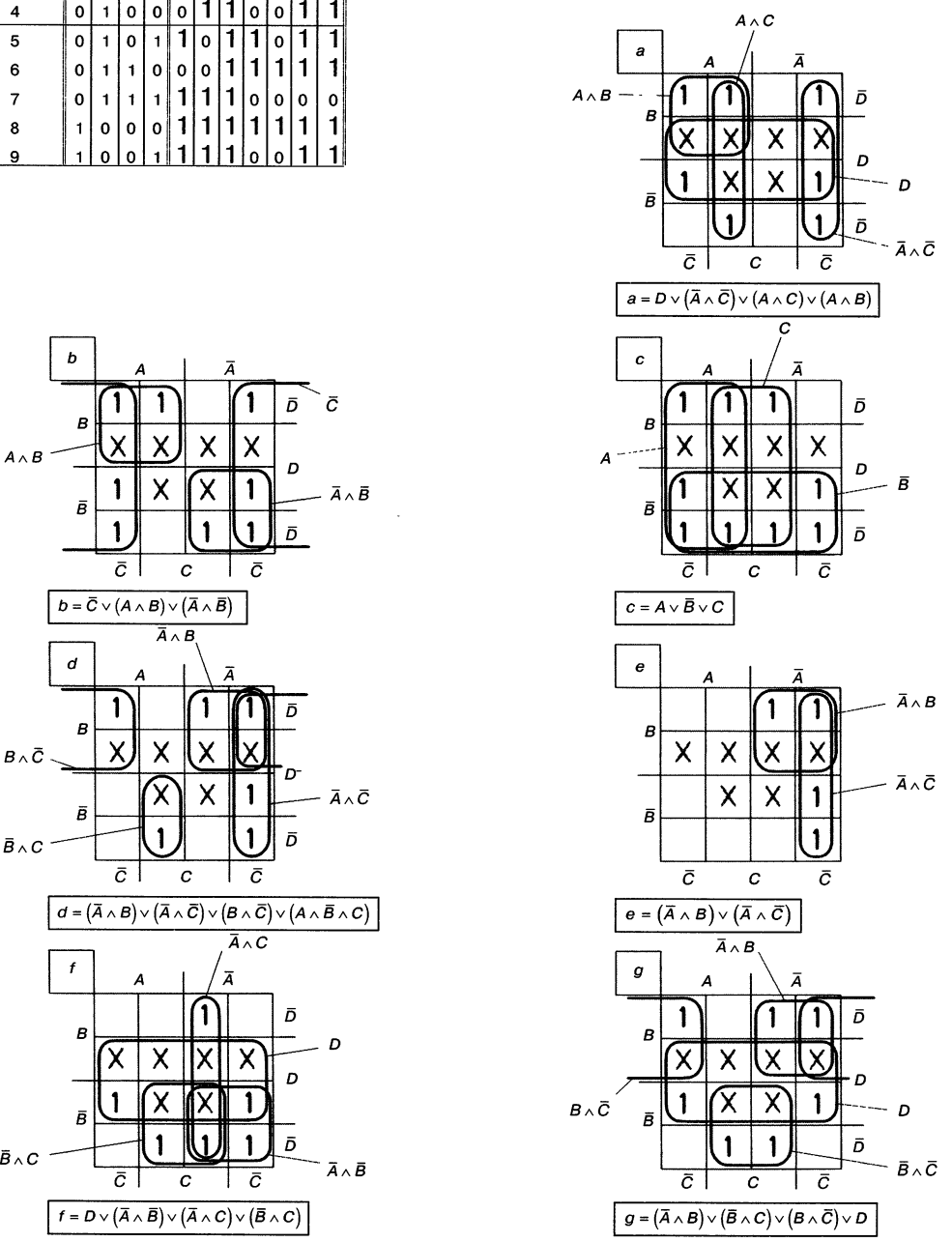


Рис. 9.19. Упрощение нормальной формы ИЛИ выходов *a*, *b*, *c*, *d*, *e*, *f* и *g* 7-сегментного преобразователя.

ично-десятичный код в 7-сегментный код. Схема подобного преобразователя кода показана на рис. 9.17.

Расчет *BCD*-7-сегментного преобразователя кода приводит, тем не менее, к более простой схеме. Таблица истинности искомого преобразователя кода показана на рис. 9.18. Для каждого из выходов *a*, *b*, *c*, *d*, *e*, *f* и *g* можно составить нормальную форму ИЛИ. Нормальные формы ИЛИ упрощаются с помощью карт Карно (рис. 9.19).

BCD-псевдотетрады участвовать в расчете не должны. Поэтому ячейки с псевдотетрадами помечаются крестиками на диаграммах (картах) Карно. Эти ячейки могут рассматриваться как содержащие 0, так и 1, по желанию разработчика. Образование групп по этой причине становится более легким.

В соответствии с найденными для входов *a*, *b*, *c*, *d*, *e*, *f* и *g* уравнениями (рис. 9.19) может строиться схема. Она представлена на рис. 9.20.

BCD-7-сегментные преобразователи кода выпускаются в виде интегральных микросхем. Распространенная микросхема TTL-семейства называется FLH551-7448. Таблица параметров, цоколевка и таблица уровней этой схе-

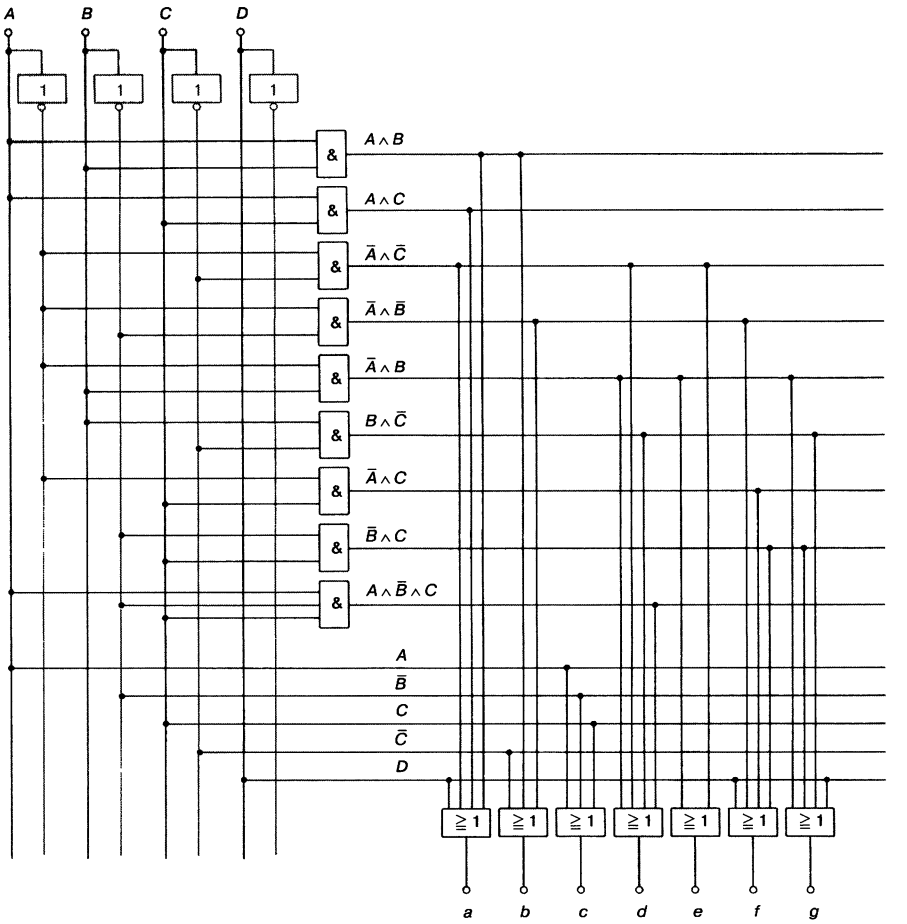


Рис. 9.20. Схема преобразователя *BCD*-кода в 7-сегментный.

BCD-7-сегментноый преобразователь FLH551-7448, FLH555-8448

Модуль FLH 551/555 принимает 4-битовые комбинации, декодирует их в зависимости от состояния входов (*BI*, *RBI*, *LT*) и выводит на выходы *a*, *b*, *c*, *d*, *e*, *f*, *g* 7-сегментный код (ТТЛ-уровень, однотактовые выходы с коллекторным сопротивлением).

Вход подавления нулей *RBI* активным низким уровнем гасит высвеченные нули. Для многозначных чисел путем соединения с выходом подавления нулей *RBQ* (внутренне соединенным с входом *BI*) возможно автоматическое гашение нуля в нескольких декадах. Через вход гашения *BI* происходит общее гашение сегментов, через вход контроля свечения *LT* производится контроль световых сегментов (свечение всех сегментов индикатора).

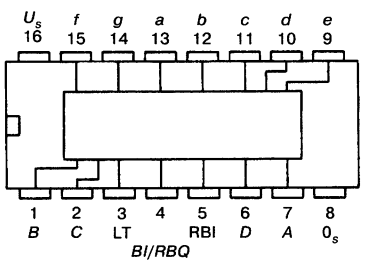
Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S	U_S = 4,75 В	4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}		2,0			В
Входное напряжение *L*-уровня	U_{IL}				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА			1,5	В
Выходное напряжение *H*-уровня						В
от *a* до *g*	U_{QH}	U_S = 5,25 В, $-I_{QH}$ = 400 мкА	2,4	4,2		В
на *BI*/*RBQ*	U_{QH}	U_S = 5,25 В, $-I_{QH}$ = 200 мкА	2,4	3,7		В
Выходное напряжение *L*-уровня						
от *a* до *g*	U_{QL}	U_S = 4,75 В, I_{QL} = 6,4 мА		0,27	0,4	В
на *BI*/*RBQ*	U_{QL}	U_S = 4,75 В, I_{QL} = 8 мА		0,27	0,4	В
H-входной ток на канал	I_I	U_S = 5,25 В, U_I = 5,5 В			1	мА
Прочие каналы						
кроме *BI*/*RBQ*	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В			40	мкА
L-входной ток на канал		U_S = 5,25 В, U_{IL} = 0,4 В				
на *BI*/*RBQ*	I_{IL}				4	мкА
Прочие каналы	I_{IL}				1,6	мкА
Ток короткого замыкания						
на *BI*/*RBQ*	$-I_Q$	U_S = 5,25 В			4	мА
Ток питания	I_S	U_S = 5,25 В		53	90	мА
Вид сверху				53	90	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Время прохождения сигнала (быстродействие)						В
к выходу 0	t_{PHL}	C_L = 15 пФ R_L = 1 кОм			100	нс
к выходам от 1 до 9	t_{PHL}				100	нс
к выходу 0	t_{PLH}				100	нс
к выходам от 1 до 9	t_{PHL}				100	нс
Логические параметры						
Коэффициент разветвления по выходу						
на *BI/RBQ*	F_Q				5	
H-сигнал от *a* до *g*	F_{QH}				10	
L-сигнал	F_{QL}				4	
Входной нагрузочный коэффициент						
на *BI/RBQ*	F_I				2,6	
прочие каналы	F_I				1	

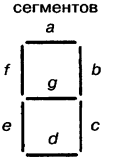
Логические соотношения

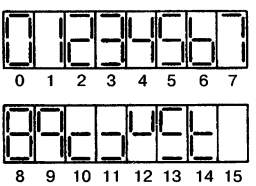
Функция	*LT*	*RBI*	*D*	*C*	*B*	*A*	*BI/RBQ*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0[1]	*H*	*H*	*L*	*L*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*
1	*H*	*X*	*L*	*L*	*L*	*H*	*H*	*L*	*H*	*H*	*L*	*L*	*L*	*L*
2	*H*	*X*	*L*	*L*	*H*	*L*	*H*	*H*	*H*	*L*	*H*	*H*	*L*	*H*
3	*H*	*X*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*L*	*H*
4	*H*	*X*	*L*	*H*	*L*	*L*	*H*	*L*	*H*	*H*	*L*	*L*	*H*	*H*
5	*H*	*X*	*L*	*H*	*L*	*H*	*H*	*H*	*L*	*H*	*H*	*L*	*H*	*H*
6	*H*	*X*	*L*	*H*	*H*	*L*	*H*	*L*	*L*	*H*	*H*	*H*	*H*	*H*
7	*H*	*X*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*L*	*L*	*L*	*L*
8	*H*	*X*	*H*	*L*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*
9	*H*	*X*	*H*	*L*	*L*	*H*	*H*	*H*	*H*	*H*	*L*	*L*	*H*	*H*
10	*H*	*X*	*H*	*L*	*H*	*L*	*H*	*L*	*L*	*L*	*H*	*H*	*L*	*H*
11	*H*	*X*	*H*	*L*	*H*	*H*	*H*	*L*	*L*	*H*	*H*	*L*	*L*	*H*
12	*H*	*X*	*H*	*H*	*L*	*L*	*H*	*L*	*H*	*L*	*L*	*L*	*H*	*H*
13	*H*	*X*	*H*	*H*	*L*	*H*	*H*	*H*	*L*	*L*	*H*	*L*	*H*	*H*
14	*H*	*X*	*H*	*H*	*H*	*L*	*H*	*L*	*L*	*L*	*H*	*H*	*H*	*H*
15	*H*	*X*	*H*	*H*	*H*	*H*	*H*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
BI[2]	*X*	*X*	*X*	*X*	*X*	*X*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
RI[3]	*H*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
LT[4]	*L*	*X*	*X*	*X*	*X*	*X*	*H*	*H*	*H*	*H*	*H*	*H*	*H*	*H*

Цоколевка
Вид сверху



Обозначения сегментов





Представление отображаемой функции

Замечания:

X ≙ *H* может быть и *H*- и *L*-сигналом

[1] При индикации нуля на входе *RBI* должен быть высокий уровень.

[2] Если ко входу *BI* приложен низкий уровень, то на сегментные выходы поступает также сигнал низкого уровня, независимо от состояния входов.

[3] Если ко входу *RBI* приложен низкий уровень, то на сегментные выходы и выход *RBQ* поступают также сигналы низкого уровня при условии, что ко входам *A*, *B*, *C*, *D* приложен низкий уровень.

[4] Если ко входу *LT* приложен низкий *L*-уровень, то на сегментные выходы поступает сигнал высокого *H*-уровня (яркое свечение) при условии, что на *BI/RBQ* действует *H*-сигнал, независимо от состояния входов *A*, *B*, *C*, *D*, *RBI*.

Рис. 9.21. Таблица параметров, цоколевка и таблица уровней схемы FLH551-7448 (Siemens).

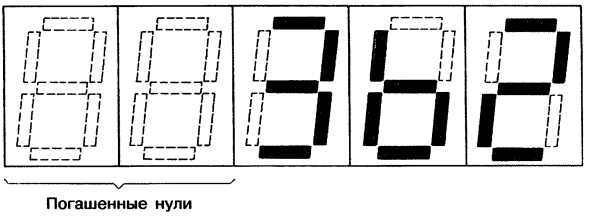


Рис. 9.22. Пятизначный 7-сегментный индикатор с гашением нулей.

мы воспроизведены на рис. 9.21. Схема предоставляет возможность подавления нулей и контроль свечения. Для многопозиционных индикаторов все нули слева от значащих цифр могут быть выключены (рис. 9.22). Нежелательные цифры могут быть затемнены.

9.2. Преобразователи уровней

9.2.1. Введение

Электронные микросхемы различных семейств могут работать с различными уровнями напряжения. Если нужно соединить схемы различных семейств друг с другом, то прежде всего следует проверить их совместимость. Под **совместимостью** понимается возможность соединения выходов микросхемы одного семейства со входом микросхемы другого семейства.

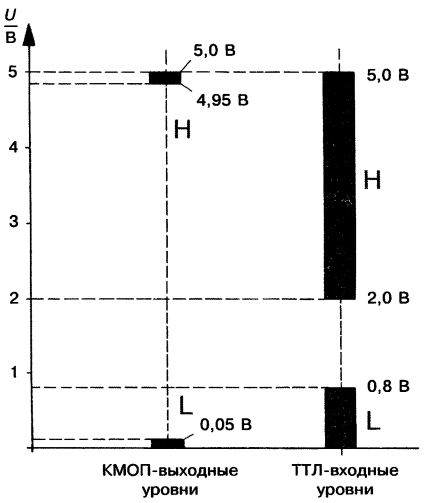


Рис. 9.23. Диаграмма уровней.

Между многими семействами действует ограниченная совместимость. Выходы КМОП-элементов могут, например, быть ТТЛ-совместимы, а входы — нет. Значит, КМОП-микросхема с напряжением питания +5 В может управлять ТТЛ-элементом. Логические уровни подходят друг другу, т. е. допустимые *H*-уровни КМОП-микросхемы попадают в область допустимых *H*-уровней ТТЛ-схемы. Также допустимые *L*-уровни КМОП-микросхемы попадают в область допустимых *L*-уровней ТТЛ-схемы (рис. 9.23). КМОП-выходы должны выдавать и принимать токи, номинальные для ТТЛ-схемы. При выполнении этих условий выходы КМОП-схемы совместимы с ТТЛ-схемой.

Если схемы различных семейств несовместимы или ограниченно совместимы, то они могут связываться друг с другом только через преобразователи уровня.

> Преобразователи уровня предназначены для преобразования уровней напряжения и тока схем одного семейства в уровни напряжения и тока схем другого семейства.

Преобразователи уровня по другому называются интерфейсными схемами (интерфейс, англ. — связь).

9.2.2. Структура преобразователей уровня

Преобразователи уровня могут быть построены на элементах различных семейств. Особенно подходят элементы НЕ и И-НЕ. Производители микросхем дают определенные рекомендации. Рассмотрим структуру преобразователя уровня TTL на CMOS.

На ТТЛ-стороне используется И-НЕ-элемент. Этот элемент должен управлять КМОП-элементом НЕ (рис. 9.24). У ТТЛ-схемы и КМОП-схемы следующие параметры:

ТТЛ-элемент		МОП-элемент	
$U_S = 5$ В $\pm$ 0,5 В	(Напряжение питания)	$U_S = 5$ В	(Напряжение питания)
$U_{QL\,max} = 0{,}4$ В	(максимальный выходной L-уровень)	$U_{IL\,max} = 1{,}5$ В	(Максимальный входной L-уровень)
$U_{QH\,min} = 2{,}4$ В	(Максимальный выходной H-уровень)	$U_{IH\,min} = 3{,}5$ В	(Максимальный входной H-уровень)
$I_{QL\,max} = 16$ мА	(Максимальный выходной ток L-уровня)	$I_{IL} = 10$ пА	(Входной ток L-уровня)
$I_{CEX\,max} = 100$ мА	(Минимальный выходной остаточный ток)	$I_{IH} = 10$ пА	(Входной ток H-уровня)

При L-состоянии выхода И-НЕ-элемента самое большое выходное напряжение 0,4 В. Это ниже максимального входного L-уровня КМОП-элемента 1,5 В. Преобразователь уровня в данном случае не требуется (рис. 9.25).

ТТЛ-элемент нуждается в L-входном токе. КМОП-элемент такой ток не выдает. Поэтому к источнику напряжения подключают резистор R_X (см. рис. 9.24).

Если И-НЕ-элемент имеет H-выходной уровень, то транзистор T_4 заперт (см. рис. 9.24). Через R_X на выход подается приблизительно +5 В.

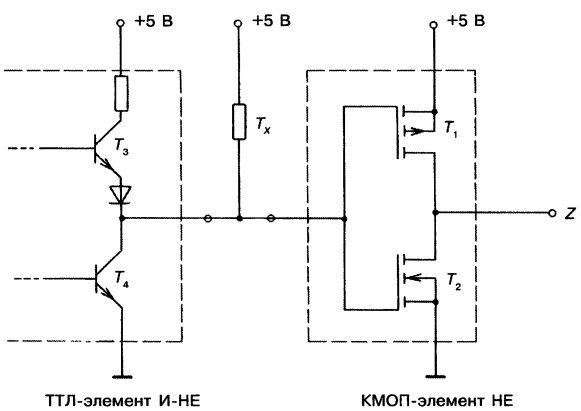


Рис. 9.24. Преобразователь уровня, построенный на ТТЛ-элементе И-НЕ и КМОП-элементе НЕ.

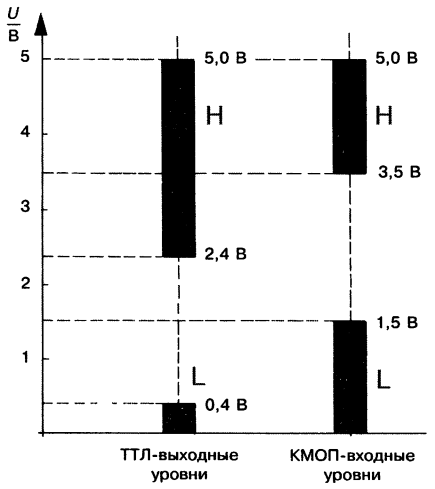


Рис. 9.25. Диаграмма уровней ТТЛ-КМОП.

Выходной уровень не может, как обычно допустимо в ТТЛ-схемах, упасть до 2,4 В. Уровень 2,4 В недопустим для *H*-входного уровня КМОП-схемы, так как ее минимальный *H*-входной уровень равен 3,5 В.

При вычислении номинала R_X нужно учитывать выполнение условий помехоустойчивости. R_X не должен быть очень большим или очень малым. Известный производитель КМОП-схем RCA приводит следующие уравнения:

$$R_{X\,\min} = \frac{U_{S\,\max} - U_{QL\,\max}}{I_{QL\,\max}};$$

$$R_{X\,\max} = \frac{U_S - U_{IH\,\min}}{I_{CEX\,\max}}.$$

Для рассмотренного примера преобразователя уровня получаются следующие значения:

$$R_{X\,\min} = \frac{5{,}5\ \text{В} - 0{,}4\ \text{В}}{16\ \text{мА}} = 319\ \text{Ом};$$

$$R_{X\,\max} = \frac{5\ \text{В} - 3{,}5\ \text{В}}{100\ \text{мкА}} = 15\ \text{кОм}.$$

Самое малое значение от R_X обеспечивает самую большую помехоустойчивость, но слишком сильно нагружает источник напряжения. Рационально выбрать значение в несколько кОм. Для рассмотренного преобразователя уровня выбрано:

$$R_X = 4{,}7\ \text{кОм}.$$

9.2.3. Интегральные преобразователи уровня

Часто применяемые преобразователи уровня выпускаются в виде микросхем. В цифровой технике управления в большом объеме применяется семейство МПЛ-медленной помехозащищенной логики. ТТЛ-элементы часто должны соединяться с МПЛ-элементами. Элементы МПЛ-семейства используют напряжение питания 12 В и 15 В с соответствующими уровнями *L* и *H*.

Преобразователь уровня FZH181 переводит ТТЛ-уровень на МПЛ-уровень. Принципиальная схема, таблица параметров и цоколевка микросхемы FZH181 представлена на рис. 9.26. Схема является в принципе И-НЕ-элементом с открытым коллектором. Рабочее сопротивление коллектора присоединяется к источнику напряжения питания МПЛ-элемента (12 В или 15 В). Выходной транзистор рассчитан на прикладываемую нагрузку.

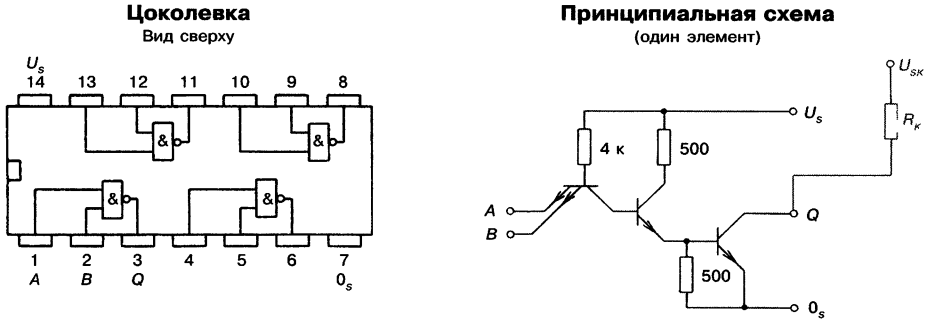
Преобразователь уровня ТТЛ-МПЛ

FZH 181

FZH 185

Микросхемы FZH 181 и FZH 185 содержат 4 ТТЛ-МПЛ преобразователя уровня, которые могут работать, как проводное И. Для вычисления общего рабочего коллекторного сопротивления применяются приведенные ниже формулы. Допустимое напряжение на выходе *Q* составляет максимально 18 В, ток — максимально 50 мА.

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Тестовая схема	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S			4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В	1	2,0			В
Входное напряжение *L*-уровня	U_{IL}	U_S = 4,75 В	8			0,8	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В, U_{IL} = 0,8 В, U_{QH} = 18 В	8			250	мкА
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В, U_{IH} = 2,0 В, I_{QL} = 16 мА	1			0,4	В
	U_{QL}	U_S = 4,75 В, U_{IH} = 2,0 В, I_{QL} = 50 мА	1			1,0	В
Статическая помехоустойчивость	U_{SS}			0,4	1,0		В
Входной ток на канал	I_I	U_S = 5,25 В, U_I = 5,5 В	3			1,0	мА
H-входной ток на канал	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В	3			80	мкА
L-входной ток на канал	$-I_{IL}$	U_S = 5,25 В, U_{IL} = 0,4 В	4			1,6	мА
Потребляемый ток уровня *H* на канал	I_{SH}	U_{SH} = 5 В, U_I = 0 В	6		1,0	2,0	мА
Потребляемый ток уровня *L* на канал	I_{SL}	U_S = 5 В, U_I = 5 В	7			12	мА
Потребляемая мощность на канал	*P*	$U_S = U_{SA}$ Скважность 1 : 1			24	37	мВт
Время переключения, при U_S = 5 В, T_U = 25 °C							
Время прохождения сигнала (быстродействие)	t_{PLH}	R_K = 760 Ом, C_L = 15 пФ, U_{SK} = 12 В	29		130	300	нс
	t_{PHL}		29		20	60	нс
Логические параметры на канал							
Коэффициент разветвления по выходу уровня	F_{QL}					10	
Входной нагрузочный коэффициент на канал	F_I					1	
Логическая функция	$Q = \overline{A \wedge B}$						



Расчет рабочего коллекторного сопротивления R_K

Сопротивление R_K рассчитывается по необходимому смещению напряжения и по входным и выходным токам следующим образом:

$$R_{KA} = \frac{U_{SK} - U_{QH}\ [\text{В}]}{nI_{QH} + NI_{IH}\ [\text{мкА}]} \qquad R_{KB} = \frac{U_{SK} - U_{QL}\ [\text{В}]}{I_{QH} + NI_{IL}\ [\text{мкА}]},$$

где U_{SK} = напряжение питания рабочего сопротивления; n = число соединений И; N = число подключенных входов.

Применяемое в схеме сопротивление должно быть в диапазоне между верхним A и нижним B граничными значениями.

В случае применения в качестве преобразователя уровня для схем FZX 181/185 ТТЛ-МПЛ 12 В:

$$R_{KA} = \frac{12 - 10\ [\text{В}]}{n250 + N1\ [\text{мкА}]} \qquad R_{KB} = \frac{12 - 1{,}0\ [\text{В}]}{50 + N1{,}5\ [\text{мкА}]};$$

FZX 181/185 ТТЛ-МПЛ 15 В:

$$R_{KA} = \frac{15 - 12\ [\text{В}]}{n250 + N1\ [\text{мкА}]} \qquad R_{KB} = \frac{15 - 1{,}0\ [\text{В}]}{50 + N1{,}8\ [\text{мкА}]},$$

где $n_A = 4$ для $N_A = 25$.

Рис. 9.26. Принципиальная схема, таблица параметров и цоколевка микросхемы FZH 181 (Siemens).

Контрольный тест

1. Нарисуйте схему преобразователя десятичного кода в *BCD*-код (двоично-десятичный).
2. Как рассчитать преобразователь кода для любых кодов?
3. Нарисуйте схему преобразователя десятичного кода в код Айкена.
4. Рассчитайте преобразователь кода Грея (рис. 8.13) в *BCD*-код (двоично-десятичный).
5. Для чего нужен преобразователь уровней?
6. КМОП-схема питается от источника напряжения 3 В. Для вывода данных ее нужно заменить на микросхему стандартной ТТЛ, чтобы управлять 7-сегментым индикатором со световыми сегментами. Какие проблемы возникают при преобразовании?

ГЛАВА 10

СЧЕТЧИКИ И ДЕЛИТЕЛИ ЧАСТОТЫ

10.1. Счет и разновидности счетчиков

Счетом — или точнее сказать **счетом на увеличение** — является непрерывное прибавление 1 к предыдущему числу. К начальному значению, которое часто равно нулю, снова и снова прибавляется 1 до завершения процесса счета.

Счетом на уменьшение является непрерывное вычитание 1. Счет на уменьшение начинается с начального значения и продолжается до конечного. Это конечное значение может быть нулем, но не обязательно.

Счет давно известен в десятичной системе счисления и является привычной операцией. Но и любая другая система счисления также пригодна для счета. Например, считать можно в двоичных числах или в шестнадцатеричной системе счисления. Также можно считать во всевозможных кодах.

Для любых задач, связанных со счетом, могут быть синтезированы электронные счетчики. Особенное значение имеют бинарные счетчики, также называемые **двоичными счетчиками**.

> Двоичные счетчики обрабатывают только сигналы 0 и 1.

Почти все применяемые в настоящее время электронные счетчики являются двоичными. Счетчики, работающие с тремя, четырьмя или большим количеством различных входных сигналов, не имеют практического значения. Поэтому двоичный счетчик можно называть просто «счетчик».

Счетчики могут работать с различными кодами или системами счисления. Системы счисления рассматриваются как особенный код. В зависимости от поставленной задачи нужен либо суммирующий счетчик, либо вычитающий, либо реверсивный.

> Счетчики различаются по обрабатываемому коду и по направлению счета.

Счетчики строятся на бинарных элементах. Стандартными элементами являются бистабильные ячейки, так называемые триггеры. Триггеры переключаются в определенные моменты времени. Если все триггеры переключаются одновременно, то это синхронный режим работы. В асинхронном режиме триггеры переключаются в разные моменты времени. Счетчики, которые работают в синхронном режиме, называются **синхронными счетчиками**. Счетчики, которые работают в асинхронном режиме, называются **асинхронными счетчиками**.

> Счетчики разделяются на синхронные и асинхронные.

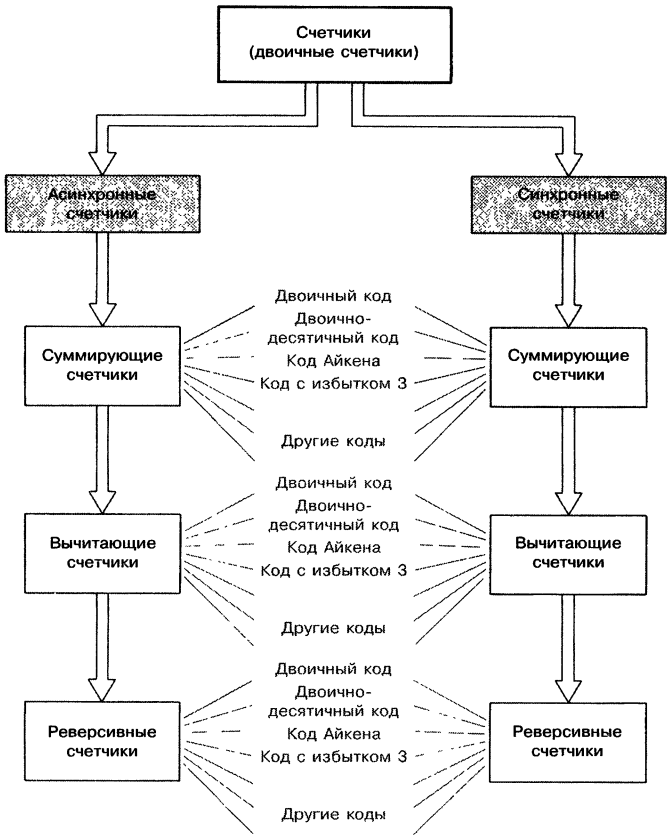


Рис. 10.1. Обзор видов счетчиков.

Существует множество возможных видов счетчиков. На рис. 10.1 приведен обзор разновидностей счетчиков.

10.2. Асинхронные счетчики

В асинхронных счетчиках триггеры переключаются общим тактовым сигналом не одновременно.

10.2.1. Асинхронные двоичные счетчики

Асинхронные двоичные счетчики работают в двоичной системе счисления. Они могут строиться на различных разновидностях триггеров. Самая простая структура получается на *T*-триггерах. *JK*- и *RS*-триггеры могут соединяться таким образом, что они работают как *T*-триггеры (рис. 10.2).

10.2.1.1. Двоичные суммирующие счетчики

На схеме (рис. 10.3) показан двоичный суммирующий счетчик, построенный на трех *T*-триггерах. Такой счетчик называется 3-битовым, или 3-раз-

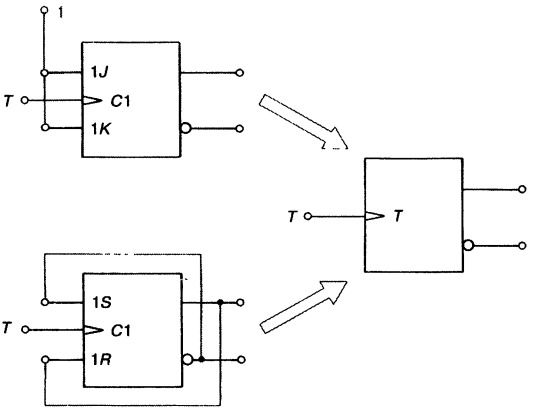


Рис. 10.2. *JK*-триггер и *RS*-триггер, включенные как *T*-триггер.

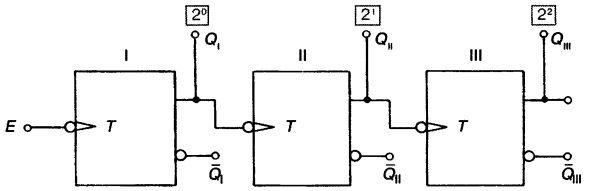


Рис. 10.3. Двоичный суммирующий счетчик.

рядным двоичным суммирующим счетчиком. Каждый триггер имеет объем памяти в один бит и отвечает за один бинарный разряд. Получаемый на выходе результат — двоичное число — имеет столько разрядов, сколько триггеров имеется в счетчике.

T-триггеры схемы на рис. 10.3 переключаются при переходе сигнала с 1 на 0, то есть обратным фронтом сигнала. Переходные процессы и временные диаграммы представлены на рис. 10.4.

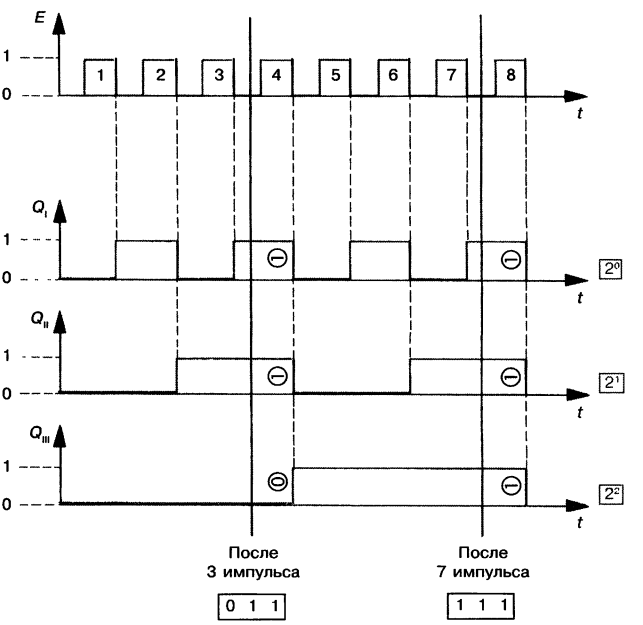


Рис. 10.4. Временные диаграммы двоичного счетчика на рис. 10.3.

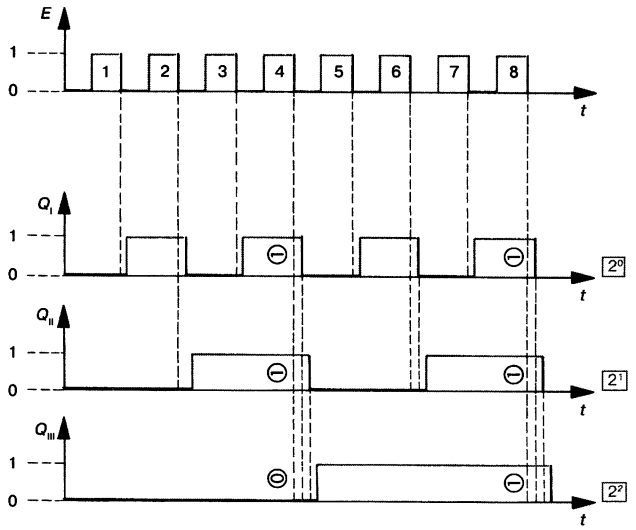


Рис. 10.5. Временные диаграммы с учетом времени прохождения сигнала (период входного сигнала 200 нс, время прохождения сигнала 30 нс).

Временные диаграммы идеализированы. Если сигнал от Q_I переходит с 1 на 0, то пройдет некоторое время, пока сигнал на Q_{II} перейдет с 0 на 1. Это время называется **временем прохождения сигнала**, или **быстродействием**. Оно составляет для триггеров ТТЛ-семейства от 30 до 50 нс. Для низкочастотных входных сигналов временем распространения сигнала можно пренебречь. Если период входного сигнала на рис. 10.4 составляет 0,2 мкс = 200 нс (5 МГЦ), то время прохождения сигнала следует учитывать. Для Q_I, Q_{II} и Q_{III} получаются сдвинутые относительно друг друга импульсы согласно рис. 10.5. Смещение импульсов является недостатком асинхронного метода. Оно приводит к уменьшению максимально возможной частоты работы.

Если суммирующий счетчик построен на *T*-триггерах, переключаемых прямым фронтом импульса, то для переключения следующего триггера применяется инверсный выход $\bar{Q}$ (рис. 10.6). 4-битный двоичный счетчик может считать до 15.

> Двоичные суммирующие счетчики считают от нуля до максимального значения, затем переключаются обратно на нуль и начинают счет сначала.

JK-MASTER-SLAVE-триггер используется как универсальный триггер. Микросхемы из нескольких таких триггеров стоят недорого. Схема FLJ131-7476 (см. рис. 7.80) содержит два *JK*-MASTER-SLAVE-триггера. На двух таких микросхемах можно построить 4-битный двоичный суммирующий счетчик. Как выглядит схема этого счетчика?

JK-триггеры должны работать как *T*-триггеры. На все *J*- и *K*-входы нужно подать состояние 1, т. е. соединить с напряжением питания. Как переключаются триггеры? С прямым или обратным фронтом? Открытая стрелка в условном обозначении указывает переключение ведущего триггера. Он переключается с передним фронтом. Ведомый триггер переключается с обратным фронтом. Передача сигнала в следующий триггер происходит с обратным фронтом — как в триггере с кружком отрицания перед стрелкой в условном обозначении. Поэтому *Q*-выход триггера нужно соединять с *C*-входом следующего триггера. Схема представлена на рис. 10.7.

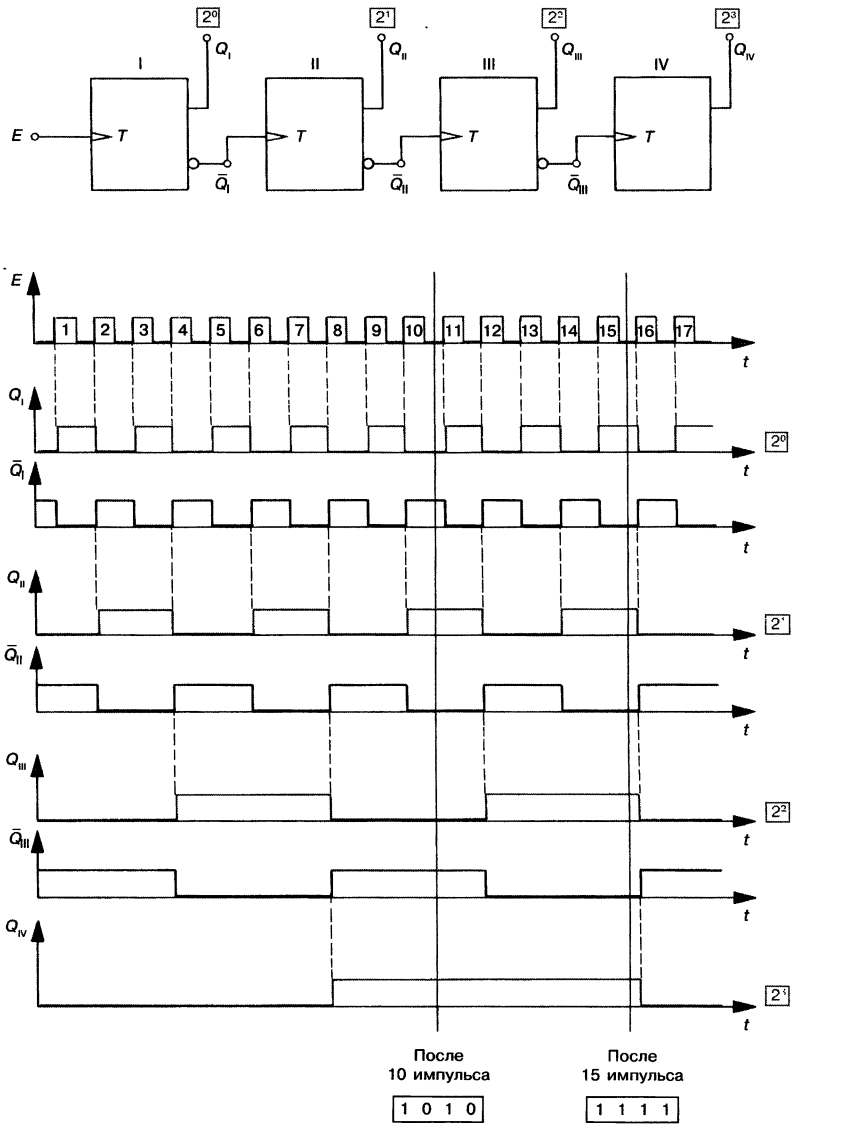


Рис. 10.6. Временные диаграммы 4-битового двоичного суммирующего счетчика.

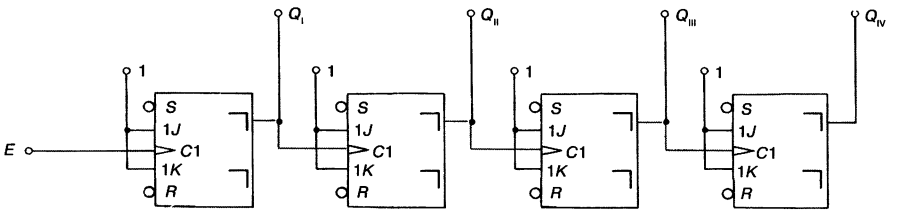


Рис. 10.7. Схема 4-битового двоичного суммирующего счетчика.

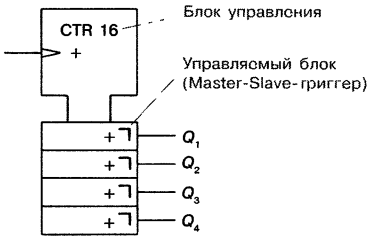


Рис. 10.8. Сборная схема 4-битового двоичного суммирующего счетчика на Master-Slave-триггерах, согласно DIN 40900, часть 12, CTR 16 означает 16 счетных разрядов (CTR — Counter, англ. счетчик).

JK-триггеры имеют нетактируемые входы сброса и установки. Входы установки обозначены символом *S*, входы сброса — *R*. Эти входы не используются.

Счетчики этого типа применяются очень часто. Поэтому они выпускаются в виде микросхем, их соответствующее условное обозначение показано на рис. 10.8. К одному управляющему блоку подключено четыре *JK*-MASTER-SLAVE-триггера. Они соединены так, чтобы получился суммирующий счетчик. Знак плюс обозначает суммирующий счетчик. Обозначение «CTR» является сокращением от Counter, от английского слова счетчик. Число указывает количество возможных счетных шагов.

4-битные двоичные суммирующие счетчики можно усовершенствовать. Возможно устанавливать счетчик на любое начальное значение. Установка происходит через нетактируемые входы *S*. Круг отрицания перед входами сигнала установки в единицу указывает, что для установки требуется 0-сигнал. Чтобы устанавливать 1-сигналами, надо перед входами установки включить элементы НЕ.

Входы сброса *R* могут использоваться для общего сброса счетчика. *R*-входы связываются друг с другом. 0-сигнал E_R независимо от такта возвращает счетчик на нуль. Схема 4-битового двоичного суммирующего счетчика с возможностью установки и сброса показана на рис. 10.9. Такой счетчик называется двоичным суммирующим счетчиком с возможностью предварительной установки.

Разрядность двоичного суммирующего счетчика говорит о возможности счета. 4-битные двоичные суммирующие счетчики могут считать до 15, 5-битные двоичные суммирующие счетчики — до 31, 6-битные двоичные суммирующие счетчики — до 63 и т. д.

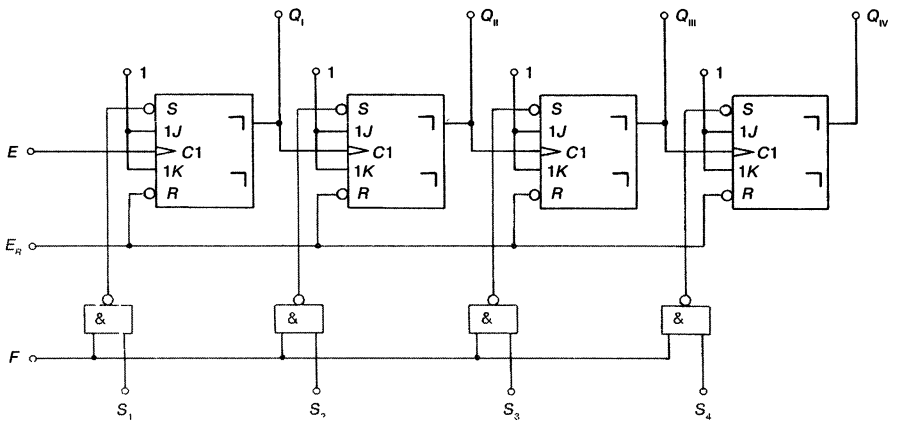


Рис. 10.9. 4-битовый двоичный суммирующий счетчик с нетактируемыми входами сброса и установки.

Количество триггеров (n)	Максимальный счет до (K)	
2	3	
3	7	
4	15	
5	31	
6	63	$K = 2^n - 1$
7	127	
8	255	
9	511	
10	1023	

Асинхронные счетчики выпускаются в виде интегральных микросхем. Микросхема FLJ 181-7493 A устроена просто и ясно. Схема и цоколевка представлены на рис. 10.10. Чтобы сделать микросхему универсальной для различных приложений, выход Q_A внутренне не соединен с входом B. Схема может быть применена, например в качестве 3-битового двоичного суммирующего счетчика, если входной сигнал подается на вход B. Если соединить выход Q_A с клеммой B, то получается 4-битовый двоичный суммирующий счетчик (выделено серым на рис. 10.11).

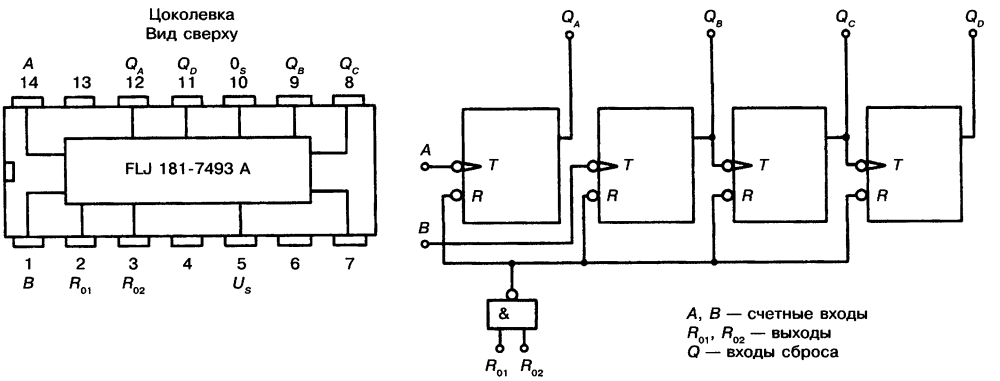


Рис. 10.10. Схема и цоколевка 4-битового двоичного суммирующего счетчика.

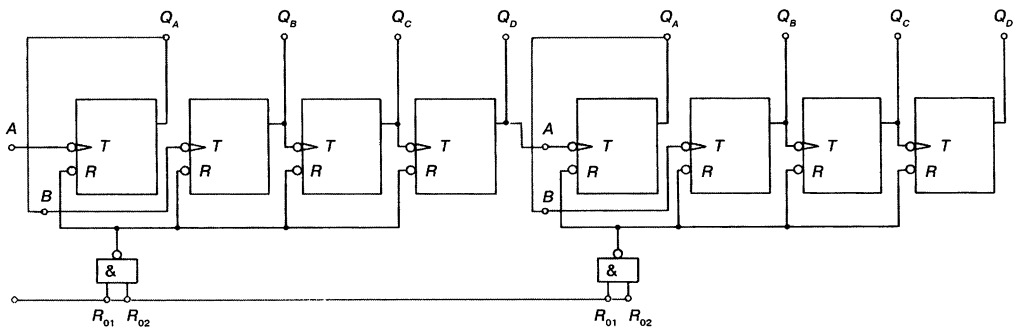


Рис. 10.11. 8-битовый двоичный суммирующий счетчик, собранный на двух микросхемах FLJ 181-7493 A.

Микросхема FLJ 181-7493 A содержит нетактируемые входы сброса, которые управляются через элементы И-НЕ. Счетчик сбрасывается на нуль, если на обоих входах R_{01} и R_{02} действует 1-сигнал.

Задание

На двух микросхемах FLJ 181-7493 A построить 8-битовый двоичный суммирующий счетчик. Счетчик должен сбрасываться в нулевое значение при подаче 1-сигнала на общий вход сброса *R*. Изобразите схему счетчика и соединения между интегральными микросхемами.

Микросхемы следует соединить таким образом, чтобы выход Q_D первой микросхемы управлял входом *A* второй. В обеих микросхемах следует объединить входы Q_A и *B*.

Входы сброса R_{01} и R_{02} обеих схем подключаются к общему входу сброса *R* (рис. 10.11). Соединение схем показано на рис. 10.12.

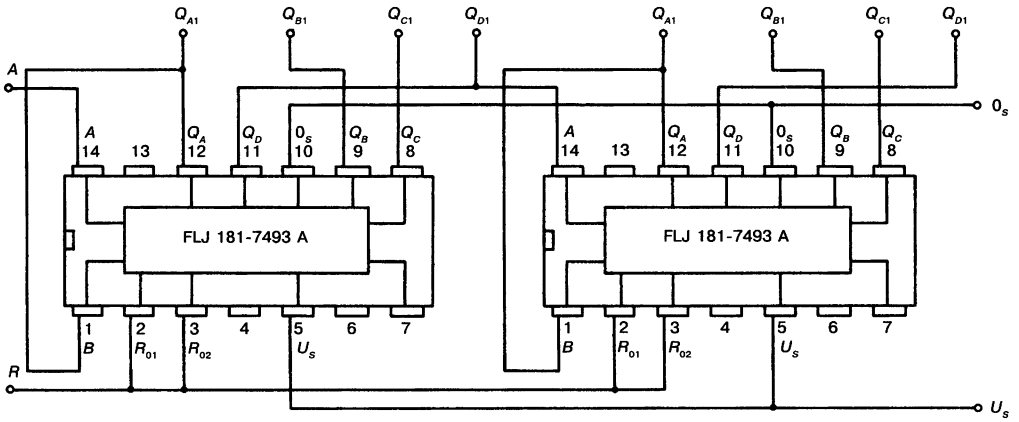


Рис. 10.12. План схемы 8-битового двоичного суммирующего счетчика на рис. 10.11.

10.2.1.2. Двоичные вычитающие счетчики

Двоичные вычитающие счетчики считают от максимального значения до нуля, затем переключаются обратно на максимальное значение и начинают обратный счет сначала.

Представленный на рис. 10.3 3-битовый двоичный суммирующий счетчик может быть легко преобразован в 3-битовый двоичный вычитающий счетчик. Для этого следует отключить от тактовых *T*-входов триггеров выходы *Q* и подключить *T*-входы к выходам $\bar{Q}$ (рис. 10.13). Временные диаграммы для 3-битового вычитающего счетчика приведены на рис. 10.14.

Если счетчик строится на триггерах, переключающихся с передним фронтом (переход с 0 на 1), то для управления следующим триггером следует использовать *Q*-выходы. На рис. 10.15 показана схема 4-битового двоичного вычитающего счетчика, который построен на таких триггерах, и соответствующие временные диаграммы.

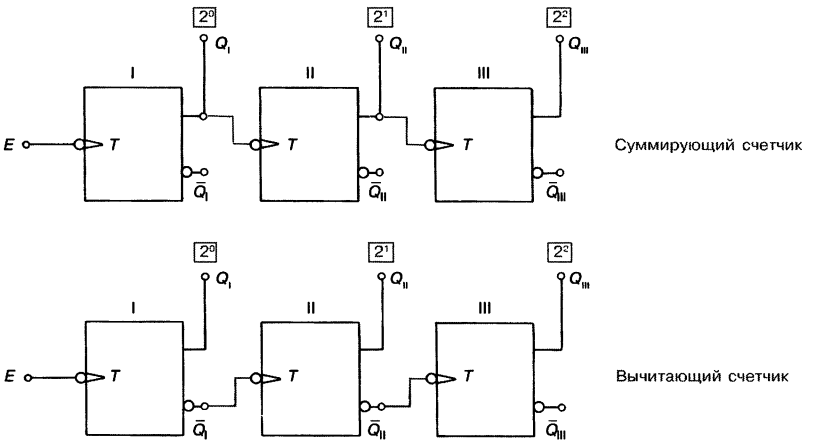


Рис. 10.13. Преобразование 3-битового двоичного суммирующего счетчика в 3-битовый двоичный вычитающий счетчик.

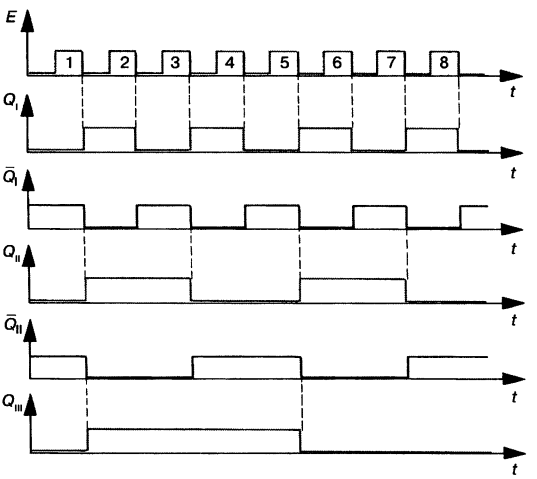


Рис. 10.14. Временные диаграммы 3-битового двоичного вычитающего счетчика на рис. 10.13.

Задание

Построить на *JK*-MASTER-SLAVE-триггерах микросхемы FLJ 131-7476 6-битовый двоичный вычитающий счетчик. Нарисовать схему этого счетчика и его условное обозначение согласно DIN 40700, ч. 14.

Каждый *JK*-MASTER-SLAVE-триггер схемы FLJ 131-7474 соединен по схеме *T*-триггера, то есть переключается (меняет состояние выходов на инверсные) на каждом такте. На входах *J* и *K* действует единичное состояние. Передача сигнала на выход происходит с обратным фронтом сигнала, как в *T*-триггере, *T*-вход которого обозначен кружком инверсии (рис. 10.16).

Как следует соединять триггеры друг с другом? Какой выходной сигнал должен использоваться для управления следующим триггером? Чтобы получить двоичный суммирующий счетчик, нужно было бы соединить *Q*-выходной сигнал с тактовым входом следующего триггера. Так как требуется получить вычитающий счетчик, для управления следует использовать инверсные $\bar{Q}$-выходные сигналы. Схема представлена на рис. 10.17.

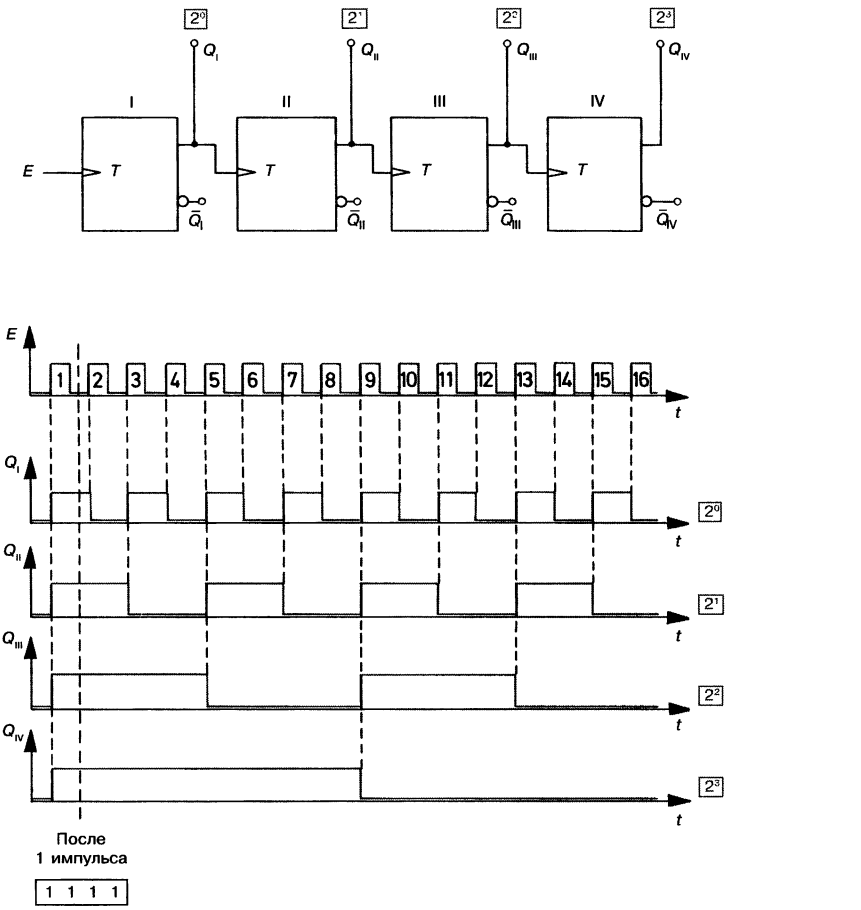


Рис. 10.15. 4-битовый двоичный вычитающий счетчик и его временная диаграмма.

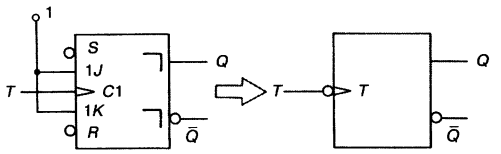


Рис. 10.16. *JK*-Master-Slave-триггер, который меняет свое состояние на инверсное на каждом такте как *T*-триггер.

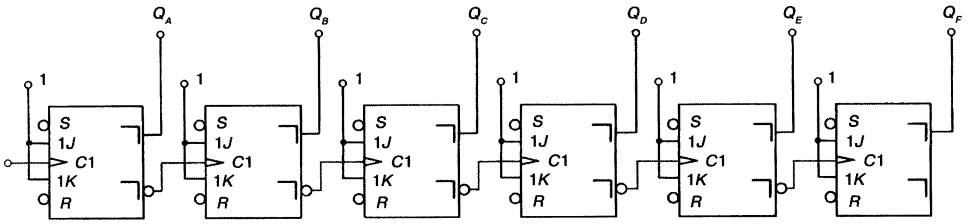


Рис. 10.17. 6-битовый двоичный вычитающий счетчик.

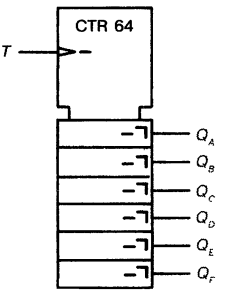


Рис. 10.18. Условное обозначение 6-битового двоичного вычитающего счетчика на рис. 10.17, 64 счетных разряда.

Условное обозначение 6-битового двоичного вычитающего счетчика согласно DIN 40700, часть 14 показано на рис. 10.18. К управляющему блоку подключены 6 управляемых функциональных блоков.

Двоичные вычитающие счетчики практически не производятся в виде микросхем. Этому есть объяснение. Дело в том, что вычитающий счетчик легко можно собрать из суммирующего. Можно инвертировать сигналы всех выходов или использовать инверсные выходы триггеров. В интегральных микросхемах инверсные выходы часто не выведены. Однако инвертирование прямых выходов не представляет никаких трудностей.

10.2.1.3. Реверсивные двоичные счетчики

Направление счета двоичного счетчика зависит от сигналов, управляющих триггерами. Можно управлять прямыми выходами Q или инверсными $\bar{Q}$. Направление счета зависит и от типа используемых триггеров, в частности от того, переключаются эти триггеры с прямым или обратным фронтом управляющего сигнала.

Если счетчик построен на T-триггерах, срабатывающих по обратному фронту, то управление Q-сигналами образует суммирующий счетчик (см. подразд. 10.2.1.1). Управление $\bar{Q}$-сигналами образует вычитающий счетчик (см. подразд. 10.2.1.2).

Зная это, легко построить реверсивный счетчик.

> В двоичном счетчике направление счета меняется переключением сигналов управления Q и $\bar{Q}$.

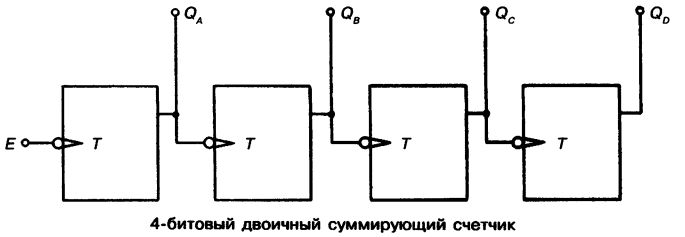
На рис. 10.19 показана структура 4-битового реверсивного двоичного счетчика.

Переключение происходит при помощи ключа. Переключение ключом имеет значительные недостатки. На практике ключи заменяются логическими элементами. На рис. 10.20 изображена схема 4-битового реверсивного двоичного счетчика. Если на управляющий вход U подан 1-сигнал, то счетчик работает как суммирующий, если 0-сигнал, то как вычитающий.

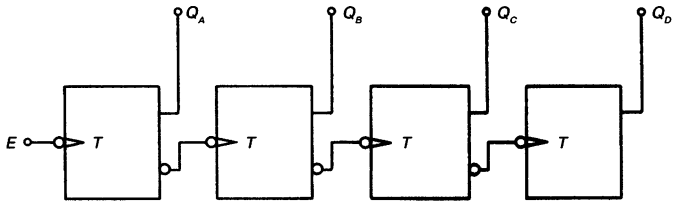
Двоичные реверсивные счетчики производятся в виде интегральных микросхем. ТТЛ-серия 4-битовых реверсивных двоичных счетчиков называется FLJ211-74191. Схемы этой серии имеют предварительную установку начального значения, т. е. начальное значение устанавливается посредством четырех информационных входов.

Двоичные реверсивные счетчики представляются согласно DIN 40700, часть 14 условным обозначением, изображенным на рис. 10.21.

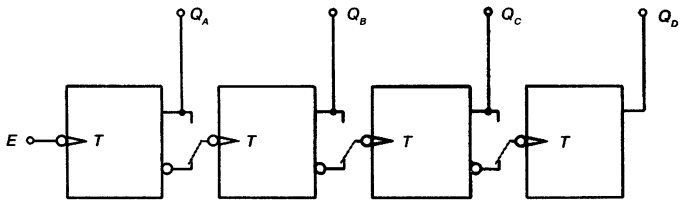
На рисунке изображен 8-битовый реверсивный двоичный счетчик. Если на управляющий вход U подан 0-сигнал, то счетчик работает как суммирующий, если 1-сигнал, то как вычитающий.



4-битовый двоичный суммирующий счетчик

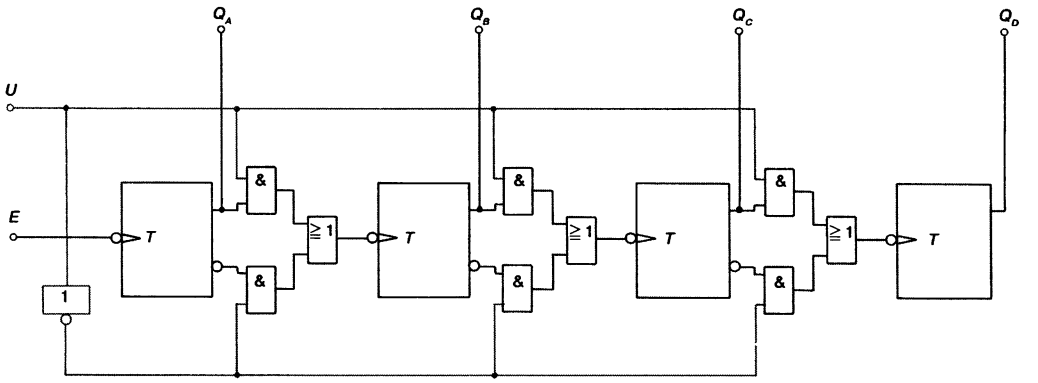


4-битовый двоичный вычитающий счетчик

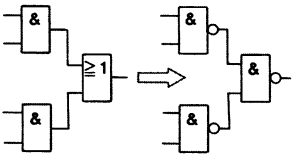


4-битовый двоичный реверсивный счетчик (принципиальная схема)

Рис. 10.19. Организация 4-битового двоичного реверсивного счетчика.



$U = 1$ суммирующий счетчик
$U = 0$ вычитающий счетчик



Равнозначная схема

Рис. 10.20. 4-битовый двоичный реверсивный счетчик.

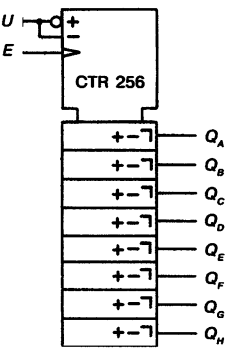


Рис. 10.21. Условное обозначение 8-битового двоичного вычитающего счетчика, 256 счетных разрядов.

10.2.2. Асинхронные *BCD*-счетчики (в двоично-десятичном коде)

BCD-счетчики в основном являются 4-битовыми. На их выходах должны быть сигналы двоично-десятичного кода.

Двоично-десятичный код выражает десятичные цифры от 0 до 9 в виде двоичных чисел, таким образом, он является десятичным кодом. Так как *BCD*-счетчики считают закодированные десятичные цифры, их часто называют десятичными счетчиками. Это обозначение приводит к недоразумениям. Возможна путаница со счетчиками, которые работают с кодом 1-из-10.

BCD-счетчики бывают суммирующими, вычитающими и реверсивными.

10.2.2.1. BCD-суммирующие счетчики

BCD-суммирующие счетчики можно построить на 4-битовых двоичных суммирующих счетчиках. Применяемые триггеры должны иметь нетактируемые входы сброса. Счетчик должен считать только до двоичного числа 1001, то есть до десятичной цифры 9, а затем сбрасываться в нуль. Сброс должен происходить в момент переключения счетчика с 1001 на 1010.

Схема 4-битового двоичного суммирующего счетчика представлена на рис. 10.22. *T*-Триггеры имеют нетактируемые входы сброса *R*. Какие дополнительные подключения требуются, чтобы сделать из 4-битового двоичного суммирующего счетчика *BCD*-суммирующий счетчик?

Для сброса счетчика в нуль требуется подать 1-сигналы на *R*-входы триггеров. Так как все триггеры должны сбрасываться одновременно, целесообразно соединить все *R*-входы друг с другом.

Сброс должен происходить при переходе счетчика с десятичной цифры 9 на десятичную цифру 10. То есть на выходах Q_B и Q_D должен быть 1-сигнал. Выходы Q_B и Q_D связываются через И-элемент. Выход И-элемента соединен с входом сброса *R*. Это дополнительное соединение выделено серым на рис. 10.22.

Такой счетчик безупречно работает при невысокой частоте синхроимпульсов. Произвольный сброс в начальное положение не может происхо-

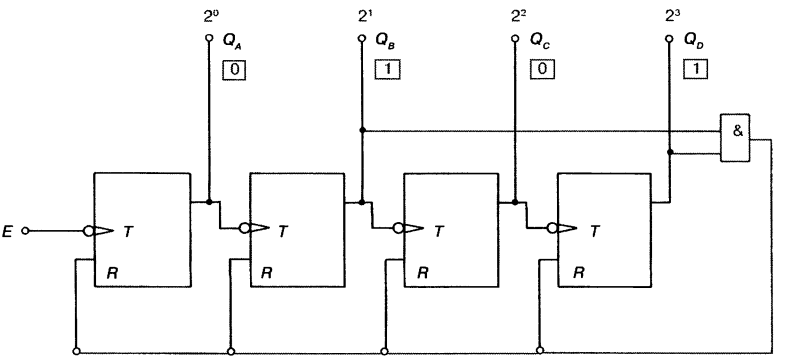


Рис. 10.22. Преобразование 4-битового двоичного суммирующего счетчика в *BCD*-суммирующий счетчик.

Десятичная цифра	BCD-код 2^3 Q_D	2^2 Q_C	2^1 Q_B	2^0 Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Рис. 10.23. *BCD*-код.

дить до достижения 10, так как только тогда на выходах Q_B и Q_D появляется 1-сигнал (рис. 10.23). При всех других комбинациях выходного сигнала — для чисел от 0 до 9 — никогда не получается одновременно $Q_B = 1$ и $Q_D = 1$.

Недостатком такого счетчика является кратковременное присутствие сигнала десятичной цифры 10. То есть счетчик считает до 10 включительно и затем удаляет состояние 10. Для ТТЛ-схем состояние 10 держится на выходах примерно 50 нс. Это очень короткое время. Однако для ряда задач этот сигнал может послужить причиной помех.

Недостаток кратковременного присутствия десятичного числа 10 устранен в схеме на рис. 10.24. Этот счетчик построен на управляемых по одному фронту *JK*-триггерах. Соединение производится последовательно с помощью временной диаграммы.

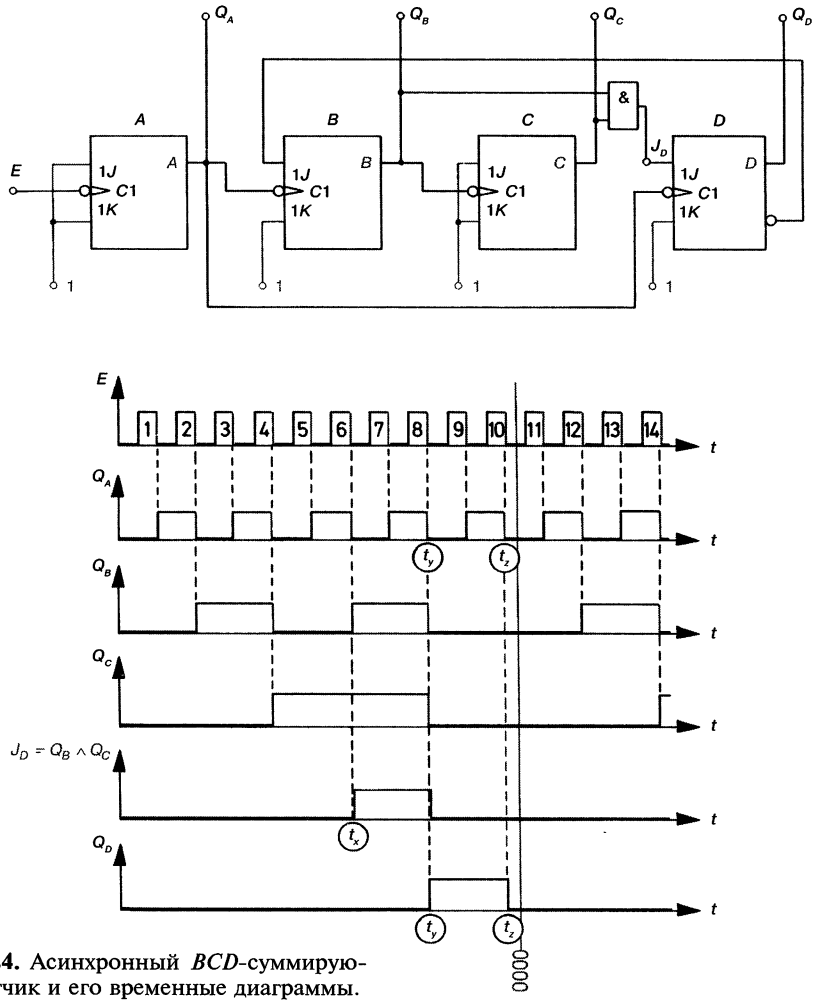


Рис. 10.24. Асинхронный *BCD*-суммирующий счетчик и его временные диаграммы.

Триггер *A* работает как *T*-триггер. Триггер *B* работает так же, как *T*-триггер, но только до тех пор, пока триггер *D* находится в состоянии покоя (чтобы *JK*-триггер работал как *T*-триггер, на его входах *J* и *K* должны быть 1-сигналы). Импульсы Q_A следуют как обычно. Порядок следования импульсов Q_B нарушается, как только триггер *D* переключается в рабочее состояние. Если 1-сигнал на J_B пропадает, то триггер B должен остаться в состоянии покоя.

Триггер *C* работает снова как *T*-триггер и управляется импульсами Q_B. Рассмотрим сигнал J_D:

$$J_D = Q_B \wedge Q_C.$$

Сигнал J_D выделен на временной диаграмме (рис. 10.24). В момент времени $t_X J_D = 1$. Теперь триггер *D* может переключиться в рабочее состояние, но только тогда, когда сигнал синхронизации Q_A переходит с 1 на 0. Это происходит в точке t_Y. $Q_D = 1$. Вскоре после этого J_D снова равен 0, и триггер *D* должен переключиться с обратным фронтом своего сигнала синхронизации Q_A в состояние покоя, а именно в момент времени t_Z. После 10-го обратного фронта импульса входного сигнала *E* на выходах счетчика будет присутствовать комбинация 0000. *BCD*-счетчик выставлен на нуль и начинает счет снова. *BCD*-суммирующий счетчик (рис. 10.24) с незначительными дополнениями выпускается как микросхема HJ 161-7490 A.

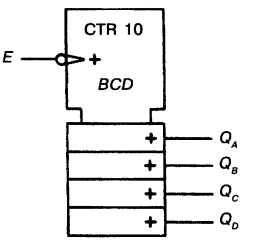


Рис. 10.25. Условное обозначение *BCD*-суммирующего счетчика, срабатывающего с обратным фронтом. CTR 10 означает 10 счетных разрядов.

Весь *BCD*-счетчик можно изобразить условным обозначением. Оно состоит из одного управляющего блока и 4 управляемых функциональных блоков, соответствующих четырем триггерам. Плюс означает суммирование. Код, с которым работает счетчик, нужно указывать в управляющем блоке (рис. 10.25).

10.2.2.2. BCD-вычитающий счетчик

BCD-вычитающие счетчики также можно построить на базе двоичного вычитающего счетчика (рис. 10.26). Вычитающие двоичные счетчики с помощью дополнительных соединений можно превратить в счетчик с обратным счетом, начиная с десятичного числа 9, то есть с двоичного числа 1001. Без дополнительных соединений он будет начинать счет с десятичного числа 15 или двоичного числа 1111.

Кратковременно (около 50 нс) на выходах находится двоичное число 1111. С этим сигналом триггеры *B* и *C* (рис. 10.26) должны быть сброшены в состояние покоя ($Q_B = 0$, $Q_C = 0$). Этот сброс производят через нетактируемые входы сброса. Все четыре выходных сигнала можно было бы подать на И-элемент. Но в этом нет необходимости. Достаточно подать сигналы

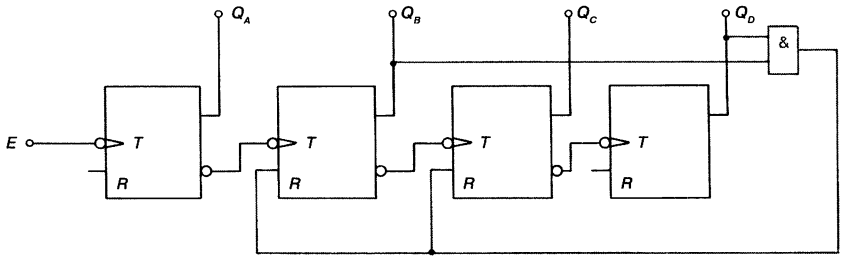


Рис. 10.26. Преобразование 4-битового двоичного вычитающего счетчика в *BCD*-вычитающий счетчик.

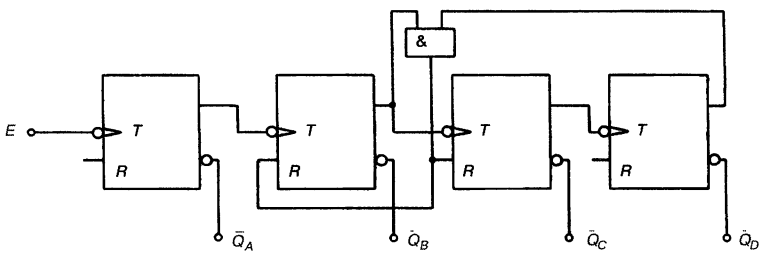


Рис. 10.27. Двоичный вычитающий счетчик.

Десятичная цифра	Q_D	Q_C	Q_B	Q_A	$\bar{Q}_D$	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	Десятичное число
0	0	0	0	0	1	1	1	1	15
1	0	0	0	1	1	1	1	0	14
2	0	0	1	0	1	1	0	1	13
3	0	0	1	1	1	1	0	0	12
4	0	1	0	0	1	0	1	1	11
5	0	1	0	1	1	0	1	0	10
6	0	1	1	0	1	0	0	1	9
7	0	1	1	1	1	0	0	0	8
8	1	0	0	0	0	1	1	1	7
9	1	0	0	1	0	1	1	0	6

Рис. 10.28. Кодовая таблица.

Q_B и Q_D на И-элемент, так как в диапазоне десятичных чисел от 9 до 0 никогда одновременно не появляются $Q_B = 1$ и $Q_D = 1$. (Возможно также использовать сигналы Q_C и Q_D.) Необходимые дополнительные соединения показаны серым на рис. 10.26.

Из *BCD*-суммирующего счетчика можно сделать достаточно простой двоичный вычитающий счетчик. Нужно только инвертировать выходные сигналы *BCD*-суммирующего счетчика или использовать $\bar{Q}$-выходы триггеров как выходы счетчика (см. рис. 10.27). Справедливость сказанного подтверждает кодовая таблица на рис. 10.28.

10.2.2.3. BCD-реверсивный счетчик

BCD-реверсивный счетчик можно построить на базе 4-битового реверсивного двоичного счетчика (см. рис. 10.20). Особенное внимание нужно обратить на включение входов сброса.

Счетчик на рис. 10.29 построен на *T*-триггерах, переключающихся с обратным фронтом. Входы сброса этих триггеров управляются 0-сигналами.

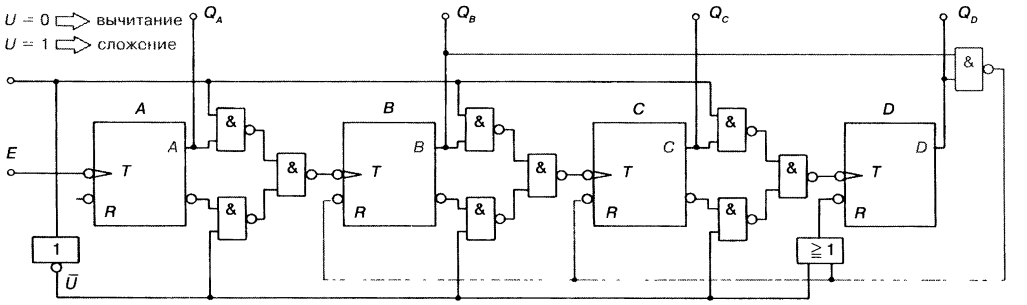


Рис. 10.29. *BCD*-реверсивный счетчик.

В режиме суммирующего счетчика на входе U должен действовать сигнал 1. При сигнале 0 на U счетчик работает в режиме вычитания.

BCD-вычитающий счетчик должен перед началом счета на вычитание быть установлен на десятичное число 9 или на двоичное число 1001. Триггеры B и C должны быть установлены на 0. При $Q_B = 1$ и $Q_D = 1$ к выходу И-НЕ-элемента приложен 0. С этим сигналом сбрасываются триггеры B и C. Триггер D не будет сброшен, так как на его входе сброса действует 1 через элемент ИЛИ.

Если счетчик работает в режиме сложения, то он должен сбрасываться на нуль, если $Q_B = 1$ и $Q_D = 1$. На выходе И-НЕ-элемента появляется 0-сигнал. С этим сигналом сбрасываются триггеры B, C и D. Триггер D теперь также сбрасывается, так как на нижней линии управления действует 0-сигнал. Сброс триггера A не является необходимым, так как он и без того стоит на нуле ($Q_A = 0$).

Задание

Построить и нарисовать схему *BCD*-счетчика (рис. 10.29) на базе микросхем FLJ 131-7476 и FLH101-7400.

Микросхема FLJ131-7476 содержит два *JK*-MASTER-SLAVE-триггера. Следовательно, необходимы две микросхемы. Микросхема FLH101-7400 (см. рис. 6.65) содержит четыре И-НЕ-элемента. Для связи между двумя

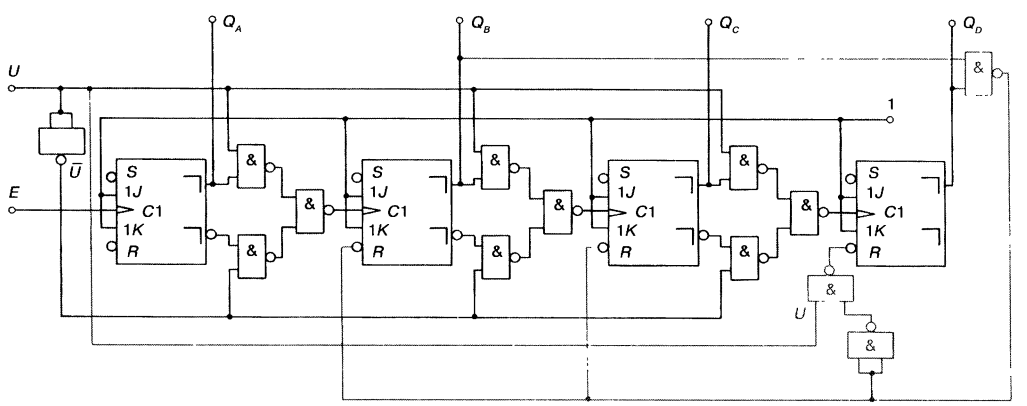


Рис. 10.30. *BCD*-реверсивный счетчик (2 × FLJ131-7476, 3 × FLH101-7400).

триггерами используются три И-НЕ-элемента, для трех триггеров нужны 9 таких элементов. Еще один И-НЕ-элемент требуется для получения сигнала сброса. Для замены ИЛИ-элемента требуются три И-НЕ-элемента. Всего 13 И-НЕ-элементов. То есть необходимы четыре микросхемы FLH101-7400 с 16 И-НЕ-элементами. Если сэкономить один И-НЕ-элемент, то необходимы только 3 микросхемы FLH101-7400.

На рис. 10.30 показана схема. Один И-НЕ-элемент можно сэкономить, если вместо сигнала $\bar{U}$ для управления сбросом триггера D использовать сигнал U.

10.2.3. Асинхронный декадный счетчик

10.2.3.1. BCD-декадный счетчик

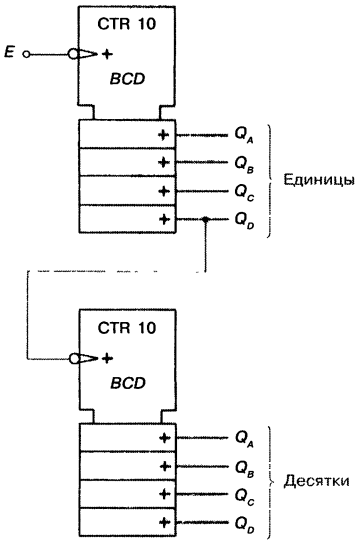


Рис. 10.31. *BCD*-декадный счетчик для двух декад.

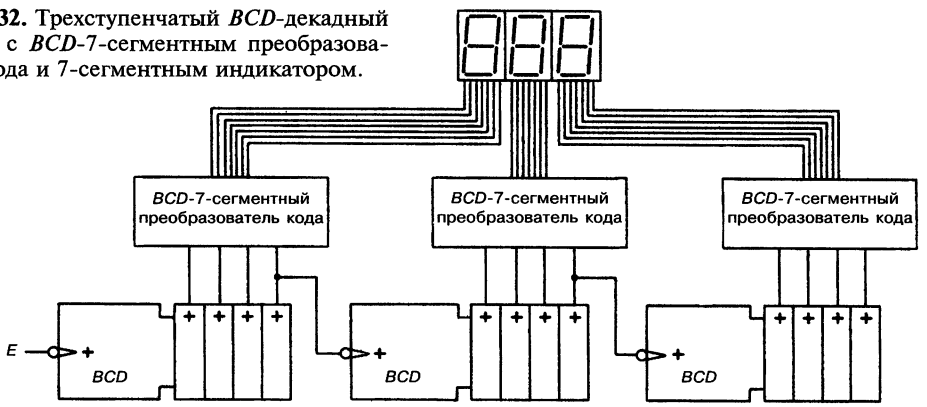
BCD-суммирующие счетчики считают от 0 до 9. Если необходимо считать дальше, нужен второй *BCD*-суммирующий счетчик. С двумя счетчиками этого вида можно считать до 99. Каждый счетчик считает одну декаду. Три счетчика считают до 999.

Декадные счетчики чаще всего строятся на двух или большем числе *BCD*-суммирующих счетчиков. Каждый счетчик имеет четыре выхода, на которых действует двоичное число, соответствующее десятичной цифре. Соединение счетчиков показано на рис. 10.31.

Если верхний счетчик сбрасывается на нуль, сигнал Q_D изменяется с 1 на 0. Обратный фронт управляет нижним счетчиком. Нижний счетчик переключается, если верхний счетчик сбрасывается в нуль. Он отвечает за переход во вторую декаду.

В декадных счетчиках численный результат может легко представляться десятичным

Рис. 10.32. Трехступенчатый *BCD*-декадный счетчик с *BCD*-7-сегментным преобразователем кода и 7-сегментным индикатором.



числом. Выходные сигналы каждого *BCD*-счетчика подаются на *BCD*-7-сегментный преобразователь кода. Десятичная цифра выдается на 7-сегментный индикатор (рис. 10.32).

10.2.3.2. Другие виды декадных счетчиков

Декадные счетчики могут строиться также из счетчиков, работающих в коде Айкена, в коде с избытком 3 или другом коде. Но такие счетчики используются редко. Десятичные счетчики этих видов строятся по тому же принципу, что и *BCD*-декадные счетчики.

10.2.4. Асинхронные счетчики по модулю *n*

10.2.4.1. Принцип действия счетчиков по модулю *n*

Для различного рода счетных задач в системах управления, обработки информации и измерения времени необходимы счетчики, ведущие счет до желаемого числа, затем сбрасывающиеся в исходное положение и считающие сначала, или останавливающиеся в ожидании новой команды на счет. Число, до которого ведется счет, может быть любое.

Такие счетчики называются **асинхронными счетчиками по модулю *n***. Буква *n* означает количество возможных состояний счетчика. *BCD*-счетчик, например, можно назвать счетчиком по модулю 10. Он считает только до 9. Однако с нулем включительно он имеет 10 возможных состояний счетчика.

10.2.4.2. Счетчики по модулю 5

Счетчики по модулю 5 должны считать до 4 и на 5-ом импульсе сбрасываться в нуль. Как выглядит схема такого счетчика?

Для простоты применим *T*-триггеры, хотя с тем же успехом можно было бы взять и универсальные *JK*-триггеры. Из *JK*-триггера, как известно, можно сделать *T*-триггер, если на входы *J* и *K* подать 1-сигнал. Для схемы необходимы три триггера (рис. 10.33). Счетчик может считать до 7. При переходе от 4 к 5 счетчик должен сброситься на нуль.

Возврат в исходное положение может осуществляться так же, как в *BCD*-счетчике. Если $Q_A = 1$ и $Q_C = 1$, то счетчик должен сбрасываться. В качестве сигнала сброса необходим 0-сигнал.

Выходы Q_A и Q_C связаны элементом И. Выход И-НЕ-элемента выдает сигнал сброса в 0, если Q_A и Q_C выдают 1-сигнал.

Условное обозначение счетчика по модулю 5 показано на рис. 10.34.

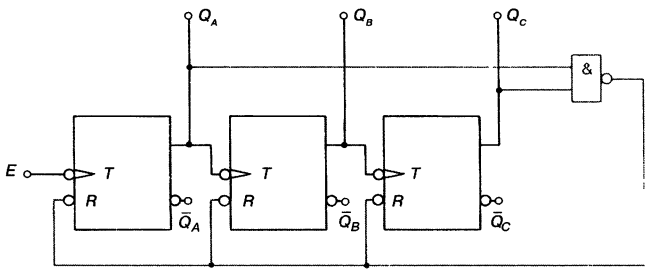


Рис. 10.33. Принципиальная схема счетчика по модулю 5.

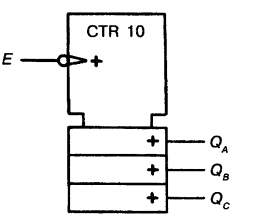


Рис. 10.34. Принципиальная схема счетчика по модулю 5, 5 счетных разрядов (CTR 5).

Задание

Как будет работать счетчик по модулю 5 (рис. 10.33), если для вывода результата вместо Q-выходов использовать $\bar{Q}$-выходы?

Для решения этого задания нужно сопоставить состояния счетчика в таблице (рис. 10.35). Таблица показывает, что счетчик считает от 7 до 3 и при появлении 2 сбрасывается. Он работает как вычитающий счетчик и имеет также 5 возможных состояний счетчика. Счетчик начинает считать с максимально возможного числа для имеющегося количества триггеров (здесь $7_{(10)} = 111_{(2)}$).

Десятичная цифра	Q_C	Q_B	Q_A	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	Десятичная цифра
0	0	0	0	1	1	1	7
1	0	0	1	1	1	0	6
2	0	1	0	1	0	1	5
3	0	1	1	1	0	0	4
4	1	0	0	0	1	1	3
Сброс	1	0	1	0	1	0	

Рис. 10.35. Сопоставление состояний счетчика.

10.2.4.3. Счетчики по модулю 60

Счетчики по модулю 60 требуются, например, для электронных часов. Секунды считаются от 0 до 60. Сколько триггеров требуется для такого счетчика? С пятью триггерами можно считать до 31, с 6 триггерами — до 63. Следовательно, необходимо шесть триггеров (рис. 10.36).

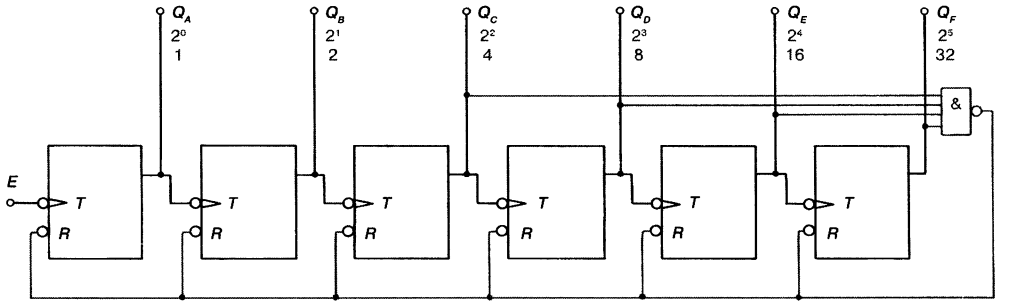
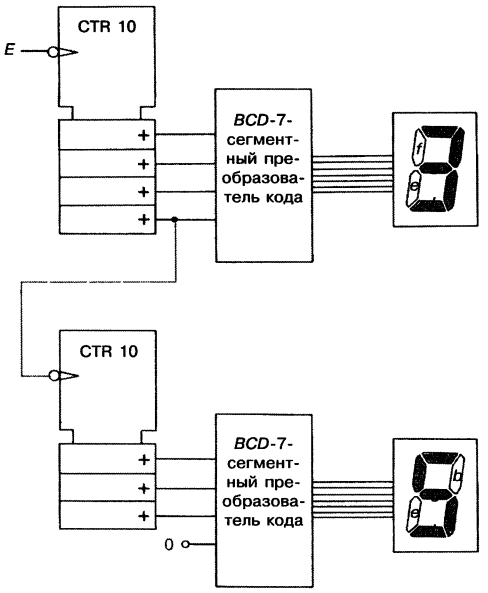


Рис. 10.36. Счетчик по модулю 60.



При появлении числа 60 счетчик должен быть сброшен на нуль. На выходах Q_C, Q_D, Q_E и Q_F должны действовать 1-сигналы. Из этих сигналов получается сигнал сброса. Счетчик годится для счета секунд, если они не должны отображаться в виде десятичных чисел.

Если секунды должны отображаться в виде десятичных чисел, то целесообразно единицы и десятки считать раздельно. Для единиц нужен счетчик по модулю 10, для де-

Рис. 10.37. Совместное включение счетчика по модулю 10 и счетчика по модулю 6 с преобразователем кода и 7-сегментным индикатором.

сятков — счетчик по модулю 6 (рис. 10.37). Выходные сигналы могут быть поданы через *BCD*-7-сегментный преобразователь кода на 7-сегментный индикатор.

10.2.4.4. Счетчики по модулю 13 с ожиданием команды сброса

Рассмотрим счетчики по модулю *n*, которые останавливают счет при достижении десятичного числа 12 и выдают на выход 1-сигнал. Счетчик должен сбрасываться при нажатии клавиши и начинать счет с начала.

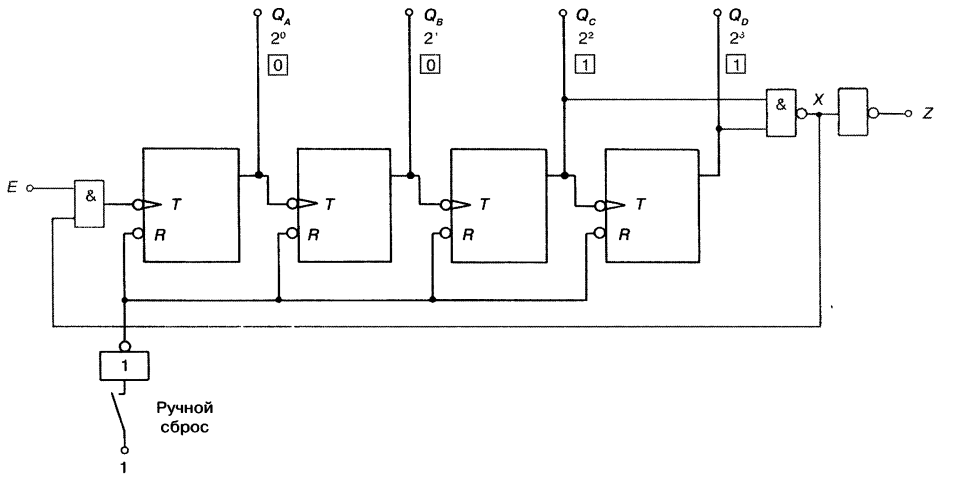


Рис. 10.38. Счетчик по модулю 13, останавливающий счет на числе 12.

Для счетчика необходимы 4 триггера. Вход E должен обладать возможностью запирания через элемент И. В качестве запрещающего сигнала используется 0-сигнал. Запрещающий сигнал получается из выходных сигналов Q_C и Q_D через И-НЕ-элемент (рис. 10.38). При достижении десятичного числа 12 ($Q_C = 1$, $Q_D = 1$) на выходе X И-НЕ-элемента действует 0-сигнал. Вход заперт. Счетчик останавливает счет. Одновременно на выходе Z появляется 1-сигнал. Нажатием клавиши счетчик сбрасывается. Вход открыт, так как теперь Q_C и Q_D имеют 0-сигналы. Счетчик начинает счет сначала. Так как возможны 13 состояний счетчика, включая нуль, то счетчик является счетчиком по модулю 13.

10.2.5. Асинхронные счетчики с произвольным коэффициентом пересчета

Асинхронные счетчики с произвольным коэффициентом пересчета — это счетчики, которые считают до определенного предварительно выбранного числа и затем останавливаются или начинают счет сначала. Счетчики могут быть как вычитающими так и суммирующими. Число, до которого нужно считать, устанавливается обычно вручную посредством переключателей. Оно может задаваться также вторым счетчиком. Такие счетчики также называются программируемыми.

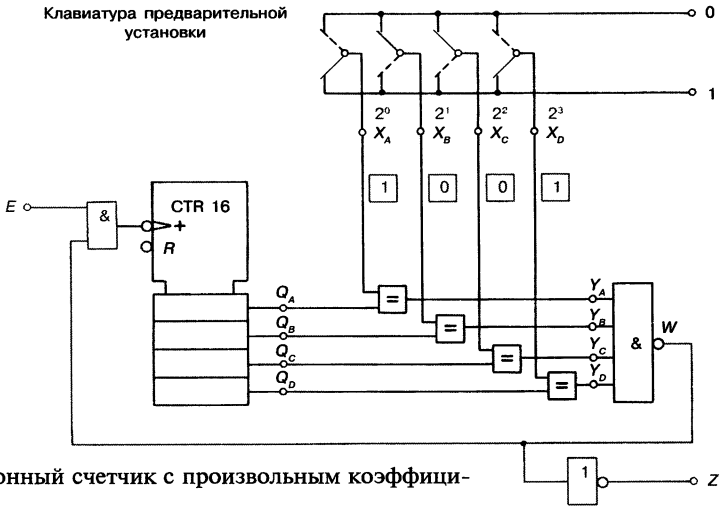


Рис. 10.39. Асинхронный счетчик с произвольным коэффициентом пересчета.

Асинхронный счетчик с произвольным коэффициентом пересчета показан на рис. 10.39. Число, до которого должен считать счетчик, устанавливается при помощи клавиатуры предварительной установки. Например, надо установить число 9. Число 9 в виде двоичного число задается на выходах X_A, X_B, X_C, X_D.

Выходные сигналы X_A, X_B, X_C, X_D сравниваются с выходными сигналами счетчика Q_A, Q_B, Q_C, Q_D. На выходах элементов эквивалентности 1-сигнал будет только тогда, когда оба входных сигнала равны. При $X_A = Q_A$, $X_B = Q_B$, $X_C = Q_C$ и $X_D = Q_D$ все выходы элементов эквивалентности равны 1. Тогда на выходе И-НЕ-элемента 0-сигнал.

0-сигнал на выходе W И-НЕ-элемента показывает, что число, до которого досчитал счетчик, и предустановленное число равны. 0-сигнал запирает вход. Счетчик прекращает счет. На выходе Z действует 1-сигнал, который может использоваться для различных задач управления.

Клавиатура предварительной установки может заменяться счетчиком, который устанавливается собственной системой управления.

Асинхронные счетчики с произвольным коэффициентом пересчета применяются, например, в современных копировальных автоматах. Желаемое число копий устанавливается вручную. Счетчик считает копии и останавливает копирование, если сделанное число копий совпадает с предварительно заданным числом.

10.2.6. Асинхронные счетчики для кода Айкена

Асинхронный счетчик, который считает в коде Айкена, представлен на рис. 10.40. Подобную схему достаточно трудно рассчитать. Однако ее можно синтезировать с помощью временных диаграмм (см. рис. 10.24).

10.2.7. Асинхронные счетчики для кода с избытком 3

Код с избытком 3 применяется часто. Однако счетчики, считающие в этом коде, встречаются в виде интегральных микросхем намного реже. На рис. 10.41 показана структура счетчика для этого кода.

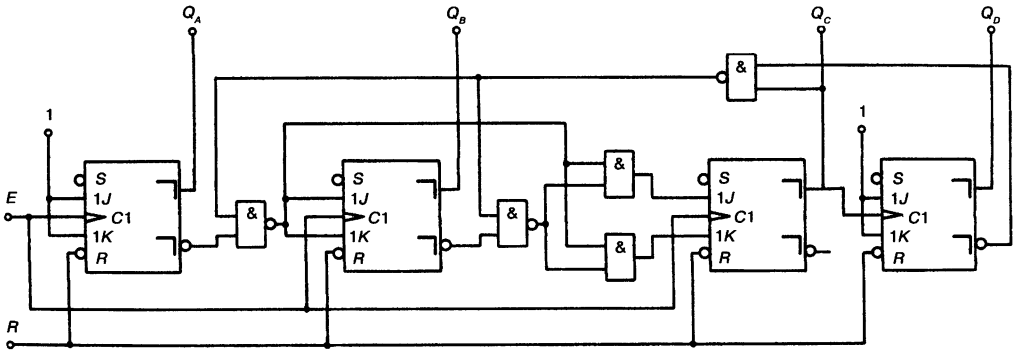


Рис. 10.40. Принципиальная схема асинхронного счетчика в коде Айкена.

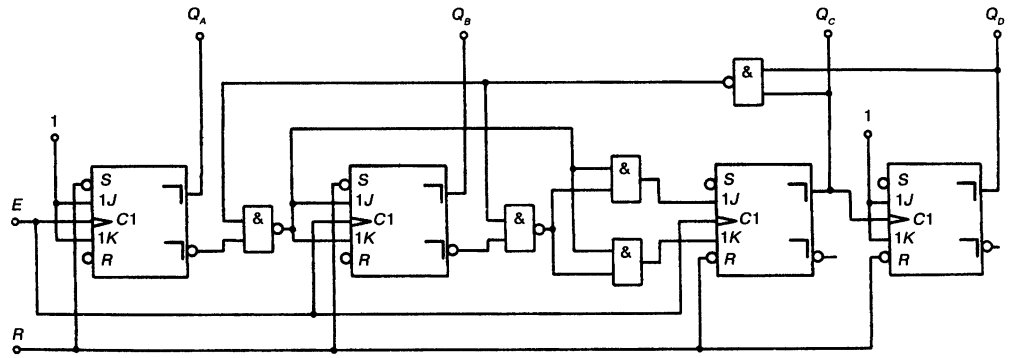


Рис. 10.41. Принципиальная схема асинхронного счетчика в коде с избытком 3.

10.3. Синхронные счетчики

10.3.1. Принцип действия

В рассмотренных до сих пор так называемых асинхронных счетчиках выход первого триггера управляет входом второго триггера, выход второго триггера — входом 3 триггера и так далее. Триггеры переключаются не одновременно, а по очереди. Задержка сигнала определяется быстродействием триггеров. Для триггеров ТТЛ-семейства время распространения сигнала составляет от 30 нс до 50 нс. На это время запаздывают сигналы на выходе триггеров относительно друг друга. Запаздывание увеличивается от триггера к триггеру. Для 12-битового асинхронного счетчика выход последнего триггера запаздывает по отношению к входу первого триггера примерно на 600 нс. На высоких частотах это приводит к ошибкам.

Если счетчик должен работать с высокой частотой, то запаздывание сигнала от триггера к триггеру недопустимо. Этого можно достигнуть только в том случае, если все триггеры, которые должны переключиться, сделают это одновременно, то есть будут управляться совместным сигналом синхронизации. Счетчики, работающие по этому принципу, называются **синхронными счетчиками**.

В синхронных счетчиках триггеры переключаются одновременно общим сигналом синхронизации.

Информация о том, должен триггер переключаться или нет, должна подаваться на триггеры до сигнала синхронизации. Поэтому кроме тактирующего входа необходимы дополнительные входы. Следовательно, синхронные счетчики не могут строиться на базе *T*-триггеров.

Синхронные счетчики чаще всего строятся на базе *JK*-триггеров.

Особенно надежны *JK*-MASTER-SLAVE-триггеры (см. разд. 7.5.7). *RS*-триггеры также могут использоваться. Однако схема с ними получается более сложная.

10.3.2. Синхронные двоичные счетчики

Синхронные двоичные счетчики считают в двоичной системе счисления. Они бывают суммирующими, вычитающими и реверсивными.

10.3.2.1 Двоичные суммирующие счетчики

Схему синхронного двоичного суммирующего счетчика можно рассчитать (разд. 10.3.3). Ее также можно синтезировать путем логического рассуждения. На рис. 10.42 изображены 4 *JK*-триггера. Все они объединены общим тактом синхронизации.

J и *K*-входы должны быть соединены таким образом, чтобы счетчик считал на увеличение в двоичной системе счисления. Как выглядит это соединение? Для ответа на этот вопрос рассмотрим временные диаграммы двоичного суммирующего счетчика на рис. 10.43. Триггер *A* должен переключаться с каждым обратным фронтом синхроимпульса. На входы J_A и K_A следует подать 1-сигналы.

Триггер *B* может переключаться тактом только в том случае, если триггер *A* установлен, то есть $Q_A = 1$. Выход Q_A должен соединяться с J_B и K_B (серым на рис. 10.42).

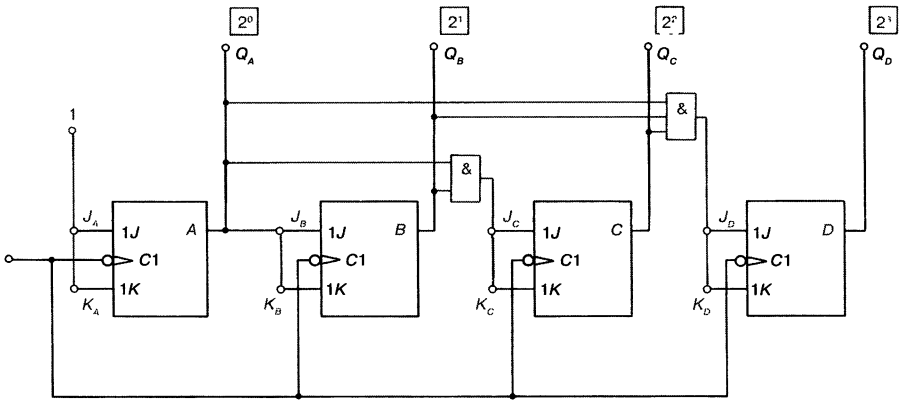


Рис. 10.42. Структура 4-битового двоичного синхронного суммирующего счетчика.

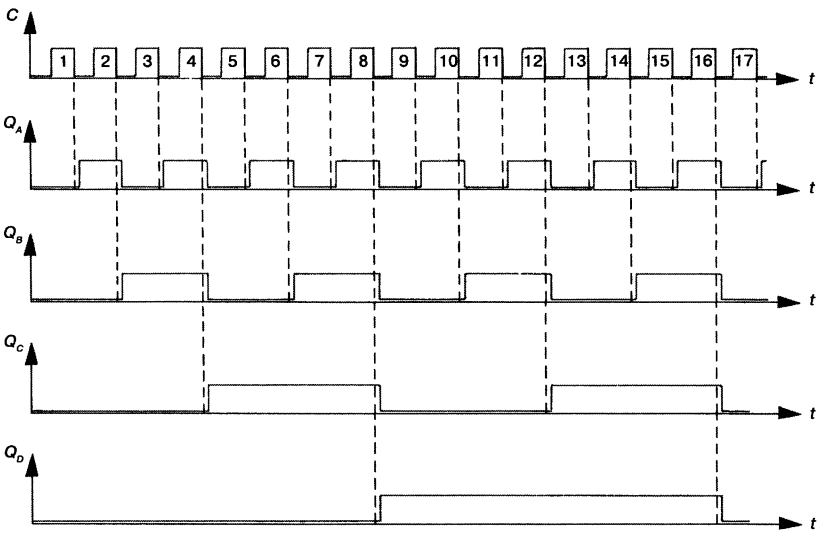


Рис. 10.43. Временная диаграмма 4-битового двоичного суммирующего счетчика.

При каких условиях переключается триггер C? Из рис. 10.43 следует, что триггер может переключаться только тогда, когда установлены триггер A и триггер B, то есть Q_A =1 и Q_B =1. Выходы Q_A =1 и Q_B =1 подаются на элемент И и соединяются с J_B и K_C (серым на рис. 10.42).

Условия, при которых переключается триггер D, можно также узнать из временной диаграммы на рис. 10.43. Триггер D может переключаться только тогда, когда $Q_A = 1$, $Q_B = 1$ и $Q_C = 1$. Выходы Q_A, Q_B и Q_C объединяются элементом И. Выход И-элемента связывается с J_D и K_D (серым на рис. 10.42). Можно было бы использовать объединение Q_A и Q_B первым И-элементом, тогда получилось бы объединение $Q_A \wedge Q_B \wedge Q_C$ одним элементом И с двумя входами.

Если необходим синхронный двоичный суммирующий счетчик с числом битов более 4, то его можно построить по принципу, изображенному на рис. 10.42. Для JK-триггеров действуют следующие правила:

> В синхронном двоичном суммирующем счетчике J и K входы каждого триггера нужно соединять друг с другом.

> В первом триггере на входы подается 1-сигнал. Каждый следующий триггер получает входной сигнал в виде результата логического умножения И Q-выходов всех предыдущих триггеров.

Задание

Синтезировать схему 5-битового синхронного двоичного суммирующего счетчика. Счетчик должен строиться на JK-MASTER-SLAVE-триггерах, которые принимают сигналы с передним фронтом и передают на выходы с обратным. Нарисовать временные диаграммы.

Для синхронных счетчиков способ управления фронтами импульса имеет второстепенное значения. Структура счетчика одинакова, переключаются

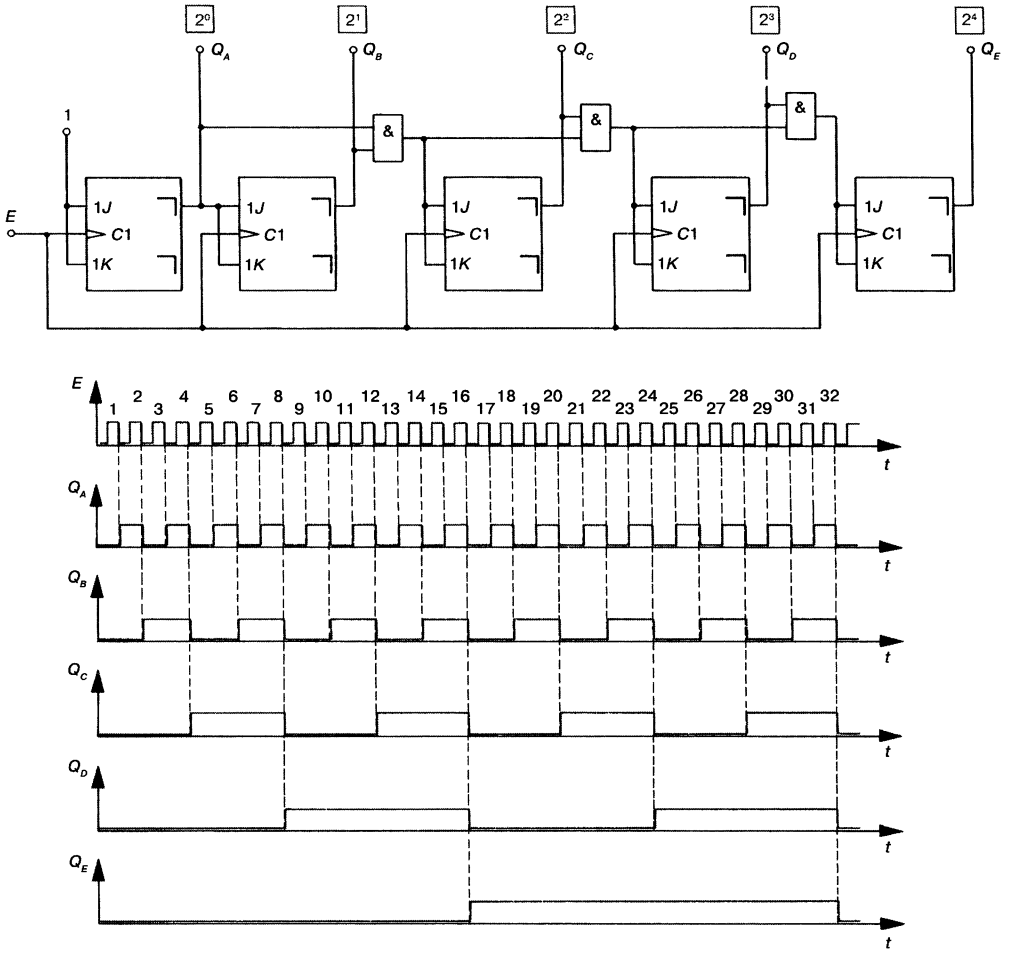


Рис. 10.44. Структура 5-битового двоичного синхронного суммирующего счетчика и его временные диаграммы.

ли триггеры с передним или обратным фронтом импульса. В зависимости от переключающего фронта смещаются только временные диаграммы, так как меняется точка переключения. Если используются *JK*-триггеры, которые принимают сигналы с передним фронтом и передают на выходы с обратным, то получается такая же временная диаграмма, как и в случае триггеров, которые переключаются с обратным фронтом. Искомая схема и соответствующие временные диаграммы представлены на рис. 10.44.

10.3.2.2. Двоичные вычитающие счетчики

Двоичный вычитающий счетчик можно построить на базе двоичного суммирующего счетчика (см. рис. 10.26). Вместо Q-выходов для подключения J- и K-входов используются $\bar{Q}$-выходы. В остальном структуры счетчиков одинаковы. На входы J и K первого триггера подается 1-сигнал. Схема синхронного 4-битового двоичного вычитающего счетчика показана на рис. 10.45.

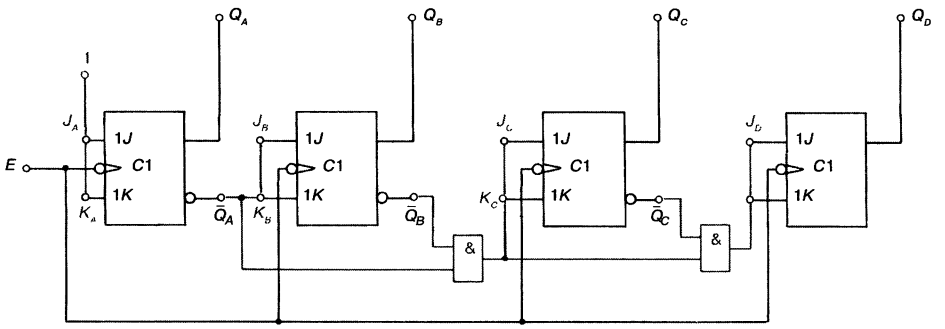


Рис. 10.45. Структура 4-битового двоичного синхронного вычитающего счетчика.

Десятичное число	Q_D	Q_C	Q_B	Q_A	$\bar{Q}_D$	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	Десятичное число
0	0	0	0	0	1	1	1	1	15
1	0	0	0	1	1	1	1	0	14
2	0	0	1	0	1	1	0	1	13
3	0	0	1	1	1	1	0	0	12
4	0	1	0	0	1	0	1	1	11
5	0	1	0	1	1	0	1	0	10
6	0	1	1	0	1	0	0	1	9
7	0	1	1	1	1	0	0	0	8
8	1	0	0	0	0	1	1	1	7
9	1	0	0	1	0	1	1	0	6
10	1	0	1	0	0	1	0	1	5
11	1	0	1	1	0	1	0	0	4
12	1	1	0	0	0	0	1	1	3
13	1	1	0	1	0	0	1	0	2
14	1	1	1	0	0	0	0	1	1
15	1	1	1	1	0	0	0	0	0

Рис. 10.46. Сопоставление выходных сигналов при применении Q-выходов и $\bar{Q}$-выходов.

Нет необходимости разрабатывать отдельную схему для синхронного двоичного вычитающего счетчика. Из любого суммирующего счетчика легко можно получить вычитающий.

Нужно лишь для вывода результата использовать инверсные $\bar{Q}$-выходы. Если $\bar{Q}$-выходы, как это бывает в интегральных микросхемах, недоступны, надо просто инвертировать прямые Q-выходы. Выходные сигналы при применении Q-выходов и $\bar{Q}$-выходов в качестве результирующих сопоставлены на рис. 10.46.

Счетчик на рис. 10.45 поможет построить синхронный реверсивный счетчик.

10.3.2.3. Двоичные реверсивные счетчики

Синхронные реверсивные двоичные счетчики можно построить как из синхронных двоичных суммирующих счетчиков (см. рис. 10.42), так и из синхронных двоичных вычитающих счетчиков. В первом случае J- и K-входы подключаются через Q-выходы, во втором — через инверсные $\bar{Q}$-выходы.

Для реверса необходимо переключать между Q- и $\bar{Q}$-выходами. Возможная схема представлена на рис. 10.47.

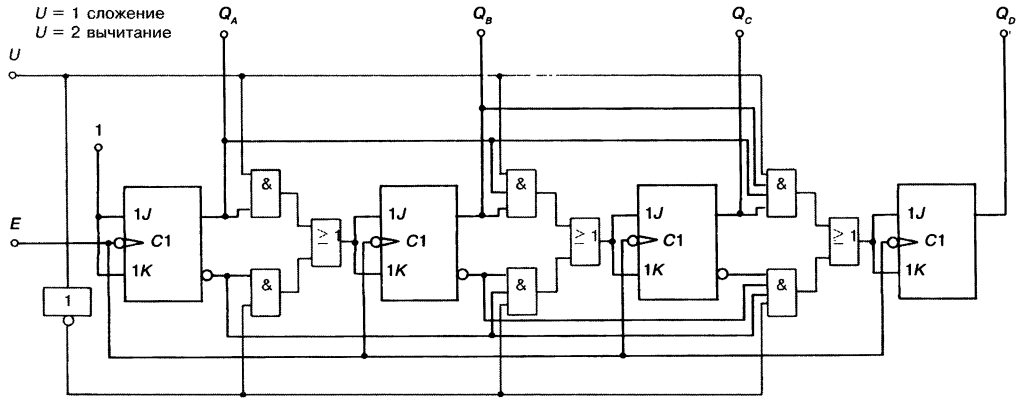


Рис. 10.47. Структура 4-битового двоичного синхронного реверсивного счетчика.

10.3.3. Расчет синхронных счетчиков

10.3.3.1. Методики расчета

Для расчета синхронных счетчиков существуют различные методы. Преимущество представленного ниже метода в его наглядности. Все шаги можно легко объяснить.

Расчет синхронного счетчика происходит в 5 шагов:

1. Составление таблицы истинности, из которой следует желаемая функция счетчика.
2. Составление и упрощение уравнений счетчика.
3. Определение характеристических уравнений применяемых триггеров.
4. Определение логических связей сравнением коэффициентов.
5. Составление схемы по найденным логическим связям.

Из таблицы истинности должен быть ясен порядок следования выходных сигналов счетчика. У 4-битового счетчика выходы Q_A, Q_B, Q_C, Q_D. В момент времени t_n (перед текущим тактом) $Q_A = 0$, $Q_B = 0$, $Q_C = 0$ и $Q_D = 0$. После текущего такта в момент времени t_{n+1} часть выходов меняет свое состояние. Какое изменение выходного сигнала желательно? Какой выходной сигнал должен выдавать счетчик? В двоичном суммирующем счетчике теперь должно быть: $Q_A = 1$, $Q_B = 0$, $Q_C = 0$ и $Q_D = 0$. Таблица истинности должна иметь столбцы как для Q_A, Q_B, Q_C, и Q_D в момент времени t_n, так и столбцы для Q_A, Q_B, Q_C, и Q_D в момент времени t_{n+1}.

Уравнения для поставленной задачи получаются из нормальной формы ИЛИ для $Q_{A(n+1)}$, $Q_{B(n+1)}$, $Q_{C(n+1)}$, $Q_{D(n+1)}$. Нормальную форму ИЛИ следует упростить с помощью диаграмм Карно. Для каждого триггера получается свое уравнение. Все уравнения вместе воспроизводят в форме алгебры логики содержание таблицы истинности.

Характеристическое уравнение описывает принцип действия триггера в форме алгебры логики. Если, например, используются *JK*-триггеры, то нужно записать характеристическое уравнение для *JK*-триггеров (см. разд. 7.7).

Характеристические уравнения триггеров и упрощенные уравнения поставленной задачи сравнивают друг с другом. Для 4 триггеров это сравнение проводят четырежды, так как каждое из 4 уравнений постав-

ленной задачи должно сравниваться с характеристическим уравнением. В результате сравнения получают логические связи для 4 *J*-входов и для 4 *K*-входов.

Имея уравнения логических связей, можно изобразить схему. Синтез схемы после нахождения уравнений связей не представляет никаких трудностей.

10.3.3.2. Пример расчета

Поясним методику расчета на примере. Рассчитаем 4-битовый синхронный суммирующий счетчик на базе *JK*-триггеров.

1. Шаг:

Составление таблицы истинности.

Для 4-битового счетчика нужно 4 триггера. Обозначим выходы этих триггеров Q_A, Q_B, Q_C, и Q_D. Для каждого из этих выходов предусмотрен столбец t_n и столбец t_{n+1} (рис. 10.48).

Начальное значение тактового счетчика в момент времени t_n равно 0000, что соответствует десятичному 0. После подачи импульса в момент времени t_{n+1} счетчик должен увеличить свое значение на 1. Он должен показывать десятичную 1. Соответствующее двоичное число 0001. Это содержание 1-й строки таблицы истинности на рис.10.48.

Рассмотрим 2-ю строку таблицы истинности. В момент времени t_n счетчик показывает двоичное число 0001. В момент времени t_{n+1} он должен показать двоичное число после очередного такта. Это двоичное число 0010.

В 3-й строке таблицы истинности в момент времени t_n счетчик показывает $0010_{(2)}$. После следующего такта счетчик должен показывать $0011_{(2)}$ (в момент времени t_{n+1}). Так строка за строкой заполняется таблица истинности, до значения счетчика $1111_{(2)}$. После следующего такта он должен

	t_n				t_{n+1}				
Десятичное число	2^3 Q_D	2^2 Q_C	2^1 Q_B	2^0 Q_A	2^3 $\bar{Q}_D$	2^2 $\bar{Q}_C$	2^1 $\bar{Q}_B$	2^0 $\bar{Q}_A$	Десятичное число
0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1	0	2
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	1	0	0	4
4	0	1	0	0	0	1	0	1	5
5	0	1	0	1	0	1	1	0	6
6	0	1	1	0	0	1	1	1	7
7	0	1	1	1	1	0	0	0	8
8	1	0	0	0	1	0	0	1	9
9	1	0	0	1	1	0	1	0	10
10	1	0	1	0	1	0	1	1	11
11	1	0	1	1	1	1	0	0	12
12	1	1	0	0	1	1	0	1	13
13	1	1	0	1	1	1	1	0	14
14	1	1	1	0	1	1	1	1	15
15	1	1	1	1	0	0	0	0	0

Рис. 10.48. Таблица истинности 4-битового двоичного синхронного суммирующего счетчика.

показать $0000_{(2)}$ и начать счет сначала. Итак, таблица истинности заполнена. Она однозначно описывает желаемую функцию счетчика.

2. Шаг:

Составление и упрощение уравнений счетчика.

Прежде всего, нужно составить нормальную форму ИЛИ $Q_{A(n+1)}$ (подробнее см. гл. 5). Нормальная форма ИЛИ состоит из 8 полных конъюнкций:

$$Q_{A(n+1)} = \Big[\underset{①}{\left(\bar{A}\wedge\bar{B}\wedge\bar{C}\wedge\bar{D}\right)}\vee\underset{②}{\left(\bar{A}\wedge B\wedge\bar{C}\wedge\bar{D}\right)}\vee\underset{③}{\left(\bar{A}\wedge\bar{B}\wedge C\wedge\bar{D}\right)}\vee$$
$$\vee\underset{④}{\left(\bar{A}\wedge B\wedge C\wedge\bar{D}\right)}\vee\underset{⑤}{\left(\bar{A}\wedge\bar{B}\wedge\bar{C}\wedge D\right)}\vee\underset{⑥}{\left(\bar{A}\wedge B\wedge\bar{C}\wedge D\right)}\vee$$
$$\vee\underset{⑦}{\left(\bar{A}\wedge\bar{B}\wedge C\wedge D\right)}\vee\underset{⑧}{\left(\bar{A}\wedge B\wedge C\wedge D\right)}\Big]_n .$$

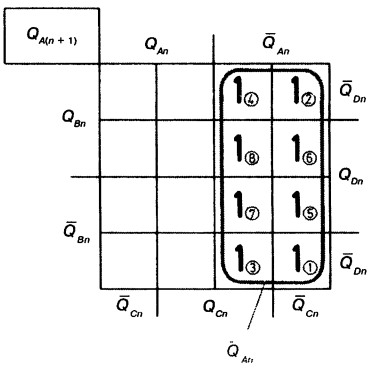


Рис. 10.49. Карта Карно нормальной формы ИЛИ $Q_{A(n+1)}$.

Эта нормальная форма ИЛИ упрощается с помощью диаграммы Карно. Соответствующая диаграмма показана на рис. 10.49.

Получается группа из 8 элементов. Упрощенная форма ИЛИ:

$$Q_{A(n+1)} = \bar{Q}_{An}$$

Это первое уравнение счетчика.

Далее нужно образовать нормальную форму ИЛИ $Q_{B(n+1)}$ Она также состоит из 8 полных конъюнкций и может быть занесена в карту Карно (рис. 10.50). Переменные Q_{An}, Q_{Bn}, Q_{Cn}, Q_{Dn} записываются как Q_A, Q_B, Q_C, Q_D. Упрощенная запись не вносит путаницы.

Из карты Карно записывается уравнение:

$$Q_{B(n+1)} = \left(Q_A\wedge\bar{Q}_B\right)\vee\left(\bar{Q}_A\wedge Q_B\right)$$

Это второе уравнение счетчика.

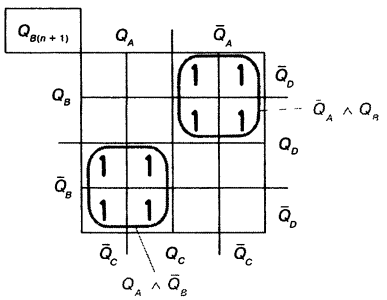


Рис. 10.50. Карта Карно нормальной формы ИЛИ $Q_{B(n+1)}$. Индексы при n для простоты убраны.

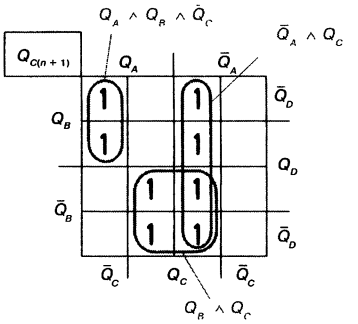


Рис. 10.51. Карта Карно нормальной формы ИЛИ $Q_{C(n+1)}$.

Уравнение поставленной задачи для $Q_{C(n+1)}$ находится аналогичным способом. Нормальная форма ИЛИ также состоит из 8 полных конъюнкций и может быть занесена в карту Карно (рис. 10.51). Уравнение проекта:

$$Q_{C(n+1)} = \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right) \vee \left(\bar{Q}_A \wedge Q_C\right) \vee \left(\bar{Q}_B \wedge Q_C\right).$$

Его можно упростить:

$$Q_{C(n+1)} = \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right) \vee \left[Q_C \wedge \left(\bar{Q}_A \vee \bar{Q}_B\right)\right];$$

$$\boxed{Q_{C(n+1)} = \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right) \vee \left(\overline{Q_A \wedge Q_B} \wedge Q_C\right).}$$

Нормальная форма ИЛИ для $Q_{D(n+1)}$ показана на рис. 10.52. Упрощенное выражение:

$$Q_{D(n+1)} = \left(Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D\right) \vee \left(\bar{Q}_A \wedge Q_D\right) \vee \left(\bar{Q}_B \wedge Q_D\right) \vee \left(\bar{Q}_C \wedge Q_D\right).$$

Переменную Q_D можно вынести за скобки:

$$Q_{D(n+1)} = \left(Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D\right) \vee \left[\left(Q_D \wedge (\bar{Q}_A \vee \bar{Q}_B \vee \bar{Q}_C\right)\right].$$

$$\boxed{Q_{D(n+1)} = \left(Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D\right) \vee \left(\overline{Q_A \wedge Q_B \wedge Q_C} \wedge Q_D\right)}$$

Это последнее из четырех уравнений счетчика.

3. Шаг:

Определение характеристических уравнений для примененных триггеров.

Как выводится характеристическое уравнение триггера определенного типа, подробно описано в разд. 7.7. Для *JK*-триггеров согласно обозначениям на рис. 10.53 характеристическое уравнение:

$$Q_{(n+1)} = \left[\left(J \wedge \bar{Q}\right) \vee \left(\bar{K} \wedge Q\right)\right]_n.$$

В правой части уравнения для упрощения опущен индекс *n*.

$$Q_{(n+1)} = \left(J \wedge \bar{Q}\right) \vee \left(\bar{K} \wedge Q\right).$$

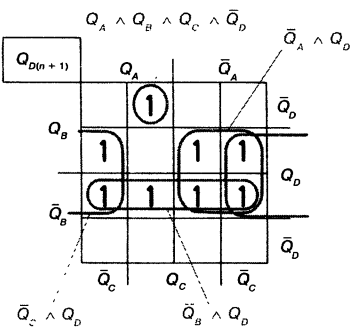


Рис. 10.52. Карта Карно нормальной формы ИЛИ $Q_{D(n+1)}$.

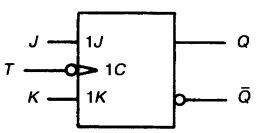


Рис. 10.53. *JK*-триггер.

Для 4 триггеров получаем 4 характеристических уравнения:

$$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right);$$

$$Q_{B(n+1)} = \left(J_B \wedge \bar{Q}_B\right) \vee \left(\bar{K}_B \wedge Q_B\right);$$

$$Q_{C(n+1)} = \left(J_C \wedge \bar{Q}_C\right) \vee \left(\bar{K}_C \wedge Q_C\right);$$

$$Q_{D(n+1)} = \left(J_D \wedge \bar{Q}_D\right) \vee \left(\bar{K}_D \wedge Q_D\right).$$

4. Шаг:

Определение логических связей сравнением коэффициентов.

Полученные в п. 2 уравнения по очереди сравниваются с характеристическими уравнениями:

$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right)$ Характеристическое уравнение триггера;

$Q_{A(n+1)} = \bar{Q}_A$ Уравнение счетчика.

Какое значение должен иметь J_A, чтобы из уравнения выделить $\bar{Q}_A$? J_A должен быть равен 1.

Какое значение должно иметь $\bar{K}_A$, чтобы из удалить из уравнения скобку, содержащую $\bar{K}_A$? $\bar{K}_A$ должен быть равен 0.

$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right)$ Характеристическое уравнение триггера;

$Q_{A(n+1)} = \left(1 \wedge \bar{Q}_A\right) \vee \left(0 \wedge Q_A\right)$ Уравнение счетчика.

Для J_A и K_A получаются следующие уравнения:

$$\bar{K}_A = 0$$

$$\boxed{J_A = 1,} \qquad \boxed{K_A = 1}$$

Это первые два уравнения логической связи. Они получены для первого триггера *A*.

Для триггера *B* получаются следующие уравнения:

$Q_{B(n+1)} = \left(J_B \wedge \bar{Q}_B\right) \vee \left(\bar{K}_B \wedge Q_B\right)$ Характеристическое уравнение триггера;

$Q_{B(n+1)} = \left(Q_A \wedge \bar{Q}_B\right) \vee \left(\bar{Q}_B \wedge Q_B\right)$ Уравнение счетчика.

$\bar{K}_B = Q_A$

$$\boxed{J_B = Q_A} \qquad \boxed{K_B = Q_A}$$

Аналогично получают уравнения для триггера C:

$$Q_{C(n+1)} = \left(J_C \wedge \bar{Q}_C\right) \vee \left(\bar{K}_C \wedge Q_C\right)$$ Характеристическое уравнение триггера;

$$Q_{C(n+1)} = (\overline{Q_A \wedge Q_B} \wedge Q_C) \vee (\overline{Q_A \wedge Q_B} \wedge Q_C)$$ Уравнение счетчика.

$$\bar{K}_C = \overline{Q_A \wedge Q_B}$$

$$\boxed{J_C = Q_A \wedge Q_B} \qquad \boxed{K_C = Q_A \wedge Q_B}$$

И для триггера D:

$$Q_{D(n+1)} = \left(J_D \wedge \bar{Q}_D\right) \vee \left(\bar{K}_D \wedge Q_D\right)$$ Характеристическое уравнение триггера;

$$Q_{D(n+1)} = (\overline{Q_A \wedge Q_B} \wedge Q_C \wedge \overline{Q_D}) \vee (\overline{Q_A \wedge Q_B \wedge Q_C} \wedge Q_D)$$ Уравнение счетчика.

$$\bar{K}_D = \overline{Q_A \wedge Q_B \wedge Q_C};$$

$$\boxed{J_D = Q_A \wedge Q_B \wedge Q_C} \qquad \boxed{K_D = Q_A \wedge Q_B \wedge Q_C}$$

Итак, найдены все уравнения логических связей.

5. Шаг:

Нарисовать схему по найденным уравнениям.
Найденные уравнения записываются вместе:

$$J_A = 1, \qquad K_A = 1;$$

$$J_B = Q_A, \qquad K_B = Q_A;$$

$$J_C = Q_A \wedge Q_B, \qquad K_C = Q_A \wedge Q_B;$$

$$J_D = Q_A \wedge Q_B \wedge Q_C, \qquad K_D = Q_A \wedge Q_B \wedge Q_C.$$

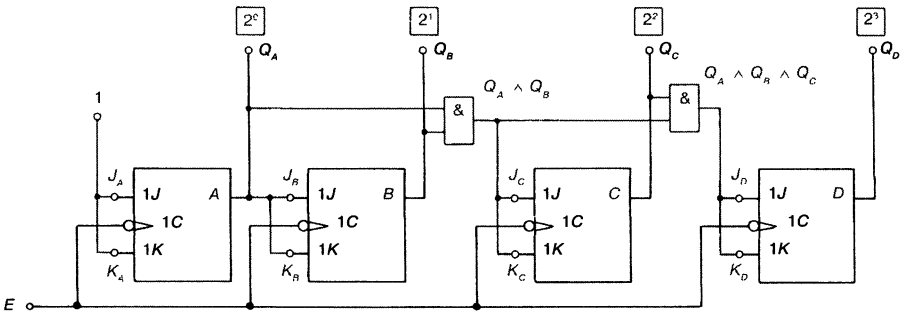


Рис. 10.54. Схема счетчика, построенная по результатам расчета.

Уравнения для *J*- и *K*-входа каждого триггера одинаковы. Поэтому входы *J* и *K* могут управляться совместно.

Схема, которая получается согласно уравнениям, представлена на рис. 10.54. В результате расчета мы получили такую же схему, которую построили в результате анализа временных диаграмм (рис. 10.43).

10.3.4. Синхронные *BCD*-счетчики

10.3.4.1. Расчет синхронного суммирующего BCD-счетчика

Согласно методике расчета (подразд. 10.3.3) можно рассчитать синхронный суммирующий *BCD*-счетчик.

1. Шаг:

Составление таблицы истинности.

Псевдотетрады можно не записывать. Их ячейки на карте Карно можно обозначать символом *X*. При образовании группы *X* может рассматриваться как 0, так и 1 на выбор разработчика.

	t_n				t_{n+1}				
Десятичная цифра	2^3 Q_D	2^2 Q_C	2^1 Q_B	2^0 Q_A	2^3 Q_D	2^2 Q_C	2^1 Q_B	2^0 Q_A	Десятичная цифра
0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1	0	2
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	1	0	0	4
4	0	1	0	0	0	1	0	1	5
5	0	1	0	1	0	1	1	0	6
6	0	1	1	0	0	1	1	1	7
7	0	1	1	1	1	0	0	0	8
8	1	0	0	0	1	0	0	1	9
9	1	0	0	1	0	0	0	0	0
(10)	1	0	1	0	×	×	×	×	Псевдотетрады
(11)	1	0	1	1	×	×	×	×	
(12)	1	1	0	0	×	×	×	×	
(13)	1	1	0	1	×	×	×	×	
(14)	1	1	1	0	×	×	×	×	
(15)	1	1	1	1	×	×	×	×	

Рис. 10.55. Таблица истинности синхронного суммирующего *BCD*-счетчика.

2. Шаг:

Составление и упрощение уравнений счетчика.

Нормальные формы ИЛИ от $Q_{A(n+1)}$, $Q_{B(n+1)}$, $Q_{C(n+1)}$, $Q_{D(n+1)}$ записываются в карту Карно и упрощаются (рис. 10.56). Упрощенные уравнения счетчика:

$$Q_{A(n+1)} = \bar{Q}_A;$$

$$Q_{B(n+1)} = \left(\bar{Q}_A \wedge Q_B\right) \vee \left(Q_A \wedge \bar{Q}_B \wedge \bar{Q}_D\right);$$

$$Q_{C(n+1)} = \left(\bar{Q}_A \wedge Q_C\right) \vee \left(\bar{Q}_B \wedge Q_C\right) \vee \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right);$$

$$Q_{D(n+1)} = \left(\bar{Q}_A \wedge Q_D\right) \vee \left(Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D\right).$$

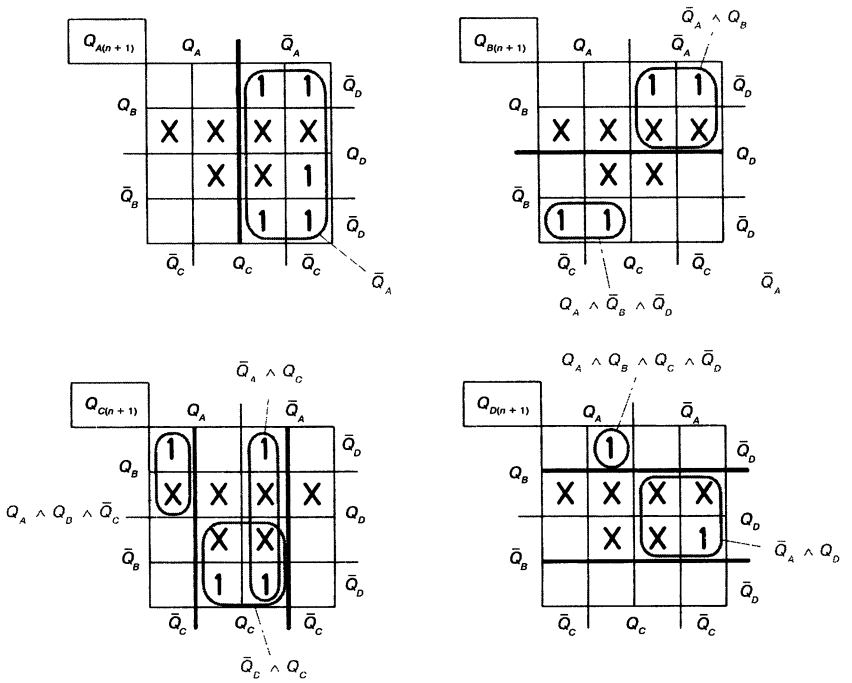


Рис. 10.56. Карта Карно для определения уравнений счетчика.

Рассмотрим карту Карно для $Q_{D(n+1)}$. Видно, что полная конъюнкция $Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D$ не упрощается группировкой. Вследствие группировки выпала бы переменная Q_D, которая нужна для сравнения коэффициентов.

Общее правило: в карте Карно для $Q_{A(n+1)}$ не может выпадать переменная величина Q_A. В карте Карно для $Q_{B(n+1)}$ не может выпадать переменная величина Q_B. В карте Карно для $Q_{C(n+1)}$ не может выпадать переменная величина Q_C. В карте Карно для $Q_{D(n+1)}$ не может выпадать переменная величина Q_D.

При образовании групп следует учитывать границы, показанные на рис. 10.56 жирными линиями. При образовании групп жирные линии не должны пересекаться. Таким образом целенаправленно происходит отказ от максимального упрощения.

3. Шаг:

Определение характеристических уравнений для применяемых триггеров.

Применяются *JK*-триггеры. Характеристическое уравнение для *JK*-триггеров, опуская индекс *n*, выглядит так:

$$Q_{(n+1)} = \left(J \wedge \bar{Q}\right) \vee \left(\bar{K} \wedge Q\right).$$

Для 4 триггеров *A*, *B*, *C* и *D* характеристические уравнения:

$$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right);$$

$$Q_{B(n+1)} = \left(J_B \wedge \bar{Q}_B\right) \vee \left(\bar{K}_B \wedge Q_B\right);$$

$$Q_{C(n+1)} = \left(J_C \wedge \bar{Q}_C\right) \vee \left(\bar{K}_C \wedge Q_C\right);$$
$$Q_{D(n+1)} = \left(J_D \wedge \bar{Q}_D\right) \vee \left(\bar{K}_D \wedge Q_D\right).$$

4. Шаг:

Определение логических связей сравнением коэффициентов.

Триггер A

$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right)$; Характеристическое уравнение триггера;

$Q_{A(n+1)} = \bar{Q}_A$; Уравнение счетчика.

$Q_{A(n+1)} = \left(J_A \wedge \bar{Q}_A\right) \vee \left(\bar{K}_A \wedge Q_A\right)$;

$Q_{A(n+1)} = \left(1 \wedge \bar{Q}_A\right) \vee \left(0 \wedge Q_A\right)$;

$\bar{K}_A = 0$;

$$\boxed{J_A = 1} \qquad \boxed{K_A = 1}$$

Триггер B

$Q_{B(n+1)} = \left(J_B \wedge \bar{Q}_B\right) \vee \left(\bar{K}_B \wedge Q_B\right)$ Характеристическое уравнение триггера;

$Q_{B(n+1)} = \left(\bar{Q}_A \wedge Q_B\right) \vee \left(Q_A \wedge \bar{Q}_B \wedge \bar{Q}_D\right)$ Уравнение счетчика.

Перед сравнением коэффициентов перепишем уравнение счетчика в другом виде:

$Q_{B(n+1)} = (\underbrace{Q_A \wedge \bar{Q}_D}_{\Downarrow} \wedge \bar{Q}_B) \vee (\underbrace{\bar{Q}_A}_{\Downarrow} \wedge Q_B)$ Характеристическое уравнение триггера;

$Q_{B(n+1)} = \left(J_B \wedge \bar{Q}_B\right) \vee \left(\bar{K}_B \wedge Q_B\right)$; Уравнение счетчика.

$K_B = \bar{Q}_A$;

$$\boxed{J_B = Q_A \wedge \bar{Q}_D} \qquad \boxed{K_B = Q_A}$$

Триггер C

$Q_{C(n+1)} = \left(J_C \wedge \bar{Q}_C\right) \vee \left(\bar{K}_C \wedge Q_C\right)$ Характеристическое уравнение триггера;

$Q_{D(n+1)} = \left(\bar{Q}_A \wedge Q_C\right) \vee \left(\bar{Q}_B \wedge Q_C\right) \vee \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right)$ Уравнение счетчика.

Перед сравнением коэффициентов перепишем уравнение счетчика в другом виде:

$$Q_{C(n+1)} = \left(Q_A \wedge Q_B \wedge \bar{Q}_C\right) \vee Q_C \wedge \left(\bar{Q}_A \vee \bar{Q}_B\right);$$

$$Q_{C(n+1)} = (\underline{Q_A \wedge \bar{Q}_B} \wedge \bar{Q}_C) \vee (\overline{\underline{Q_A \wedge Q_B}} \wedge Q_C)$$

$$Q_{C(n+1)} = \left(J_C \wedge \bar{Q}_C\right) \vee \left(K_C \wedge Q_C\right);$$

$$K_C = \overline{Q_A \wedge Q_B};$$

$$\boxed{J_C = Q_A \wedge Q_B} \qquad \boxed{K_C = Q_A \wedge Q_B}$$

Триггер D

$Q_{D(n+1)} = \left(J_D \wedge \bar{Q}_D\right) \vee \left(\bar{K}_D \wedge Q_D\right);$ Характеристическое уравнение триггера;

$Q_{D(n+1)} = \left(\bar{Q}_A \wedge Q_D\right) \vee \left(Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D\right).$ Уравнение счетчика.

Перед сравнением коэффициентов перепишем уравнение счетчика в другом виде:

$$J_{D(n+1)} = (\underline{Q_A \wedge Q_B \wedge Q_C} \wedge \bar{Q}_D) \vee (\underline{Q_A} \wedge Q_D)$$

$$J_{D(n+1)} = \left(J_D \wedge \bar{Q}_D\right) \vee \left(K_D \wedge Q_D\right);$$

$$\bar{K}_D = \bar{Q}_A;$$

$$\boxed{J_D = Q_A \wedge Q_B \wedge Q_C} \qquad \boxed{K_D = Q_A}$$

Уравнения логических связей определены.

5. Шаг:

Построение схемы после определения уравнений логических связей.

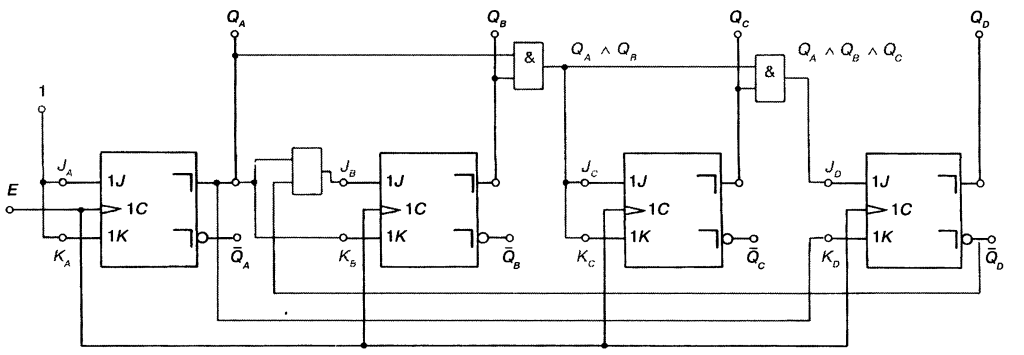


Рис. 10.57. Синхронный суммирующий *BCD*-счетчик.

Запишем вместе найденные уравнения логических связей:

$$J_A = 1, \qquad K_A = 1;$$
$$J_B = Q_A \wedge \bar{Q}_D, \qquad K_B = Q_A;$$
$$J_C = Q_A \wedge Q_B, \qquad K_C = Q_A \wedge Q_B;$$
$$J_D = Q_A \wedge Q_B \wedge Q_C, \qquad K_D = Q_A.$$

Триггеры *A*, *B*, *C* и *D* соединяются в соответствии с уравнениями связи. Схема рассчитанного счетчика изображена на рис. 10.57.

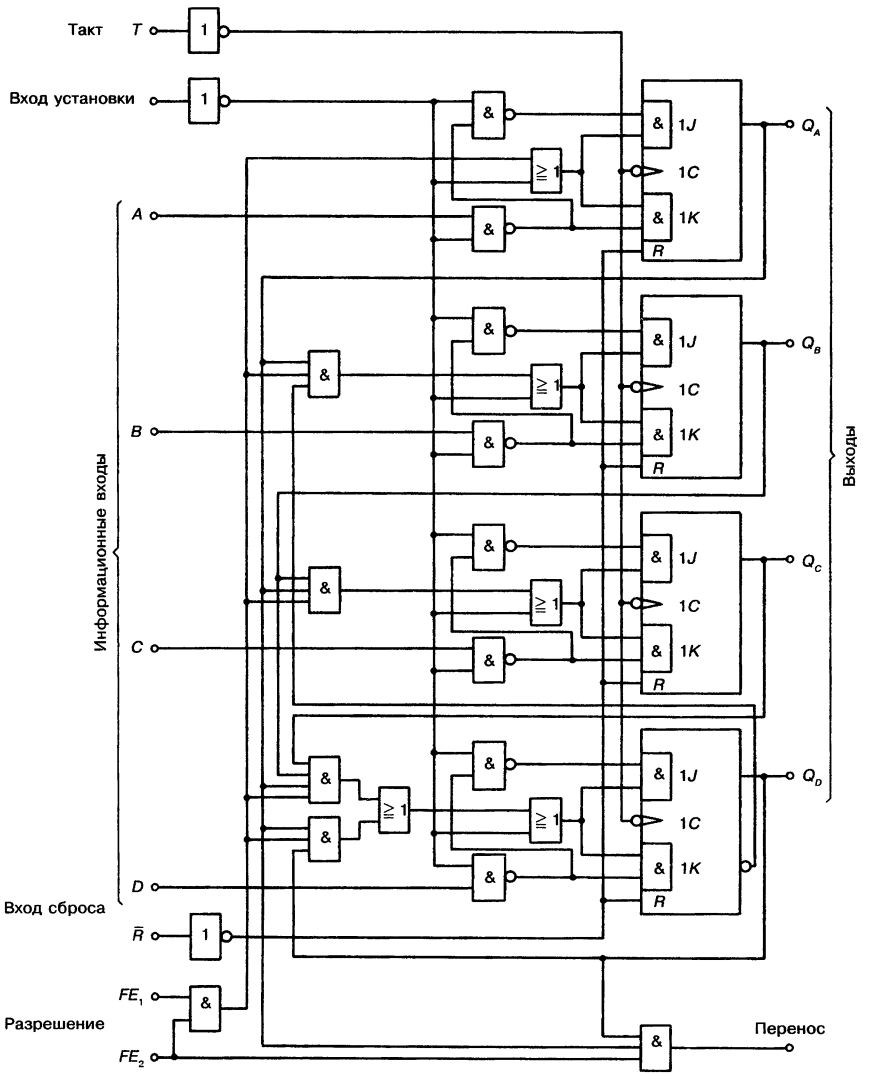


Рис. 10.58. Структура и цоколевка интегральной микросхемы FLJ401-74160 (синхронный суммирующий *BCD*-счетчик).

10.3.4.2. Интегральные синхронные суммирующие BCD-счетчики

Синхронные суммирующие *BCD*-счетчики применяются в большом количестве и поэтому выпускаются в виде интегральных микросхем. Микросхемы можно усложнять без значительного удорожания изделия. Интегральные *BCD*-счетчики часто имеют тактируемые или нетактируемые входы сброса. Также часто *BCD*-счетчики имеют возможность предварительного программирования, т. е. через специальные входы можно установить счетчик на начальное значение, с которого он начинает считать.

Типичной интегральной микросхемой является схема ТТЛ-семейства FLJ401-74160. Структура микросхемы представлена на рис. 10.58.

Вход сброса $\overline{R}$ независим от тактового сигнала. 0-сигнал на входе сброса сбрасывает счетчик.

Вход установки $\overline{S}$ работает с входами *A*, *B*, *C* и *D*. Предварительная установка счетчика через информационные входы возможна только при условии, что на входе установки действует 0-сигнал. Триггер *A* может быть установлен 1-сигналом на входе *A* и сброшен 0-сигналом на входе *A*. Также и триггеры *B*, *C* и *D* могут быть установлены и сброшены через соответствующие входы. Установка и сброс происходит синхронно с передним фронтом синхроимпульса.

Другими словами, при наличии сигнала установки сигналы на информационных входах *A*, *B*, *C* и *D* переходят в счетчик.

Особенную роль играют входы разрешения FE_1 и FE_2. Если на одном из входов разрешения действует 0-сигнал, то счетчик может быть предустановлен, но не может считать. Счет производится, если на обоих входах разрешения FE_1 и FE_2 действует 1-сигнал. Если на одном входе FE_2 действует 1-сигнал, то разрешен только перенос.

Счетчик FLJ-74160 является универсальным. Если какие-либо из его возможностей не востребованы, то можно не подключать соответствующие выходы.

10.3.5. Синхронный счетчик для кода с избытком 3

Синтезируем схему синхронного суммирующего счетчика, работающего в коде с избытком 3 на базе *JK*-триггеров, переключающихся с обратным фронтом (рис. 10.59). Схема должна быть рассчитана по алгоритму, рассмотренному в подразд. 10.3.3.

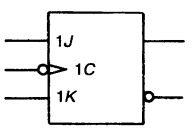


Рис. 10.59. *JK*-триггер.

1. Шаг:

Составление таблицы истинности.

На рис. 10.60 изображен код с избытком 3. После каждого такта счетчик должен быть выставлен на шаг вперед. На основании этого условия получается таблица истинности (рис. 10.61). Шесть неиспользуемых тетрад являются псевдотетрадами (рис. 10.62). Они могут быть не задействованы в коде с избытком 3 и пропущены в таблице истинности. Их ячейки в карте Карно обозначаются символом *X*. Согласно рис. 10.62 используются символы от X_a до X_f. При группировке *X* может на выбор считаться как 1 или как 0.

Десятичная цифра	D	C	B	A
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Рис. 10.60. Код с избытком 3.

Десятичная цифра	t_n				t_{n+1}				Десятичная цифра
	Q_D	Q_C	Q_B	Q_A	Q_D	Q_C	Q_B	Q_A	
0	0	0	1	1	0	1	0	0	1
1	0	1	0	0	0	1	0	1	2
2	0	1	0	1	0	1	1	0	3
3	0	1	1	0	0	1	1	1	4
4	0	1	1	1	1	0	0	0	5
5	1	0	0	0	1	0	0	1	6
6	1	0	0	1	1	0	1	0	7
7	1	0	1	0	1	0	1	1	8
8	1	0	1	1	1	1	0	0	9
9	1	1	0	0	0	0	1	1	0

Рис. 10.61. Таблица истинности синхронного суммирующего счетчика в коде с избытком 3.

Q_D	Q_C	Q_B	Q_A	
0	0	0	0	x_a
0	0	0	1	x_b
0	0	1	0	x_c
1	1	0	1	x_d
1	1	1	0	x_e
1	1	1	1	x_f

Рис. 10.62. Псевдотетрады.

2. Шаг:

Составление и упрощение уравнений счетчика.

Нормальные формы ИЛИ от $Q_{A(n+1)}$, $Q_{B(n+1)}$, $Q_{C(n+1)}$, $Q_{D(n+1)}$ записываются в карту Карно и упрощаются (рис. 10.63).

Получаются нижеследующие упрощенные и преобразованные уравнения счетчика. Преобразования целесообразны для облегчения дальнейшего сравнения коэффициентов с характеристическими уравнениями триггеров.

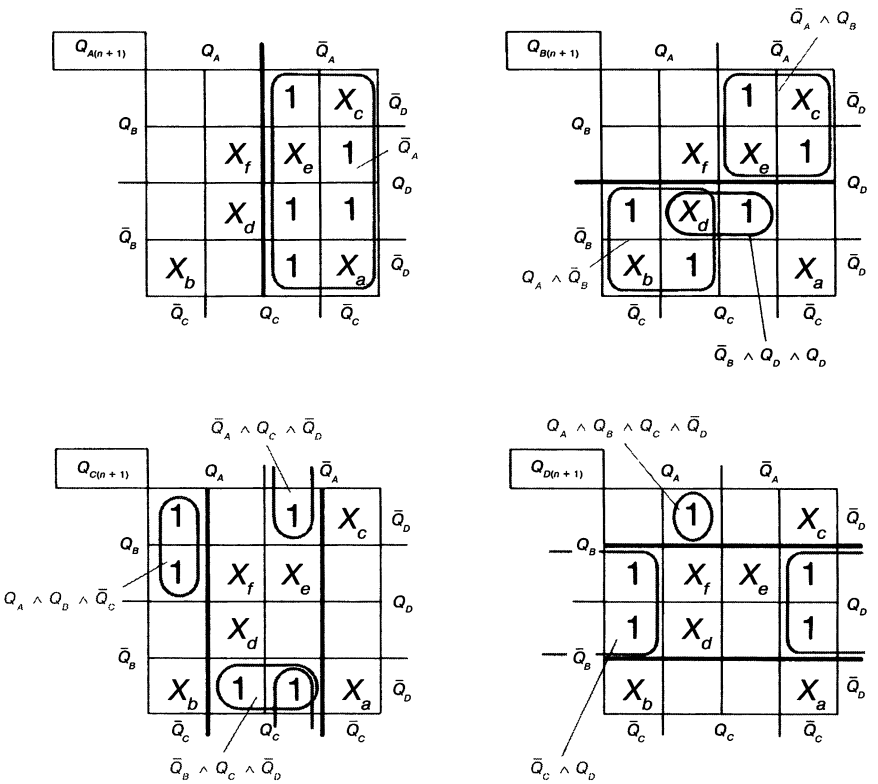


Рис. 10.63. Карта Карно для определения упрощенных уравнений счетчика.

Уравнения счетчика:

$$Q_{A(n+1)} = \bar{Q}_A,$$
$$Q_{A(n+1)} = (1 \wedge \bar{Q}_A) \vee (0 \wedge Q_A); \quad ①$$

$$Q_{B(n+1)} = (\bar{Q}_B \wedge Q_C \wedge Q_D) \vee (Q_A \wedge \bar{Q}_B) \vee (\bar{Q}_A \wedge Q_B),$$
$$Q_{B(n+1)} = [\bar{Q}_B \wedge ((Q_C \wedge Q_D) \vee Q_A)] \vee (\bar{Q}_A \wedge Q_B),$$
$$Q_{B(n+1)} = [[(Q_C \wedge Q_D) \vee Q_A] \wedge \bar{Q}_B] \vee (\bar{Q}_A \wedge Q_B); \quad ②$$

$$Q_{C(n+1)} = (Q_A \wedge Q_B \wedge \bar{Q}_C) \vee (\bar{Q}_A \wedge Q_C \wedge \bar{Q}_D) \vee (\bar{Q}_B \wedge Q_C \wedge \bar{Q}_D),$$
$$Q_{C(n+1)} = (Q_A \wedge Q_B \wedge \bar{Q}_C) \vee [[(\bar{Q}_A \wedge \bar{Q}_D) \vee (\bar{Q}_B \wedge \bar{Q}_D)] \wedge Q_C],$$
$$Q_{C(n+1)} = (Q_A \wedge Q_B \wedge \bar{Q}_C) \vee [[(\bar{Q}_D \wedge (Q_A \wedge \bar{Q}_B)] \wedge Q_C]; \quad ③$$

$$Q_{D(n+1)} = (Q_A \wedge Q_B \wedge Q_C \wedge \bar{Q}_D) \vee (\bar{Q}_C \wedge \bar{Q}_D). \quad ④$$

3. Шаг:

Характеристические уравнения.

Требуется определить характеристические уравнения для применяемых триггеров. Использованы *JK*-триггеры. Характеристическое уравнение для *JK*-триггеров, опуская индекс *n*:

$$Q_{(n+1)} = (J \wedge \bar{Q}) \vee (\bar{K} \wedge Q).$$

Для изготовления счетчика нужны 4 триггера, обозначенные символами *A*, *B*, *C* и *D*. Они имеют следующие характеристические уравнения:

$$Q_{A(n+1)} = (J_A \wedge \bar{Q}_A) \vee (\bar{K}_A \wedge Q_A); \quad ⑤$$

$$Q_{B(n+1)} = (J_B \wedge \bar{Q}_B) \vee (\bar{K}_B \wedge Q_B); \quad ⑥$$

$$Q_{C(n+1)} = (J_C \wedge \bar{Q}_C) \vee (\bar{K}_C \wedge Q_C); \quad ⑦$$

$$Q_{D(n+1)} = (J_D \wedge \bar{Q}_D) \vee (\bar{K}_D \wedge Q_D). \quad ⑧$$

4. Шаг:

Определение уравнений логических связей.

Уравнения счетчика и характеристические уравнения уже определены. Сравнением их коэффициентов определяются уравнения связи.

Триггер A

$$Q_{A(n+1)} = \underset{\Downarrow}{\left(J_A \wedge \bar{Q}_A\right)} \vee \underset{\Downarrow}{\left(\bar{K}_A \wedge Q_A\right)} \qquad ⑤$$

$$Q_{A(n+1)} = \left(1 \wedge \bar{Q}_A\right) \vee \left(0 \wedge Q_A\right) \qquad ①$$

$$J_A = 1, \qquad \boxed{J_A = 1}$$

$$\bar{K}_A = 0, \qquad \boxed{K_A = 1}$$

Триггер B

$$Q_{B(n+1)} = \underset{\Downarrow}{\left(J_B \wedge \bar{Q}_B\right)} \vee \underset{\Downarrow}{\left(\bar{K}_B \wedge Q_B\right)} \qquad ⑥$$

$$Q_{B(n+1)} = \left[\overbrace{\left[\left(Q_C \wedge Q_D\right) \vee Q_A\right]} \wedge \bar{Q}_B\right] \quad \left(\bar{Q}_A \wedge Q_B\right) \qquad ②$$

$$\boxed{J_B = Q_A \vee \left(Q_C \wedge Q_D\right)}$$

$$\bar{K}_B = \bar{Q}_A \qquad \boxed{K_B = Q_A}$$

Триггер C

$$Q_{C(n+1)} = \underset{\Downarrow}{\left(J_C \wedge \bar{Q}_C\right)} \vee \underset{\Downarrow}{\left(\bar{K}_C \wedge Q_C\right)} \qquad ⑦$$

$$Q_{C(n+1)} = \left(\overbrace{\left(Q_A \wedge Q_B\right)} \wedge \bar{Q}_C\right) \vee \left[\left[\overbrace{\left(\bar{Q}_D \wedge \left(\bar{Q}_A \vee \bar{Q}_B\right)\right.}\right] \wedge Q_C\right] \qquad ③$$

$$\boxed{J_C = Q_A \wedge Q_B}$$

$$\bar{K}_C = \bar{Q}_D \wedge \left(\bar{Q}_A \vee \bar{Q}_B\right)$$

$$K_C = \overline{\bar{Q}_D \wedge \left(\bar{Q}_A \vee \bar{Q}_B\right)} = Q_D \vee \overline{\bar{Q}_A \vee \bar{Q}_B}$$

$$K_C = Q_D \vee \left(Q_A \vee Q_B\right)$$

$$\boxed{K_C = \left(Q_A \wedge Q_B\right) \vee Q_D}$$

Триггер D

$$Q_{D(n+1)} = \underset{\Downarrow}{\left(J_D \wedge \bar{Q}_D\right)} \vee \underset{\Downarrow}{\left(\bar{K}_D \wedge Q_D\right)} \quad ⑧$$

$$Q_{D(n+1)} = \left(\overline{Q_A \wedge Q_B \wedge Q_C} \wedge \bar{Q}_D\right) \vee \left(\bar{Q}_C \wedge Q_D\right) \quad ④$$

$$\boxed{J_D = Q_A \wedge Q_B \wedge Q_C}$$

$$\bar{K}_D = \bar{Q}_C \qquad \boxed{K_D = Q_C}$$

Если известны уравнения связи, то можно строить счетчик. Триггеры нужно соединять соответственно уравнениям связи. Готовая схема счетчика представлена на рис. 10.64. Счетчик может начинать счет с любого числа. Если он должен считать с нуля, то необходима дополнительная схема.

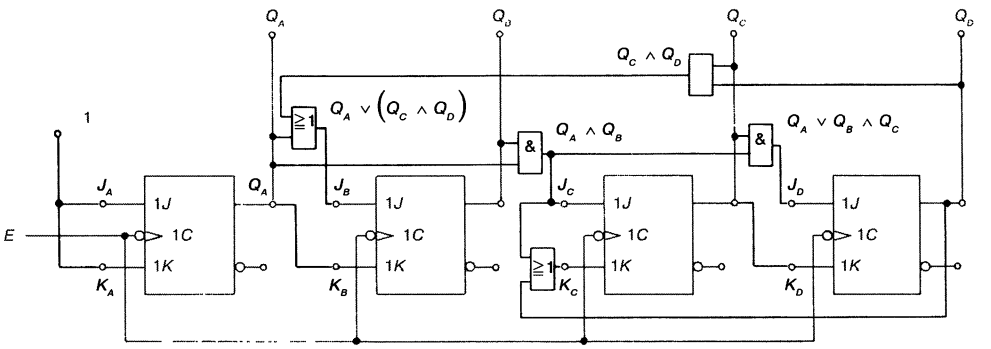


Рис. 10.64. Синхронный суммирующий счетчик в коде с избытком 3.

Иногда существует возможность группировать карту Карно иначе при одинаковом уровне упрощения. В нашем примере карты это имеет место для $Q_{C(n+1)}$. При другой группировке получаются другие уравнения счетчика и другие уравнения логических связей, представляющие равноценное решение задания.

10.4. Делители частоты

Делители частоты — это схемы, понижающие частоту прямоугольных сигналов с определенным коэффициентом пересчета.

Один триггер производит деление частоты в отношении 2 : 1. На двух триггерах можно построить делитель частоты с отношением 4 : 1. Делители частоты бывают двух видов: делители с фиксированным коэффициентом пересчета и делители частоты с регулируемым коэффициентом пересчета. Последние называются также программируемыми делителями частоты.

10.4.1. Асинхронные делители частоты с фиксированным коэффициентом пересчета *K*

В качестве делителей частоты могут использоваться уже известные схемы.

> Любой асинхронный двоичный счетчик может быть использован как делитель частоты с фиксированным коэффициентом пересчета.

Рассмотрим схему и временную диаграмму 3-битового двоичного суммирующего счетчика на рис. 10.65. Первый триггер счетчика делит на два частоту входного сигнала *E*. Второй триггер делит пополам уже разделенную частоту еще раз. Третьим триггером частота делится еще раз на два. 3-битовый двоичный суммирующий счетчик работает как делитель частоты с частотным соотношением 8 : 1.

Двоичные вычитающие счетчики также могут быть применены в качестве делителей частоты (рис. 10.66). Разделенные сигналы имеют лишь иную фазу, чем в двоичных суммирующих счетчиках.

Четные коэффициенты пересчета легко получить. Каждый триггер делит частоту с коэффициентом 2:

$$f_T = \frac{f_E}{2^n},$$

где f_E — входная частота; f_T — разделенная частота; n — количество триггеров.

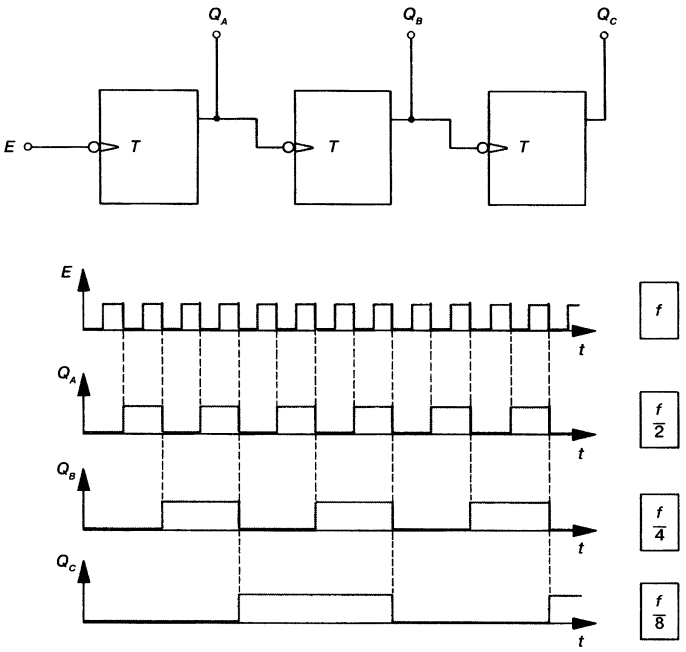


Рис. 10.65. Асинхронный 3-битовый суммирующий счетчик в качестве делителя частоты с коэффициентом пересчета 8 : 1.

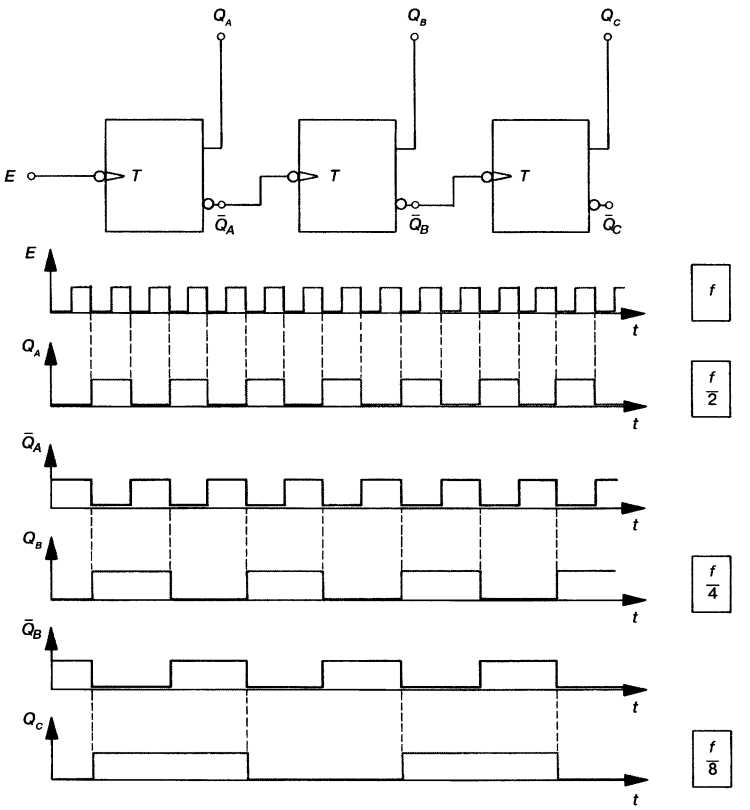


Рис. 10.66. Асинхронный 3-битовый вычитающий счетчик в качестве делителя частоты с коэффициентом пересчета 8 : 1.

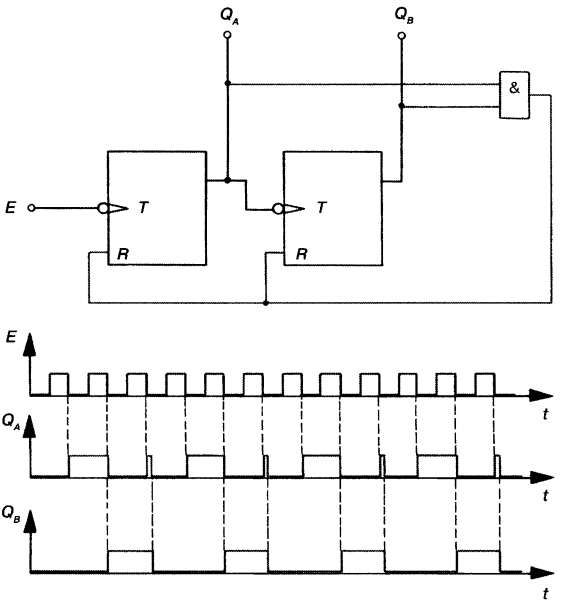


Рис. 10.67. Делитель частоты с коэффициентом пересчета 3 : 1 и временная диаграмма.

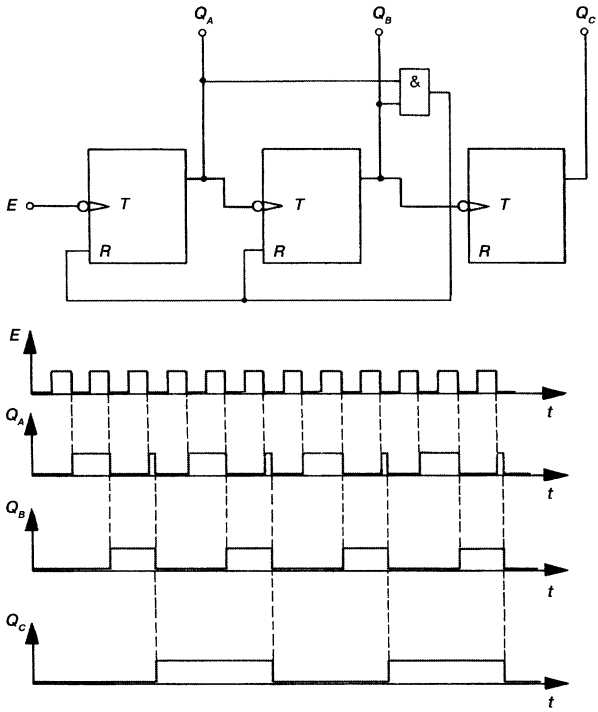


Рис. 10.68. Делитель частоты с коэффициентом пересчета 6 : 1 и временная диаграмма.

Как можно получить нечетные коэффициенты пересчета? Чтобы получить нечетные коэффициенты пересчета, используемые в схеме триггеры должны иметь входы сброса. Делитель частоты с коэффициентом пересчета 3 : 1 представлен на рис. 10.67.

Выходной сигнал Q_B имеет другое импульсно-фазовое соотношение, чем входной сигнал E. Для многих практических задач это неудобно. Если присоединить следующий триггер, то снова получится импульсно-фазовое соотношение 1 : 1 (рис. 10.68).

Задание

Построить схему делителя частоты с коэффициентом пересчета 10 : 1. Импульсно-фазовое соотношение выходного сигнала должно быть 1 : 1.

Сначала следует разработать схему делителя частоты 5 : 1. Затем к этой схеме подключить делитель частоты 2 : 1, то есть еще один триггер (рис. 10.69).

10.4.2. Синхронные делители частоты с фиксированным коэффициентом пересчета *K*

Все вышесказанное для синхронных двоичных счетчиков действует и для асинхронных двоичных счетчиков.

> Любой синхронный двоичный счетчик может также быть использован как делитель частоты с фиксированным коэффициентом пересчета.

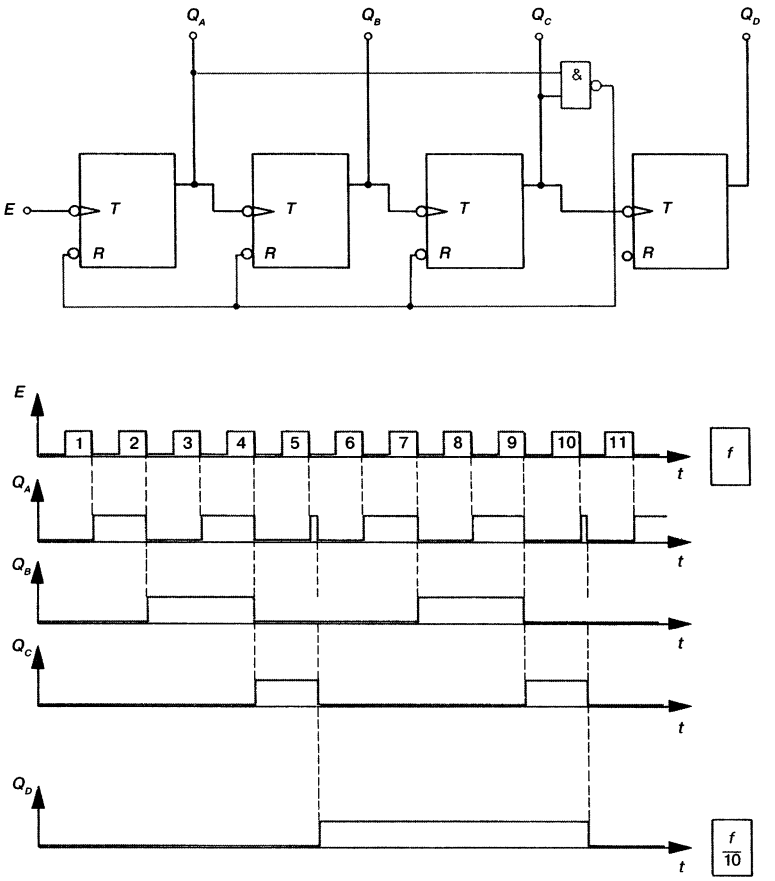


Рис. 10.69. Делитель частоты с коэффициентом пересчета 10 : 1 и временная диаграмма.

Это правило работает только для четных коэффициентов пересчета, т. е. 2 : 1, 4 : 1, 8 : 1, 16 : 1 и т. д. Для других коэффициентов пересчета, особенно для нечетных, нужно изменить схему включения входов триггеров.

На рис. 10.70 изображена схема и временная диаграмма синхронного делителя частоты с коэффициентом пересчета 3 : 1.

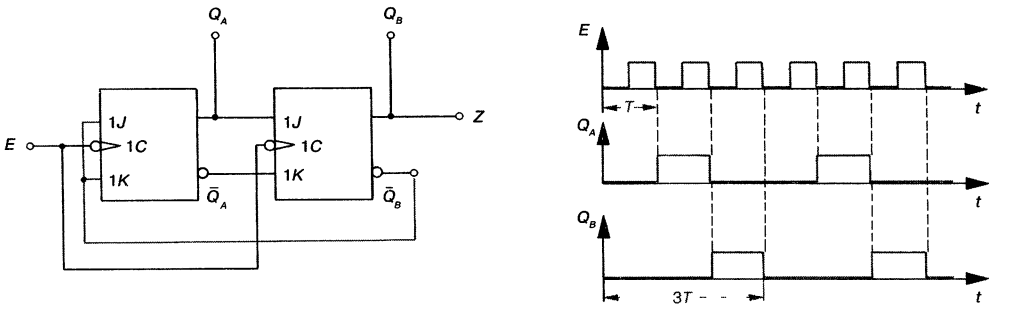
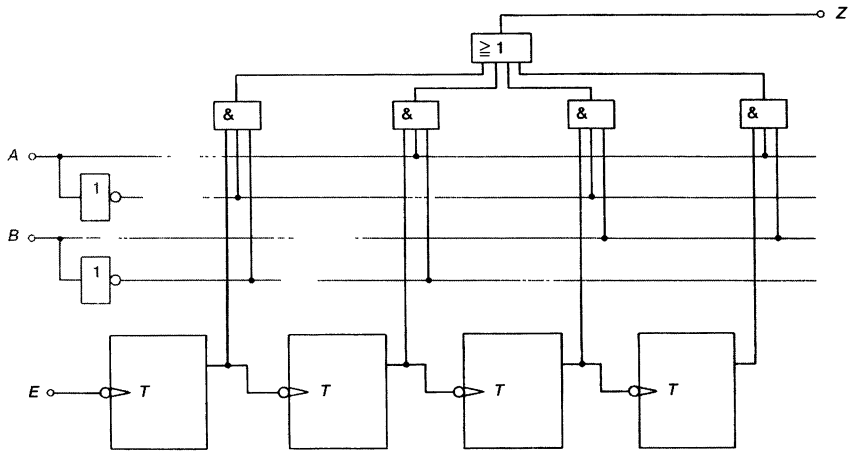


Рис. 10.70. Синхронный делитель частоты с коэффициентом пересчета 3 : 1 и временная диаграмма.

10.4.3. Делитель частоты с регулируемым коэффициентом пересчета

Делители частоты с регулируемым коэффициентом пересчета являются в принципе переключаемыми делителями частоты. Они проводят несколько делений частоты. Сигнал с желаемой частотой подается через схему выборки на выход. Делитель частоты на рис. 10.71 делит частоту входного сигнала в соотношении 2 : 1, 4 : 1, 8 : 1 и 16 : 1. Сигналами на *A* и *B* выбирается нужный сигнал на выходе *Z*.



B	A	Коэффициент пересчета
0	0	2 : 1
0	1	4 : 1
1	0	8 : 1
1	1	16 : 1

Рис. 10.71. Делитель частоты с регулируемым коэффициентом пересчета.

Схема делителя частоты может изменяться посредством переключений, так что можно также получать нечетные коэффициенты пересчета.

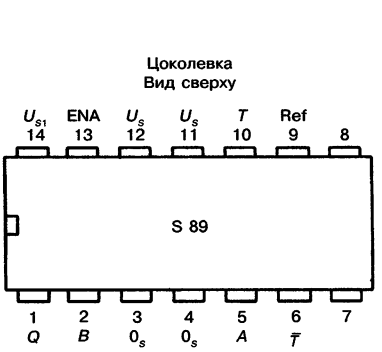
Важной характеристикой делителя частоты является максимальная частота, которую он может разделить. В ЭСЛ-технике (см. семейство ЭСЛ в разд. 6.7) можно строить делители частоты для частот до 900 МГЦ. Регулируемый делитель частоты для частоты до 500 МГЦ выпускается в виде микросхемы с названием S 89. Схема этого счетчика с цоколевкой приведена на рис. 10.72. Возможные коэффициенты пересчета 50 : 1, 51 : 1, 100 : 1, 101 : 1, 102 : 1, 200 : 1 и 202 : 1.

Нужный коэффициент пересчета устанавливается входами управления *A*, *B* и *ENA*.

Таблица команд и основные параметры микросхемы приведены на рис. 10.73.

Контрольный тест

1. Чем различаются синхронные и асинхронные счетчики?
2. Изобразите схему асинхронного 8-битового суммирующего счетчика на базе *JK*-триггеров, управляемых прямым фронтом.



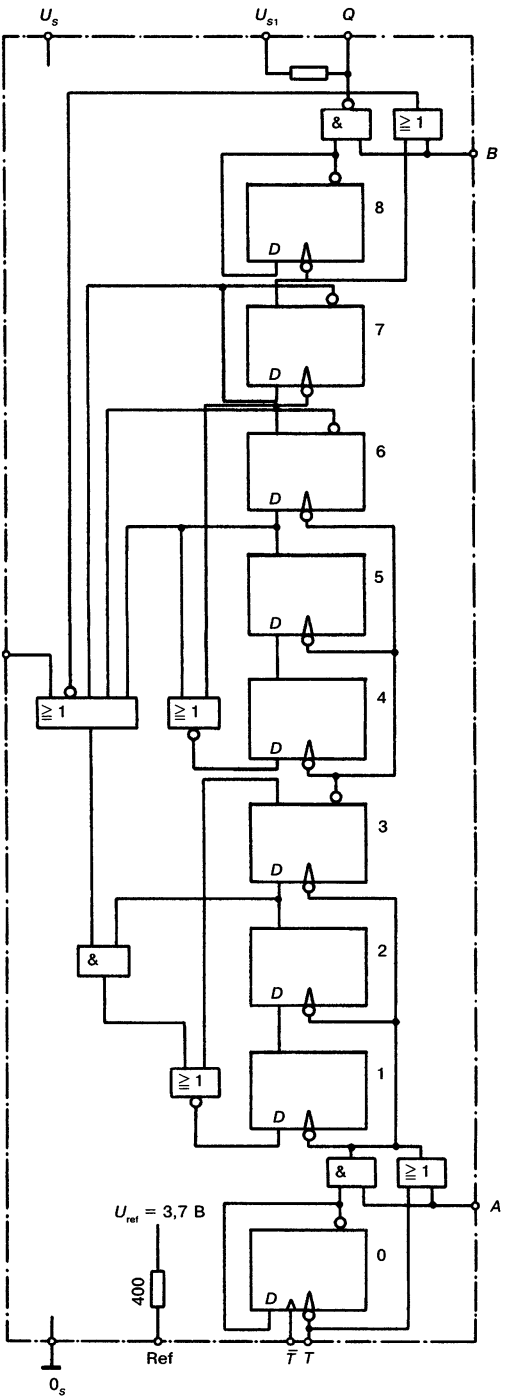


Рис. 10.72. Схема и цоколевка программируемого счетчика S 89 (Siemens).

		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S		4,75		5,25	В
Потребляемый ток	I_S	Входы, выходы открыты		55	85	мА
Входное напряжение *L*-уровня на ENA	$U_{ENA\ L}$				1	В
Входное напряжение *H*-уровня на ENA	$U_{ENA\ H}$	T_U = −30 °C	3,2			В
Входное напряжение *H*-уровня на ENA	$U_{ENA\ H}$	T_U = 25 °C	3,0			В
Входное напряжение *H*-уровня на ENA	$U_{ENA\ H}$	T_U = 80 °C	2,8			В
Потребляемый ток *H*-уровня на ENA	$I_{ENA\ H}$	$U_{ENA} = U_{ENA\ H} = f(T_U)$		0,17	0,3	мА
Потребляемый ток *H*-уровня на ENA	$I_{ENA\ H}$	U_{ENA} = 9 В		1,7	3	мА
Входное напряжение *L*-уровня на *A* или *B*	$U_{AB\ L}$				1,5	В
Входное напряжение *H*-уровня на *A* или *B*	$U_{AB\ H}$		$U_S - 0{,}1$		$U_S + 0{,}1$	В
Потребляемый ток *H*-уровня на *A* или *B*	$I_{AB\ H}$	$U_{AB} = U_S$		0,5	1	мА
Пороговое напряжение срабатывания на *T*	U_T	U_S = 5 В		3,7		В
Увеличение напряжения переключения на *T* статическое (*T* и Ref соединены)	$U_{T\ SS}$		250		1600	мВ
Увеличение напряжения переключения на *T* при 500 МГц (*T* и Ref соединены)	$U_{T\ SS}$	U_S = 5 В	250		400	мВ
Выходное напряжение на Q_1	U_{Q1}	I_{Q1} = 3,2 мА			0,4	В
R между Q_1 и Q_2	R_{Q2}		1,8	2,5	3,2	Ом
Параметры						
Напряжение питания	U_S		4,5		5,5	В
Максимальная входная частота	$f_{T\ max}$	при 50/51, 100/101	250[1)]			МГц
Максимальная входная частота	$f_{T\ max}$	при 100/102, 200/202	500[1)]			МГц
Минимальная входная частота синусоидальная	$f_{T\ min}$	при 50/51, 100/101	20[1)]			МГц
Минимальная входная частота синусоидальная	$f_{T\ min}$	при 100/102, 200/202	40[1)]			МГц

[1)] Амплитуда (*SS*) на *T*: 250 мВ ≤ $U_{T\ SS}$ ≤ 400 мВ; U_S: 4,75 ≤ U_S ≤ 5,5 В.

Рис. 10.73. Таблица команд с параметрами программируемого счетчика S 89.

Таблица команд

A	B	ENA	f_T/f_Q
H	H	H	200
H	H	L	202
H	L	H	100
H	L	L	102
L	H	H	100
L	H	L	101
L	L	H	50
L	L	L	51

f_T – входная частота
f_Q – выходная частота

3. Как можно из асинхронного 4-битового суммирующего счетчика на базе *T*-триггеров сделать *BCD*-суммирующий счетчик? *T*-триггеры должны переключаться с обратным фронтом синхроимпульса и иметь нетактируемый вход сброса, который сбрасывает триггер 0-сигналом.

4. Что понимают под счетчиком по модулю *n*?

5. Изобразите схему счетчика по модулю 19 (суммирующего) на базе триггеров из вопроса 3.

6. Измените схему счетчика по модулю 19 из вопроса 5 таким образом, чтобы он начинал счет с 18 и считал на уменьшение до 0.

7. Как работает схема, представленная на рис. 10.74?

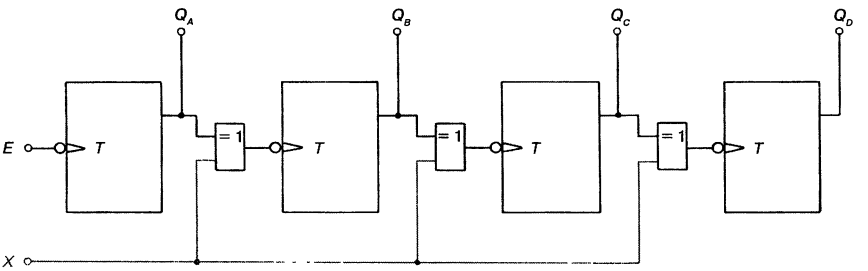


Рис. 10.74.

8. Чем отличаются счетчики с предварительной установкой от программируемых счетчиков?

9. Измените схему 4-битового суммирующего счетчика на рис. 10.7 таким образом, чтобы получился программируемый счетчик.

10. Изобразите схему синхронного 5-битового суммирующего счетчика на базе *JK*-триггеров, управляемых обратным фронтом.

11. Опишите методику расчета синхронных счетчиков.

12. Как можно из синхронного 4-битового суммирующего счетчика с минимальными изменениями в схеме сделать синхронный 4-битовый вычитающий счетчик?

13. 4-битовый синхронный суммирующий счетчик должен работать как делитель частоты с коэффициентом пересчета 8 : 1. Как изменится схема?

14. Требуется нарисовать схему делителя частоты с коэффициентом пересчета 14 : 1 на базе триггеров из вопроса 3. Импульсно-фазовое соотношение выходного сигнала должно быть 1 : 1.

ГЛАВА 11

ЦИФРОВЫЕ СХЕМЫ ВЫБОРКИ И СВЯЗИ

11.1. Цифровой коммутатор, мультиплексор и демультиплексор

Цифровые коммутаторы выбирают из множества возможных входных данных нужные и передают их на выходы.

Входные сигналы поступают с разделением по времени. Схема, которая последовательно передает определенные входные сигналы на выходы, называется **мультиплексор** (рис. 11.1).

Мультиплексор — это цифровой коммутатор с разделением по времени.

Аналогично можно передавать входные сигналы с разделением по времени на один из выходов.

Схема, передающая входные сигналы с разделением по времени на один из выходов, называется **демультиплексор**.

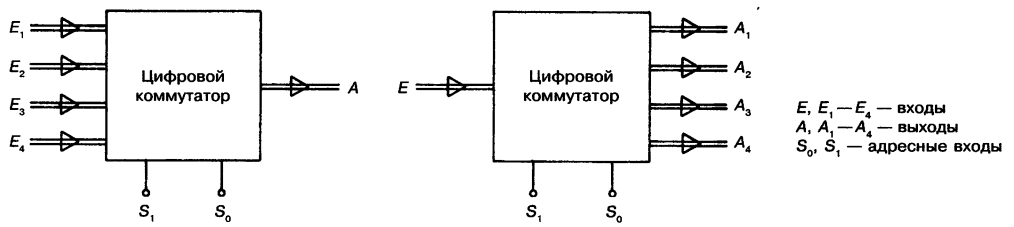


Рис. 11.1. Цифровой коммутатор-мультиплексор, демультиплексор.

11.1.1. Мультиплексор «4 в 1»

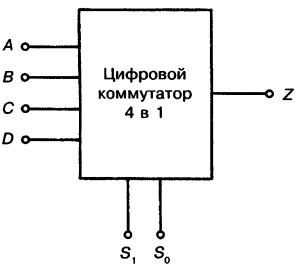


Рис. 11.1а. Мультиплексор 4 в 1.

Рассмотрим принцип действия мультиплексора на простой схеме. Мультиплексор «4 в 1» имеет четыре входа. Каждый из эти четырех входов должен иметь возможность подключения к выходу (рис. 11.1а).

Мультиплексор работает как переключатель с 4 положениями. В положении 1 *A* соединен с *Z*. В положении 2 *B* соединен с *Z* и т. д.

Переключение происходит посредством управляющих входов. Для управления 4 различными положениями требуются 2 управляющих входа. С 2 бита-

Положение	S_1	S_0	$Z =$
1	0	0	A
2	0	1	B
3	1	0	C
4	1	1	D

Рис. 11.2. Таблица истинности мультиплексора «4 в 1».

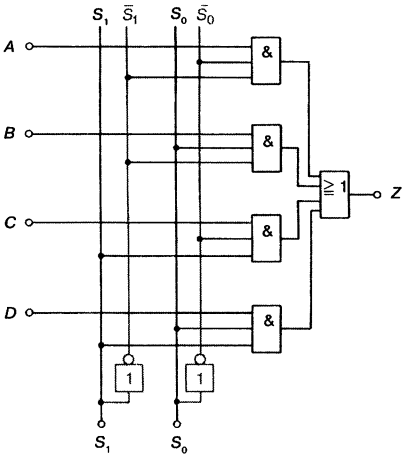


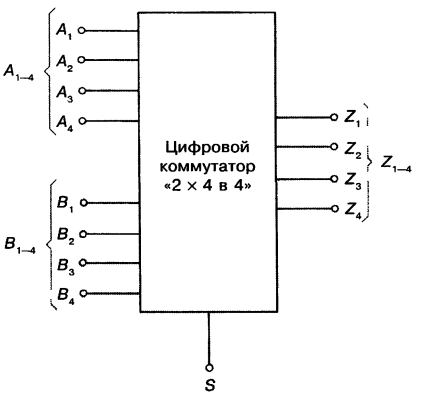
Рис. 11.3. Схемы мультиплексоров «4 в 1».

ми можно образовать четыре комбинации, которые устанавливают четыре положения (рис. 11.2).

Схему мультиплексора «4 в 1» легко разработать. Переменные S_1 и S_2 должны присутствовать в прямой и инверсной форме. Входы выбираются посредством И-элементов при условии подачи на управляющие входы соответствующей комбинации (рис. 11.3).

11.1.2. Цифровой селектор «2 × 4 в 4»

Цифровой селектор «2 × 4 в 4» имеет два входа по 4 бита и 4-битовый выход (рис. 11.4). На выход Z подаются либо четыре A-входа, либо четыре B-входа. Так как возможны только два положения, в наличии имеется только один адресный вход S (рис. 11.4). Схема этого цифрового коммутатора представлена на рис. 11.5.



Положение	S	$Z_{1-4} =$
1	0	A_{1-4}
2	1	B_{1-4}

Рис. 11.4. Селектор «2 × 4 в 4».

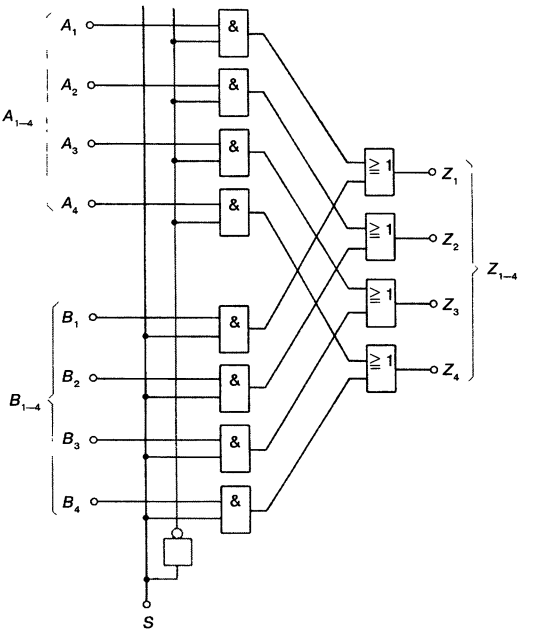


Рис. 11.5. Принципиальная схема селектора «2 × 4 в 4».

11.1.3. Цифровой селектор «4 × 8 в 8»

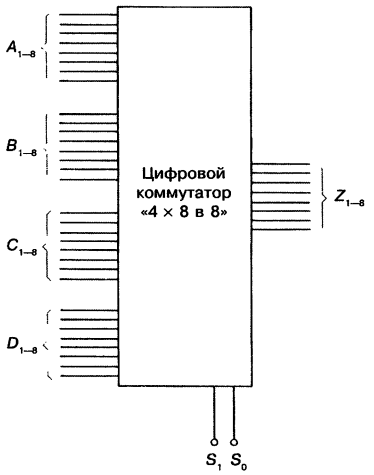


Рис. 11.6. Цифровой селектор «4 × 8 в 8».

Этот селектор имеет большое значение в микропроцессорной технике (рис. 11.6). 8-битовые слова выборочно передаются на 8-битовые выходы. Возможны четыре положения коммутатора. Передача сигнала управления производится с помощью двух адресных входов S_0 и S_1 (2-битовые команды).

Команда $S_0 = 0$, $S_1 = 0$ переключает восемь A-входов на восемь Z-выходов ($Z_1 = A_1$, $Z_2 = A_2$, $Z_3 = A_3$, $Z_4 = A_4$ и т. д.). Если на выход должны быть переданы B-входы, должна быть команда $S_0 = 1$, $S_1 = 0$. Для коммутации C- и D-входов соответственно $S_0 = 0$, $S_1 = 1$ и $S_0=1$, $S_1 = 1$.

11.1.4. Цифровой селектор-мультиплексор 16 в 1

Цифровой селектор-мультиплексор «16 в 1» имеет 16 входов, каждый из которых может быть передан на выход. То есть в наличии имеются 16 возможных положений. Так как каждое положение выбирается отдельной командой, необходимы 16 команд. Для формирования 16 команд необходимы 4 бита. Схема должна иметь 4 управляющих входа (рис. 11.7).

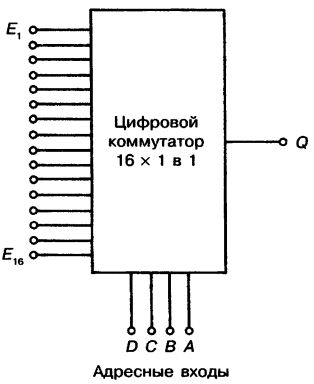


Рис. 11.7. Мультиплексор «16 в 1».

Цифровой селектор-мультиплексор «16 в 1» выпускается в виде микросхемы с обозначением FLY 111-74150. Схема и цоколевка показаны на рис. 11.8. Поступающие на входы сигналы инвертируются на выходе.

На рис. 11.9 представлена таблица данных мультиплексора FLY 111-74150.

Цифровой селектор «16 в 1» работает как мультиплексор, если 16 возможные 4-битовые управляющие команды по очереди поступают в управляющие входы. Каждая команда длится, например, в течение миллисекунды, затем следует следующая команда. Счет начинается, как правило, с 0000 и продолжается до 1111. Затем цикл начинается снова.

11.1.5. Демультиплексор «1 в 4»

Демультиплексор работает наоборот по сравнению с мультиплексором. Поданный на вход сигнал будет выборочно передан на один из выходов. Управление производится управляющими битами.

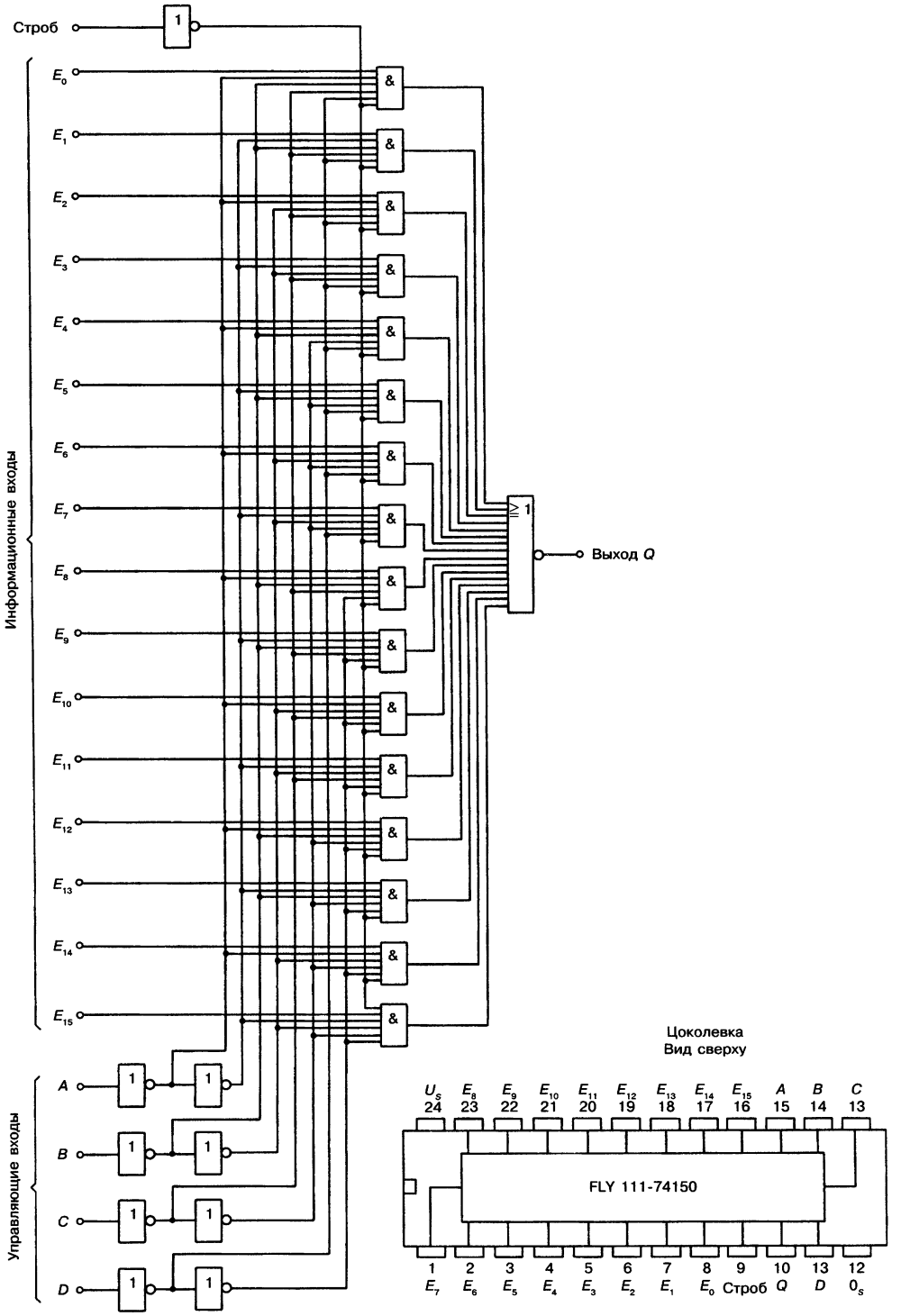


Рис. 11.8. *Принципиальная схема и цоколевка мультиплексора 16 в 1* FLY 111-74150 (Siemens).

16-битовый мультиплексор

FLY 111-74150 FLY 115-84150

Микросхема FLY 111/115 имеет 16 входов от E_0 до E_{15}, на них одновременно подаются информационные сигналы *H*- или *L*-уровней, входы выбираются в зависимости от бинарной комбинации состояний управляющих входов *A*, *B*, *C*, *D* и последовательно в инверсной форме передаются на выход *Q*. *H*-сигналом на входе *Strobe* выход *Q* блокируется (*H*-сигнал) независимо от состояния входов.

Эти микросхемы находят применение при последовательной передаче данных по линии в комбинации с микросхемой FLY 141/145.

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
Напряжение питания	U_S	U_S = 4,75 В	4,75	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}		2,0			В
Входное напряжение *L*-уровня	U_{IL}				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 12 мА			1,5	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В, $-I_{QH}$ = 400 мкА	2,4	3,4		В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В, I_{QL} = 16 мА		0,2	0,4	В
Входной ток на канал	I_I	U_S = 5,25 В, U_I = 5,5 В			1	мА
H-входной ток на канал	I_{IH}	U_S = 5,25 В, U_{IH} = 2,4 В			40	мкА
L-входной ток на канал	$-I_{IL}$	U_S = 5,25 В, U_{IL} = 0,4 В			1,6	мА
Ток короткого замыкания	$-I_Q$	U_S = 5,25 В, U_{QL} = 0 В	18		55	мА
Потребляемый ток	I_S	U_S = 5,25 В, U_{IL} = 4,5 В		40	68	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						
Время прохождения сигнала (быстродействие)						
От *A*, *B*, *C*, *D* к *Q*	t_{PHL}	R_L = 400 Ом C_L = 15 пФ		22	33	нс
	t_{PLH}			23	35	нс
От *Strobe* к *Q*	t_{PHL}			21	30	нс
От E_0 до E_{15} к *Q*	t_{PLH}			15,5	24	нс
	t_{PHL}			8,5	14	нс
	t_{PLH}			13	20	нс
Логические параметры						
H-коэффициент разветвления по выходу на канал	F_{QH}				20	
L-коэффициент разветвления по выходу на канал	F_{QL}				10	
Входной нагрузочный коэффициент на канал	F_I				1	

Рис. 11.9. Таблица данных цифрового селектора FLY 111-74150.

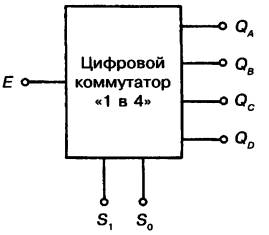


Рис. 11.10. Демультиплексор «1 в 4».

Положение	S_1	S_0	E =
1	0	0	Q_A
2	0	1	Q_B
3	1	0	Q_C
4	1	1	Q_D

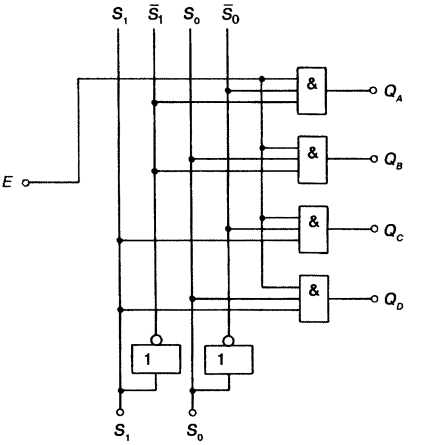


Рис. 11.11. Схема демультиплексора «1 в 4».

Демультиплексор «1 в 4» имеет один вход и четырех выхода (рис. 11.10). Необходимы четыре положения и, следовательно, четыре возможные комбинации. Для четырех комбинаций требуются два управляющих входа (S_0 и S_1).

Схема демультиплексора «1 в 4» представлена на рис. 11.11. Сформированный управляющими сигналами входной сигнал поступает на И-элемент.

11.2. Дешифраторы

Для управления различными модулями требуются так называемые **адреса**. В цифровой технике под адресом понимают последовательность 1-0 определенной длины, т. е. бинарное слово с установленным количеством битов. Существуют, например, 2-битовые адреса, 4-битовые адреса и т. д.

> **Дешифратор** является схемой с определенным количеством выходов. На выходы, выбранные через адресные входы, подается 1-сигнал.

11.2.1. 2-битовые дешифраторы

Если у дешифратора четыре выхода, то требуются два адресных входа. Такой дешифратор управляется 2-битовым адресом. С 2 битами можно получить различные адреса (рис. 11.12). Схема 2-битового дешифратора показана на рис. 11.13.

11.2.2. 4-битовые дешифраторы

3-битовым адресом могут быть выбраны 8 выходов. 4-битовые адреса позволяют выбрать 16 выходов. Можно записать зависимость:

$$n = 2^k$$

где n — число выходов; k — число адресных входов.

На рис. 11.14 изображен 4-битовый дешифратор с таблицей истинности. Возможны 16 различных адресов, и могут быть выбраны 16 различных выходов.

Адрес (номер)	A_2	A_1	Q_A	Q_B	Q_C	Q_D
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	1	0	0	0	1	0
4	1	1	0	0	0	1

Рис. 11.12. 2-битовый дешифратор и его таблица истинности.

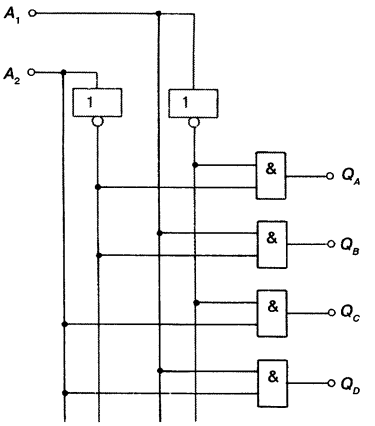


Рис. 11.13. Схема 2-битового дешифратора.

A_4	A_3	A_2	A_1	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}	Q_{14}	Q_{15}
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0					1											
0	1	0	1						1										
0	1	1	0							1									
0	1	1	1								1								
1	0	0	0									1							
1	0	0	1										1						
1	0	1	0											1					
1	0	1	1												1				
1	1	0	0													1			
1	1	0	1														1		
1	1	1	0															1	
1	1	1	1																1

Нули не вписаны для лучшего обзора таблицы

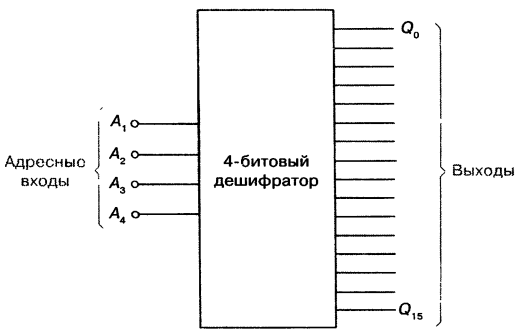


Рис. 11.14. 4-битовый дешифратор и его таблица истинности.

11.3. Цифровые компараторы

Цифровым компаратором называется схема, которая сравнивает друг с другом два бинарных выражения A и B и сообщает, является ли $A > B$, $A = B$ или $A < B$.

Равенство двух бинарных выражений легко выявить. Все биты обоих выражений должны совпадать, иначе они не равны.

Определить, что A больше, чем B или наоборот, сложнее. Здесь нужно знать применяемый код. Сравнение возможно только при условии, что компаратор спроектирован для работы с кодом, в котором закодированы выражения A и B.

Обычно компараторы спроектированы для двоичной системы счисления, в частности для работы в двоично-десятичном коде.

11.3.1. 1-битовый компаратор

Простейшим компаратором является 1-битовый компаратор. Он сравнивает два бинарных слова длиною в 1 бит. Схема имеет три выхода (рис. 11.15). На выходе X появляется 1, если $A > B$. Если $A = B$, то $Y = 1$. Если $A < B$, то $Z = 1$.

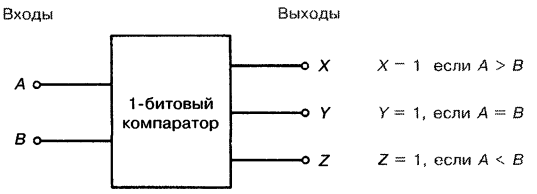
Схема компаратора, приведенная на рис. 11.15, может быть синтезирована на базе схемы неравнозначности. Однако ее можно рассчитать также с помощью нормальной формы ИЛИ (рис. 11.16).

$$X = A \wedge \bar{B};$$

$$Z = \bar{A} \wedge B;$$

$$Y = (\bar{A} \wedge \bar{B}) \vee (A \wedge B);$$

$$Y = \overline{(\bar{A} \wedge B) \vee (A \wedge \bar{B})}.$$



Вар.	B	A	X	Y	Z
1	0	0	0	1	0
2	0	1	1	0	0
3	1	0	0	0	1
4	1	1	0	1	0

$\Rightarrow X = A \wedge \bar{B}$

$\Rightarrow Z = \bar{A} \wedge B$

$Y = (\bar{A} \wedge \bar{B}) \vee (A \wedge B)$

Рис. 11.15. 1-битовый компаратор и его таблица истинности.

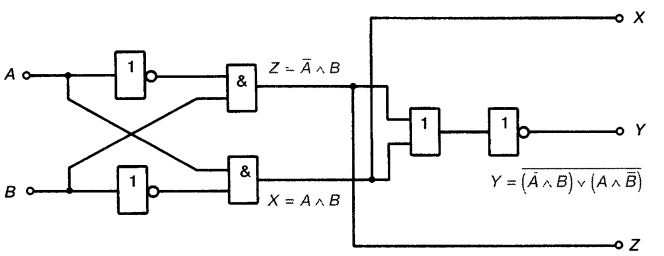


Рис. 11.16. Схема 1-битового компаратора.

11.3.2. 3-битовый компаратор для *BCD*-кода (двоично-десятичного кода)

3-битовый компаратор сравнивает друг с другом два 3-битовых слова. На рис. 11.17 изображен 3-битовый компаратор для двоично-десятичного кода и его таблица истинности. Таблица истинности сокращена. У полной таблицы истинности для шести переменных возможны 64 комбинации.

Вар.	2^2 A_3, B_3	2^1 A_2, B_2	2^0 A_1, B_1	$A > B$ X	$A = B$ Y	$A < B$ Z
1	$A_3 > B_3$	X	X	1	0	0
2	$A_3 < B_3$	X	X	0	0	1
3	$A_3 = B_3$	$A_2 > B_2$	X	1	0	0
4	$A_3 = B_3$	$A_2 < B_2$	X	0	0	1
5	$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	1	0	0
6	$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	0	0	1
7	$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	0	1	0

Рис. 11.17. 3-битовый компаратор для *BCD*-кода и его таблица истинности.

Сначала должны сравниваться наиболее значимые биты, т. е. A_3 с B_3. Если $A_3 > B_3$, то $A > B$. Если $A_3 < B_3$, то $A < B$. Если $A_3 = B_3$, то переходим на следующий по значимости разряд. Если $A_2 > B_2$, то $A > B$. Если $A_2 < B_2$, то $A < B$.

Если $A_2 = B_2$, то переходим на следующий по значимости разряд. Если $A_1 > B_1$, то $A > B$. Если $A_1 < B$, то $A < B$. Если все три бита равны, то $A = B$.

При проектировании схемы 3-битового компаратора воспользуемся схемой 1-битового компаратора (рис. 11.15 и рис. 11.16 соответственно). В схеме должен присутствовать вход запрета (рис. 11.18).

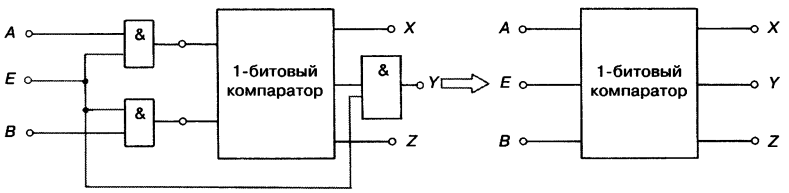


Рис. 11.18. 1-битовый компаратор со схемой запрета.

Три 1-битовых компаратора со схемой запрета должны быть включены таким образом, чтобы соответствовать таблице истинности (рис. 11.17). Общая схема показана на рис. 11.17.

Случай 1:

$$A_3 > B_3 \; X_3 = 1 \Rightarrow X = 1 \Rightarrow A > B.$$

Так как $Y_3 = 0$, то входы 1-битовых компараторов I и II будут заперты.

Случай 2:

$$A_3 < B_3 \; Z_3 = 1 \Rightarrow Z = 1 \Rightarrow A < B.$$

Так как $Y_3 = 0$, то входы 1-битовых компараторов I и II будут заперты.

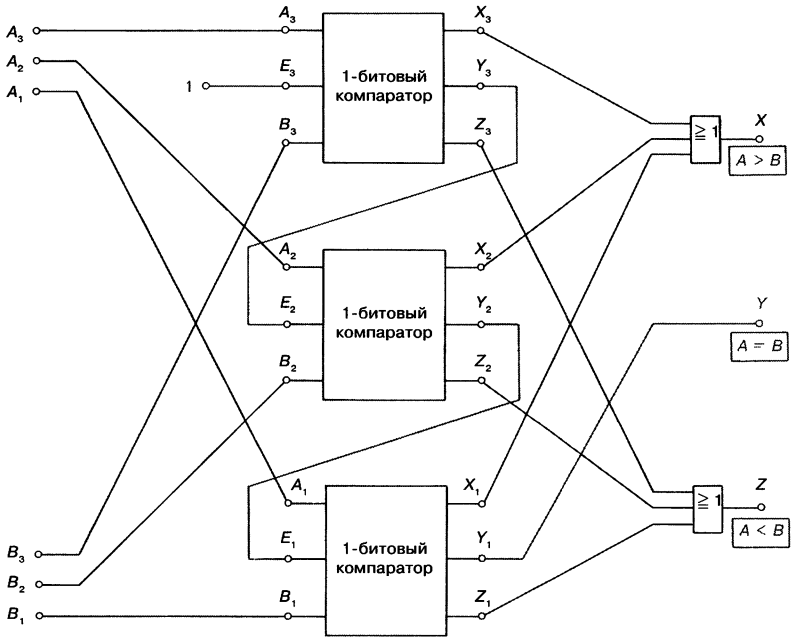


Рис. 11.19. Схема 3-битового компаратора для *BCD*-кода.

Случай 3:

$$A_3 = B_3 \qquad A_2 > B_2.$$

Теперь $Y_3 = 1$. Вход 1-битового компаратора II открывается. Так как $A_2 > B_2$, то $Y_2 = 0$. Входы 1-битового компаратора I остаются заблокированными. X_2 будет равен 1 и, следовательно, $X = 1$.

Случай 4:

$$A_3 = B_3 \qquad A_2 < B_2.$$

То же, что и в случае 3, только $Z_2 = 1$ и, следовательно, $Z = 1$.

Случай 5:

$$A_3 = B_3 \qquad A_2 = B_2.$$

Выход $Y_2 = 1$. Поэтому открывается вход 1-битового компаратора I. Так как $A_1 > B_1$, то $X_1 = 1$ и, следовательно, $X = 1$.

Случай 6:

$$A_3 = B_3 \qquad A_2 = B_2.$$

То же, что и в случае 5. Так как $A_1 < B_1$, то $Z_1 = 1$ и, следовательно, $Z = 1$.

Случай 7:

$$A_3 = B_3 \qquad A_2 = B_2 \qquad A_1 = B_1.$$

$Y_1 = 1$ и, следовательно, $Y = 1$.

11.3.3. 4-битовый компаратор для двоичного кода

Структура 4-битового компаратора похожа на структуру 3-битового компаратора, только нужно добавить еще один 1-битовый компаратор с запирающимися входами.

4-битовые компараторы для двоичного кода выпускаются в виде интегральных микросхем. На рис. 11.20 представлена цоколевка и таблица истинности схемы FLH 431-7485. Схема принадлежит к ТТЛ-семейству. Полная таблица данных приведена на рис. 11.21.

FLH 431-7485
FLH 435-8485

Логические соотношения

Информационные входы				Входы переноса			Выходы		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 > B_1$	$A_0 = B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 < B_1$	$A_0 = B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 > B_1$	$A_0 < B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 < B_1$	$A_0 < B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 > B_2$	$A_1 < B_1$	$A_0 = B_0$	L	L	H	L	L	H
$A_3 = B_3$	$A_2 < B_2$	$A_1 > B_1$	$A_0 < B_0$	X	X	H	L	L	H
$A_3 = B_3$	$A_2 < B_2$	$A_1 < B_1$	$A_0 < B_0$	H	H	L	L	L	L
$A_3 = B_3$	$A_2 < B_2$	$A_1 < B_1$	$A_0 < B_0$	L	L	L	H	H	L

$X \triangleq L$ (X соответствует L- или H-сигналу)

Цоколевка

Вид сверху

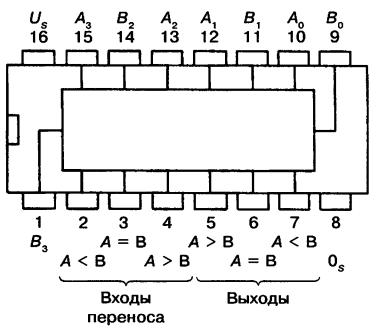


Рис. 11.20. Цоколевка и сокращенная таблица истинности схемы FLH 431-7485 (Siemens).

4-битовый компаратор

FLH 431-7485

FLH 435-8485

Модуль FLH 431/435 сравнивает два двоичных 4-битовых слова (слово A и слово B) и различает три состояния: $A > B$, $A = B$, $A < B$.

Этот модуль может расширяться без дополнительной логики тремя входами переноса для сравнения двух слов любой длины. При этом для каждого следующего 4-битового слова увеличивается время задержки на время задержки двух вентилей. Например, при сравнении двух 8-битовых слов задержка достигает 38 нс. Типичная задержка прохождения для 4-битовых слов — 24 нс.

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел B	Тип	Верхний предел A	Единица измерения
Напряжение питания	U_S	U_S = 4,75 В	4,75	5,0	5,25	В
Входное напряжение H-уровня	U_{IH}		2,0			В
Входное напряжение L-уровня	U_{IL}				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_1$ = 12 мА			1,5	В
Выходное напряжение H-уровня	U_{QH}	U_S = 4,75 В, $-I_{QH}$ = 400 мкА U_{IH} = 2 В, U_{IL} = 0,8 В	2,4	3,4		В
Выходное напряжение L-уровня	U_{QL}	U_S = 4,75 В, I_{QL} = 16 мА U_{IH} = 2 В, U_{IL} = 0,8 В		0,2	0,4	В

Статические параметры в температурных зонах 1 и 5		Условия испытаний	Нижний предел *B*	Тип	Верхний предел *A*	Единица измерения
H-входной ток на канал	I_{IH}	U_{IH} = 2,4 В, U_I = 5,5 В, U_S = 5,25 В			120	мкА
кроме *A* < *B* и *A* > *B*	I_I				1,0	мА
H-входной ток на канал	I_{IH}				40	мкА
кроме *A* < *B* и *A* > *B*	I_I				1,0	мА
L-входной ток на канал, кроме *A* < *B* и *A* > *B*	$-I_{IL}$	U_S = 5,25 В, U_{IL} = 0,4 В			4,8	мА
L-входной ток на канал, кроме *A* < *B* и *A* > *B*	$-I_{IL}$	U_{IL} = 0,4 В			1,6	мА
Ток короткого замыкания	$-I_Q$	U_S = 5,25 В	18		55	мА
Потребляемый ток	I_S	U_S = 5,25 В		56	88	мА
Время переключения, при U_S = 5 В, T_U = 25 °C						
Время прохождения сигнала (быстродействие)						
Вход от *A* к *B*	t_{PLH}	R_L = 400 Ом, C_L = 15 пФ		17	26	нс
Выход *A* < *B* или *A* > *B*	t_{PHL}			20	30	нс
Вход от *A* к *B*	t_{PLH}			23	35	нс
Выход *A* = *B*	t_{PHL}			20	30	нс
Вход от *A* к *B* или *A* = *B*	t_{PLH}			7	11	нс
Выход *A* > *B*	t_{PHL}			11	17	нс
Вход *A* = *B*	t_{PLH}			13	20	нс
Выход *A* = *B*	t_{PHL}			11	17	нс
Вход от *A* > *B* или *A* = *B*	t_{PLH}			7	11	нс
Выход *A* > *B*	t_{PHL}			11	17	нс
Коэффициент разветвления по выходу на канал	F_Q				10	
Входной нагрузочный коэффициент на канал на канал *A* < *B* или *A* > *B*	F_I				1	
Входной нагрузочный коэффициент на любом другом канале	F_I				3	

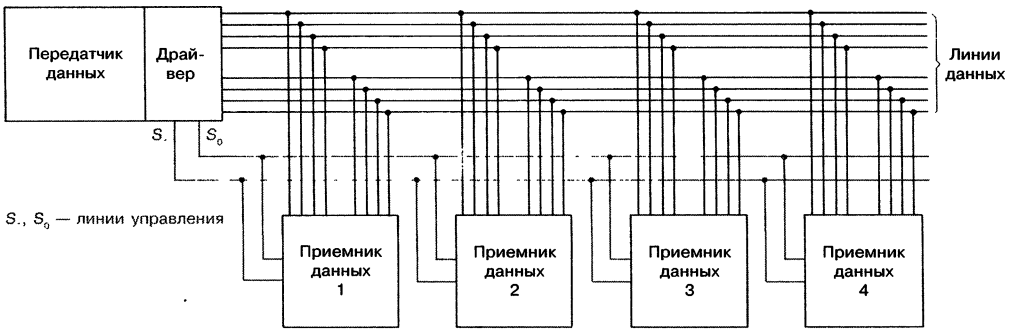
Рис. 11.21. Таблица данных схемы FLH 431-7485 (Siemens).

11.4. Шины данных

11.4.1. Структура и принцип действия

Под **шиной** понимают систему для распределения и передачи бинарной информации.

Слово шина происходит от латинского слова omnibus (для всех). Все элементы системы, передающие или принимающие бинарную информацию, связаны друг с другом шиной.



Вар.	S_1	S_0	Номер приемника данных
1	0	0	1
2	0	1	2
3	1	0	3
4	1	1	4

Рис. 11.22. Последовательная однонаправленная шина с передатчиком и приемником данных.

Если шина предназначена для передачи информации только в одном направлении, то она называется однонаправленной. Если данные могут передаваться в обоих направлениях, то такая шина называется двунаправленной.

Передача данных по шине может быть параллельной или последовательной. Системы шин разделяются на параллельные и последовательные. В параллельных системах для каждого бита передаваемого бинарного слова выделен отдельный провод. Для передачи 8-битового слова необходимо 8 проводов. Эти 8 проводов называются линиями данных. Для управляющих сигналов необходимы дополнительные линии (рис. 11.22).

В последовательных шинах для передачи данных достаточно одной линии. Отдельные биты передаются по очереди и объединяются в месте приема в первоначальное бинарное слово (рис. 11.23).

Шины последовательной передачи данных работают медленнее, чем системы параллельного обмена. Для последовательной передачи необходимы дорогие схемы параллельно-последовательных и последовательно-параллельных преобразователей. Поэтому шины последовательного обмена

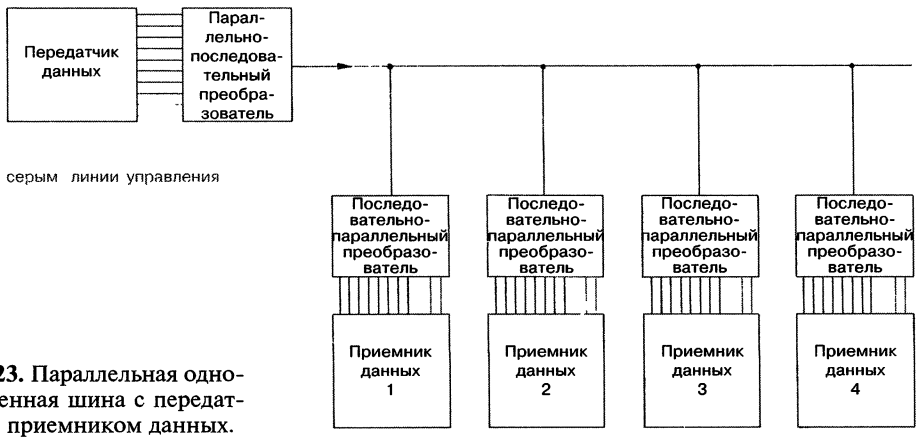


Рис. 11.23. Параллельная однонаправленная шина с передатчиком и приемником данных.

используются только там, где дополнительные линии стоят очень дорого, например, при больших расстояниях между передатчиком и приемником данных.

В большинстве случаев расстояние между передатчиком и приемником незначительно, так что параллельная шинная система является хорошим решением.

Каждая шина должна уметь выбрать нужный приемник данных. Выборка происходит с помощью управляющих входов. Простой пример показан на рис. 11.22. Через две линии управления могут задаваться четыре различных команды. Каждая команда включает один из приемников данных на прием.

Если приемники данных не работают на прием, то их информационные входы должны быть высокоомными. Приемник данных не должен влиять на сигналы, находящиеся на линиях шины. В схемах МОП-семейства входы всегда высокоомные. Здесь не возникает больших проблем. В схемах ТТЛ-семейства наряду с двумя обычными *L* и *H*-состояниями должно быть третье высокоомное состояние на входе. Высокоомное состояние обеспечивается специальными дополнениями в схеме. ТТЛ-схемы, входы которых (и выходы) имеют высокоомное состояние, называются TRI-STATE-ТТЛ-схемы. (TRI-STATE — три возможных состояния).

К шине также могут присоединяться несколько передатчиков данных. Само собой разумеется, что должна обеспечиваться невозможность передачи данных двумя передатчиками одновременно. Неактивный передатчик данных не должен влиять на находящиеся на линиях шины сигналы. Выходы должны быть высокоомными.

Элементы шины могут работать время от времени и как передатчики данных, и как приемники. В такой системе высокие требования предъявляются к управляющим сигналам. Шина работает в двунаправленном режиме.

Шинные системы передачи данных используются в областях, где необходима передача данных различным приемникам с возможностью их выборки. В основном это микропроцессорная техника (см. гл. 15) и системы цифровой обработки данных.

11.4.2. Стандарты шин

Шинные системы передачи данных могут быть организованы в множестве вариантов. Можно конфигурировать шины, начиная с числа линий данных и заканчивая линиями управления.

Каждая микропроцессорная система имеет собственную шинную систему. Обычно эта система состоит из 8-битовой шины данных, 8-битовой шины команд и 16-битовой шины адреса. По шине адреса передаются адреса памяти (см. гл. 15). Всего в шине 32 линии.

Тенденция развития шинных систем складывается в сторону 16-битных локальных шин данных, 24-битовых локальных адресных шин и 16-битовых локальных шин команд, т. е. общих 56-битовых шин. Чтобы элементы различных производителей могли работать друг с другом в шинной системе, возникла необходимость в стандартизации шин.

Некоторые стандарты для шин уже сформировались. Это, например, система S100 или Altair. Эта шинная система использует 100 линий, т. е. стандарт с запасом для расширения.

Еще один стандарт для шинных систем передачи данных — это IEC-шина, также называемая IEEE-488-шиной (IEEE — Institute of Electronic and Electrical Engineers). Этот стандарт используется прежде всего в компьютерной технике.

Контрольный тест

1. Как работает цифровой селектор?
2. Объясните различие между мультиплексором и демультиплексором.
3. Разработайте схему мультиплексора «8 в 1».
4. Как построен селектор «3 × 4 в 4»? Сколько необходимо управляющих входов? Нарисуйте блок-схему аналогично рис. 11.4.
5. Разработайте схему демультиплексора «2 × 2 в 2».
6. Как работает дешифратор адреса?
7. Нарисуйте схему 3-битового дешифратора.
8. Объясните принцип действия цифрового 1-битового компаратора.
9. Что понимают под шинной системой?
10. Какие преимущества имеют стандартизованные шинные системы?

ГЛАВА 12

РЕГИСТРЫ И ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

12.1. Сдвигающие регистры

Сдвигающие регистры являются схемами, которые принимают информацию последовательно во времени, бит за битом, сохраняют в течение определенного времени и передают дальше.

Для построения сдвигающих регистров используются триггеры. Хорошо подходят управляемые по фронту импульса *D*-триггеры, *RS*-триггеры и *JK*-триггеры.

Высококачественные сдвигающие регистры строятся часто на *JK*-MASTER-SLAVE-триггерах. Различные популярные сдвигающие регистры производятся в виде интегральных микросхем.

12.1.1. Сдвигающие последовательные регистры

Простой 4-разрядный сдвигающий регистр показан на рис. 12.1. Он состоит из четырех *D*-триггеров, которые переключаются прямым фронтом. Принцип действия *D*-триггеров объяснен в разд. 7.7.7.

Если на входе *E* действует 1-сигнал и сигнал синхронизации изменяется с 0 на 1, то триггер *A* устанавливается. На его выходе Q_A появляется 1. Если на вход поступает 0-сигнал, то со 2-м передним фронтом синхроимпульса триггер *A* сбросится и установится триггер *B*. Сигнал 1 появится на выходе Q_B. С 3-м передним фронтом синхроимпульса триггер B сбросится и установится триггер *C*. Q_C будет 1. С 4-м передним фронтом синхроимпульса триггер *C* сбросится и установится триггер *D* ($Q_D = 1$).

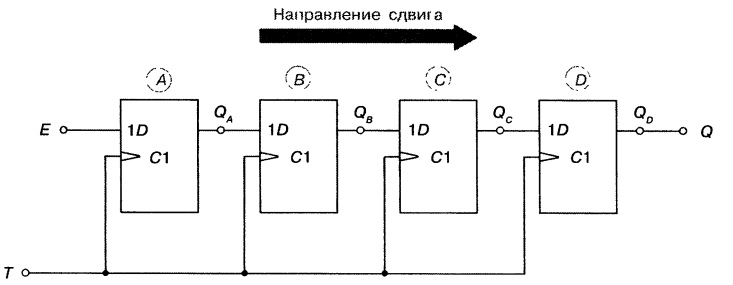


Рис. 12.1. 4-разрядный сдвиговый регистр для последовательного ввода-вывода.

Номер такта	Состояния после такта номер				
	E	Q_A	Q_B	Q_C	$Q_D = Q$
	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1
5	0	0	0	0	0

Рис. 12.2. Функциональная таблица 4-битового сдвигающего регистра.

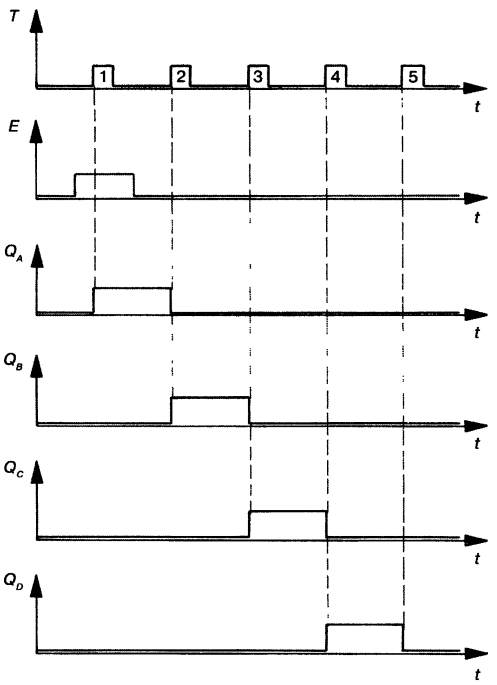


Рис. 12.3. Временная диаграмма 4-битового сдвигающего регистра.

1-сигнал, который в начале был приложен ко входу E, последовательно переходит от триггера к триггеру. Сейчас он действует на выходе сдвигающего регистра. С передним фронтом 5-го такта триггер D также сбрасывается. Говорят, сдвигающий регистр «пустой». Он не содержит больше никакой информации.

На рис. 12.2 в функциональной таблице представлены пошаговые сдвиги. На рис. 12.3 изображена временная диаграмма.

Какая временная диаграмма получится, если перед 3-м тактом к входу E снова будет приложен 1-сигнал, и этот 1-сигнал будет действовать и после 3-го такта?

Если перед 3-м тактом к входу E снова будет приложен 1-сигнал, то триггер A установится с передним фронтом 3-го такта. С передним фронтом 4-го такта триггер A сбросится, а триггер B установится. С передним фронтом 5-го такта сбросится триггер B и установится триггер C — и так далее. Временная диаграмма изображена на рис. 12.4.

Подаваемые на вход 1- и 0-сигналы принимаются в сдвигающий регистр с временным сдвигом или по очереди (последовательный ввод данных). После приема информации сигналы синхронизации могут больше не подаваться. Информация сохраняется до тех пор, пока снова не будут подаваться сигналы синхронизации. После подачи сигналов синхронизации информация бит за битом будет подаваться на выход Q (последовательный вывод данных).

Пример

Двоичное число 0101 должно быть записано в сдвигающий регистр (рис. 12.1). Для этого необходимо четыре такта. Перед 1-м тактом содержа-

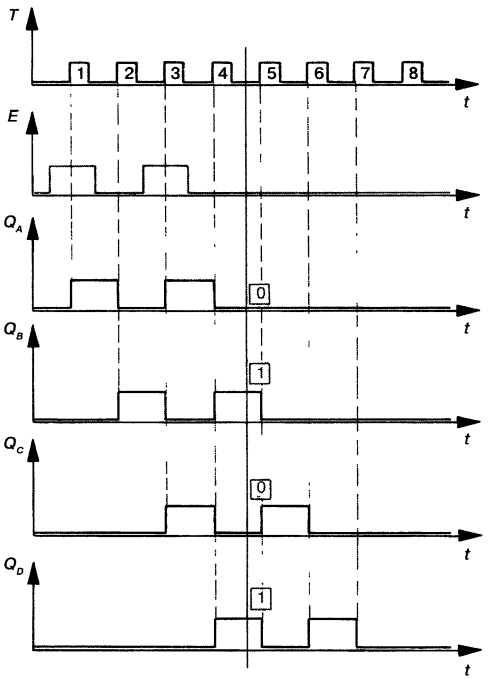


Рис. 12.4. Временная диаграмма.

Номер такта	Состояния после такта номер				
	E	2^3 Q_A	2^2 Q_B	2^1 Q_C	2^0 Q_D
	1	0	0	0	0
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	1	0
4	0	0	1	0	1
5	0	0	0	1	0
6	0	0	0	0	1
7	0	0	0	0	0
8	0	0	0	0	0

Рис. 12.5. Функциональная таблица.

ние первого бита (значение 2^0) должно действовать на входе E (1-сигнал). Перед 2-м тактом содержание второго бита (значение 2^1) должно действовать на входе E. Это в данном примере 0-сигнал. Перед 3-м тактом содержание третьего бита (значение 2^2) должно действовать на входе E (1-сигнал). Перед 4-м тактом содержание четвертого бита (значение 2^3) должно действовать на входе E (0-сигнал). После 4-го такта введено двоичное число 0101. Временная диаграмма процесса изображена на рис. 12.4.

Теперь сигналы синхронизации можно больше не подавать. Информация будет сохранена в регистре в течение длительного времени.

Для трансляции данных через выход Q требуются еще 4 такта. Первый бит оказывается на выходе Q перед 5-м тактом, 2 бит — после 5-го такта. 3-й бит после 6-го такта, и 4-й бит после 7-го такта. После 8-го такта сдвигающий регистр пуст (рис. 12.5).

Условное графическое обозначение сдвигающих регистров определяется DIN 40900, часть 12. Для триггера на рис. 12.6 оно изображено на рис. 12.1. SRG означает сдвигающий регистр (от англ. shift register). Число после SRG является числом битов.

Сдвигающие регистры производятся в больших количествах в виде интегральных микросхем. Интегральная микросхема FLJ 221-7491A содержит 8-битовый сдвигающий регистр для последовательного ввода—вывода. Цо-

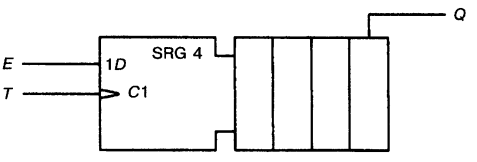


Рис. 12.6. Условное обозначение 4-битового сдвигающего регистра с последовательным вводом—выводом на базе D-триггера.

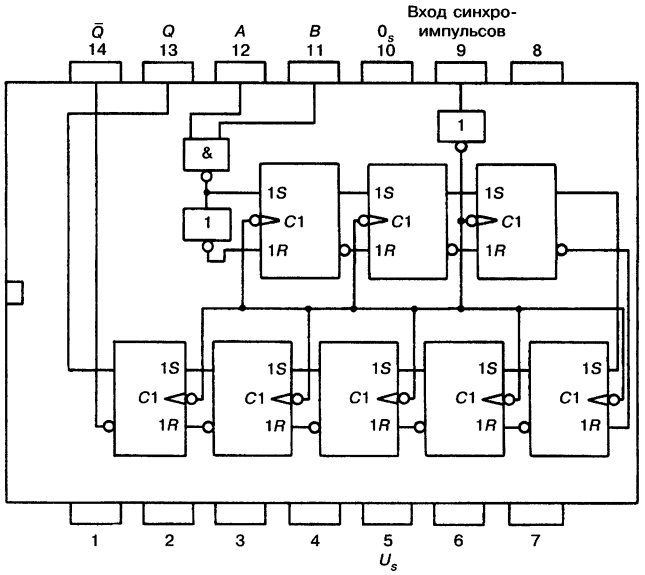


Рис. 12.7. Цоколевка и блок-схема интегральной микросхемы FLJ 221-7491A.

колевка и блок-схема этой микросхемы представлены на рис. 12.7. Сдвигающий регистр построен на *RS*-триггерах, переключающихся с обратным фронтом синхроимпульсов. Так как в линию управления включен элемент НЕ, то переключение происходит с передним фронтом синхроимпульса на входе 9.

12.1.2. Сдвигающие регистры с параллельным считыванием

Сдвигающие регистры всегда имеют возможность последовательного ввода—вывода данных. Без этого свойства схема не может называться сдвигающим регистром.

Сдвигающий регистр с параллельным считыванием может транслировать хранимые данные одновременно. Сдвигающий регистр на рис. 12.8 имеет возможность параллельного считывания. *Q*-выводы триггеров выведены на особенные клеммы микросхемы. На них в распоряжении разработчика имеются все четыре бита.

Для построения сдвигающего регистра использовались *RS*-триггеры. Эти триггеры сбрасываются иначе, чем *D*-триггеры (см. разд. 7.5.2). Сбрасывание триггера происходит только в том случае, если на *R*-входе действует 1-сигнал при фронте синхроимпульса. 0-сигналы не приводят к переключению триггеров. Поэтому входной сигнал должен подаваться на НЕ-элемент и *R*-вход

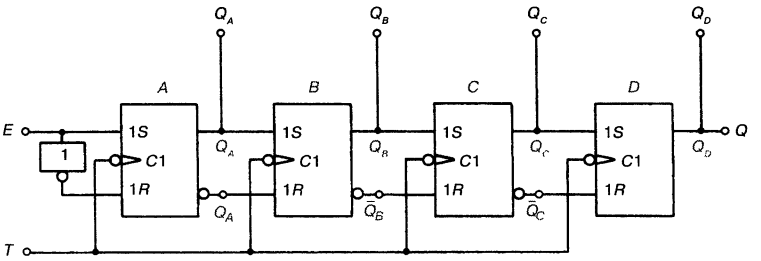


Рис. 12.8. Сдвигающий регистр с параллельным считыванием.

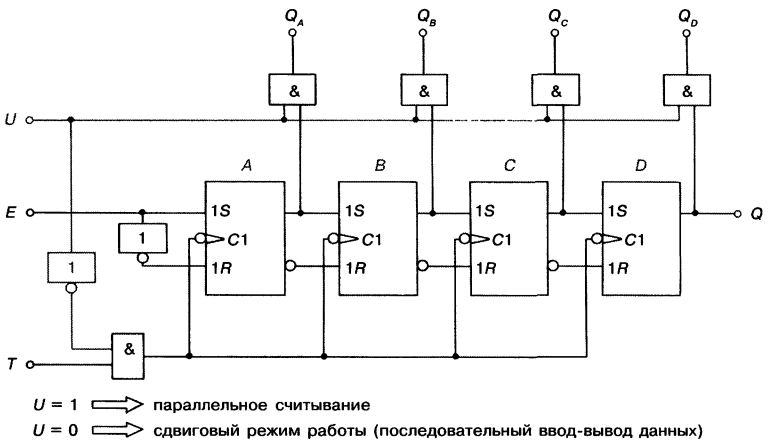


Рис. 12.9. Сдвигающий регистр с параллельным считыванием и схемой блокировки.

первого триггера. Если на входе E действует 0-сигнал, то на входе R первого триггера действует 1-сигнал и триггер сбрасывается с приходом следующего синхроимпульса.

Каждый Q-выход мог бы быть связан с R-входом следующего триггера через элемент НЕ. Элемент НЕ можно не применять, так как инвертированный Q-сигнал имеется в распоряжении на выходе $\overline{Q}$. Выход $\overline{Q}$ непосредственно связывается с R-входом следующего триггера (см. рис. 12.8).

Хранимая в сдвигающем регистре информация может быть считана на выходах Q_A, Q_B, Q_C и Q_D. Во время параллельного считывания сдвигающий регистр не должен производить сдвиговых операций, иначе считываемые данные будут искажены. Таким образом, последовательное и параллельное считывания одновременно производиться не могут.

Сдвигающий регистр на рис. 12.9 имеет схему блокировки, которая предотвращает сдвигающие операции регистра при проведении параллельного считывания и, с другой стороны, блокирует параллельное считывание, если сдвигающий регистр производит сдвигающие операции. Если на входе U действует 0-сигнал, то возможна подача синхроимпульса и параллельное считывание заблокировано. Если на входе U действует 1-сигнал, то возможно параллельное считывание, а подача сдвигающих синхроимпульсов заблокирована.

12.1.3. Сдвигающие регистры с параллельным вводом—выводом данных

Для многих задач удобно иметь наряду с последовательным вводом данных возможность параллельного ввода данных. Это параллельная запись данных может происходить как тактируемо, так и не тактируемо.

Сдвигающий регистр на рис. 12.10 предлагает наряду с возможностью параллельного считывания возможность параллельной записи данных. Параллельные считывание и запись не тактируемы. Сдвигающий регистр построен на JK-триггерах, которые имеют нетактируемые входы установки и сброса. Информационными входами для параллельного ввода данных являются A, B, C и D.

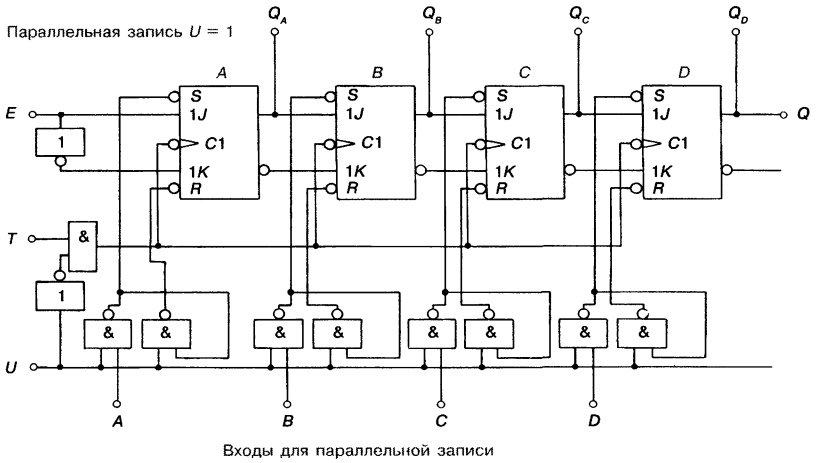


Рис. 12.10. Сдвигающий регистр с нетактируемым параллельным вводом-выводом данных.

Параллельный ввод данных и последовательный ввод—вывод являются взаимоисключающими операциями. Если на входе U действует 0-сигнал, то подача синхроимпульсов разрешена. Сдвигающий регистр может работать в режиме последовательного ввода. При $U = 1$ возможен параллельный ввод. Сигнал синхронизации заблокирован. При необходимости параллельный вывод данных также может быть заблокирован, как показано на рис. 12.9.

Как построен сдвигающий регистр с параллельным тактирующим вводом данных? Есть различные варианты схемы такого регистра. Один из них показан на рис. 12.11. *J*- и *K*-входы каждого триггера выполнены переключаемыми. При $U = 0$ сигналы на входы триггеров подаются последовательно. При $U = 1$ сигналы на входы триггеров подаются с входов для парал-

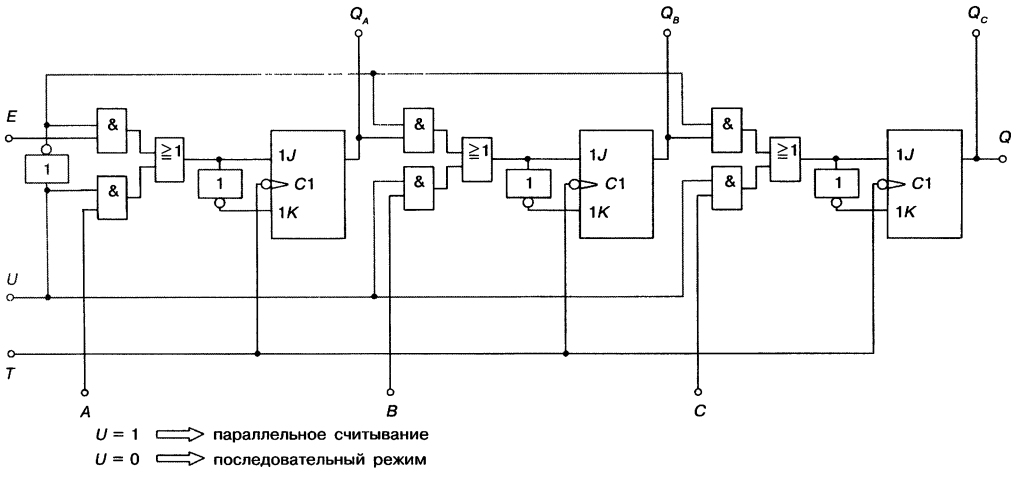


Рис. 12.11. Сдвигающий регистр с тактируемым параллельным вводом и нетактируемым выводом данных.

лельной записи (*A*, *B*, *C*, *D*). Установка или сброс триггеров происходит с обратным фронтом синхроимпульсов в соответствии со свойствами триггеров, примененных в схеме на рис. 12.1.

12.1.4. Кольцевой сдвигающий регистр

Кольцевым сдвигающим регистром называется сдвигающий регистр, выход которого замкнут на вход.

В кольцевом сдвигающем регистре данные циркулируют в замкнутом цикле по кругу. Принципиальная схема кольцевого сдвигающего регистра представлена на рис. 12.12.

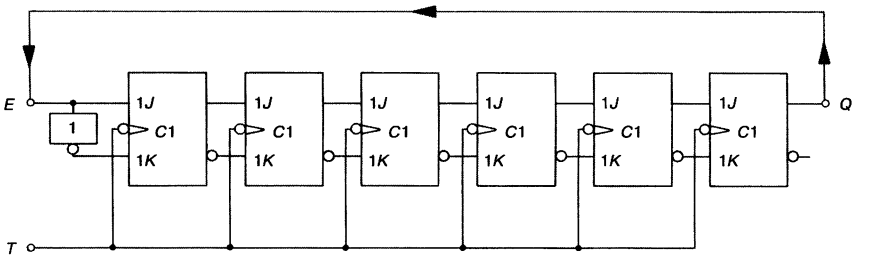


Рис. 12.12. Принципиальная схема кольцевого сдвигающего регистра.

Данные могут вводиться в кольцевой сдвигающий регистр последовательно или параллельно. Они могут считываться также последовательно или параллельно. На рис. 12.13 показан кольцевой сдвигающий регистр с последовательным вводом данных и на выбор последовательным или параллельным выводом данных. При $U = 1$ регистр соединен по схеме кольцевого сдвигающего регистра. Выходные сигналы поступают на вход. При $U = 0$ возможен ввод данных с последовательного входа E. Последовательный вывод данных возможен с выхода Q, если $K = 1$. Через вход R можно независимо от синхроимпульсов сбросить регистр 0-сигналом и удалить записанную в него информацию.

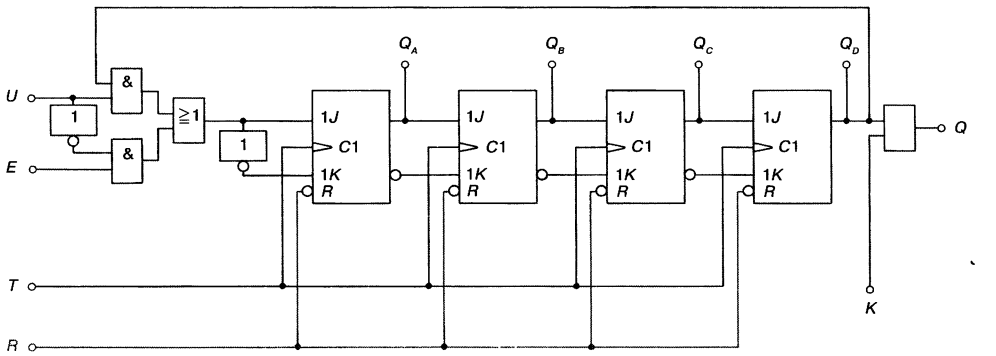


Рис. 12.13. Кольцевой регистр с последовательным вводом и на выбор последовательным или параллельным выводом.

12.1.5. Сдвигающие реверсивные регистры

Сдвигающие реверсивные регистры имеют большое значение для задач управления. Их структура базируется на структуре рассмотренных ранее сдвигающих регистров. Направление передачи информации в триггеры должно меняться.

Принципиальная схема сдвигающего реверсивного регистра изображена на рис. 12.14. Механические переключатели должны заменяться соответствующими цифровыми схемами. Черное положение ключа и линии действительны для сдвига вправо. Серое положение ключа и линии действительны для сдвига влево.

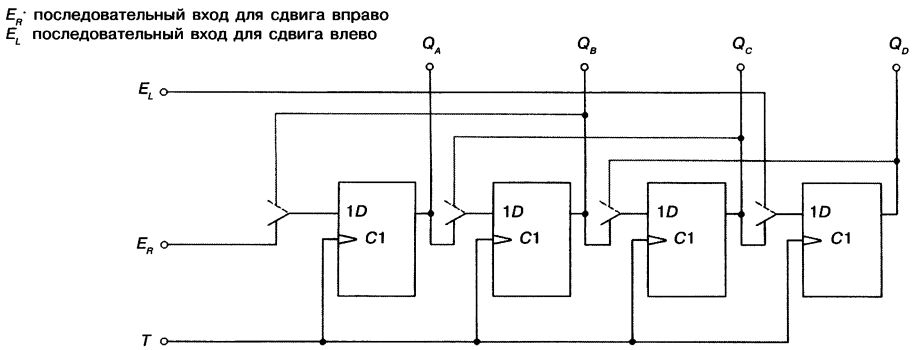


Рис. 12.14. Принципиальная схема реверсивного сдвигающего регистра.

8-битовый сдвигающий реверсивный регистр выпускается в виде микросхемы FLJ 311-74198. Этот сдвигающий регистр универсален. Он обладает возможностью последовательного и параллельного ввода—вывода данных, схемой блокировки синхроимпульсов и имеет независимый от синхроимпульсов вход сброса. Схема этого сдвигающего регистра и его цоколевка представлены на рис. 12.15.

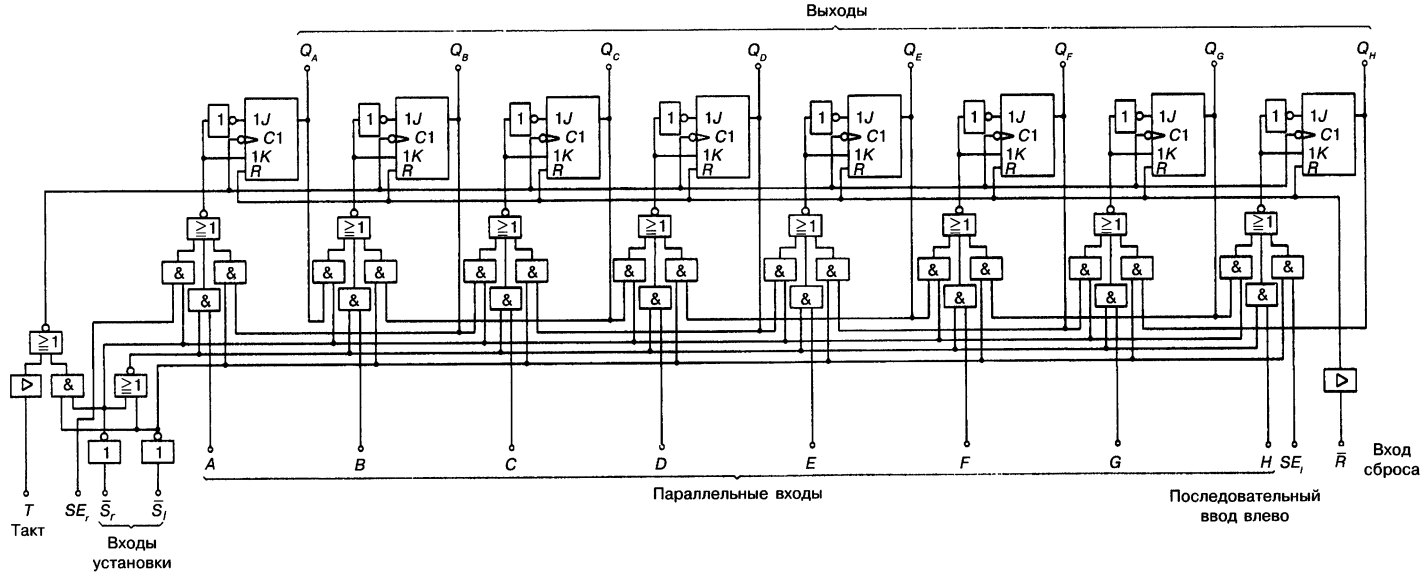
12.2. Регистры хранения

Регистры хранения, как и сдвигающие регистры, построены на базе триггеров, только в них не происходит перемещение битов данных. Отдельные триггеры устанавливаются или сбрасываются. Регистр памяти записывает бинарное слово фиксированной длины. Информация может считываться с выходов и передаваться дальше. Данные удаляются, если в них больше нет необходимости.

> Задачей регистров хранения является сохранение бинарных слов в течение определенного времени.

Структура 4-битового регистра хранения представлена на рис. 12.16. *JK*-триггеры устанавливаются 1-сигналами на *J*-входах. Установка производится синхроимпульсами. Каждый триггер может устанавливаться независимо. Сброс триггеров происходит подачей 1-сигналов на *K*-входы. Возможность сброса блокируется 0-сигналом на входе *L*. Для удаления всей хранимой информации предназначен нетактируемый вход сброса *R*.

Регистры хранения применяются в схемах вычисления и управления.



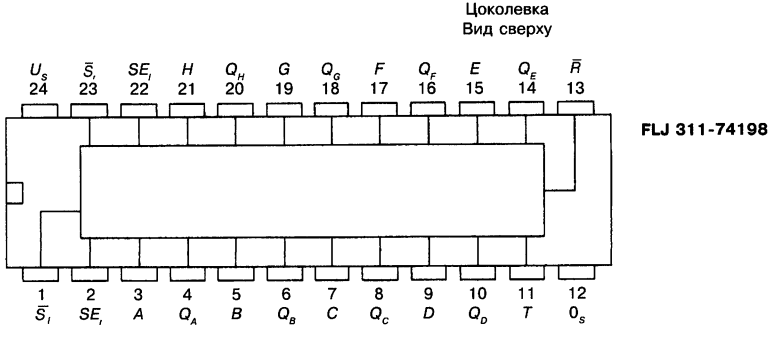


Рис. 12.15. Принципиальная схема и цоколевка реверсивного сдвигающего регистра FLJ 311-74198 (Siemens).

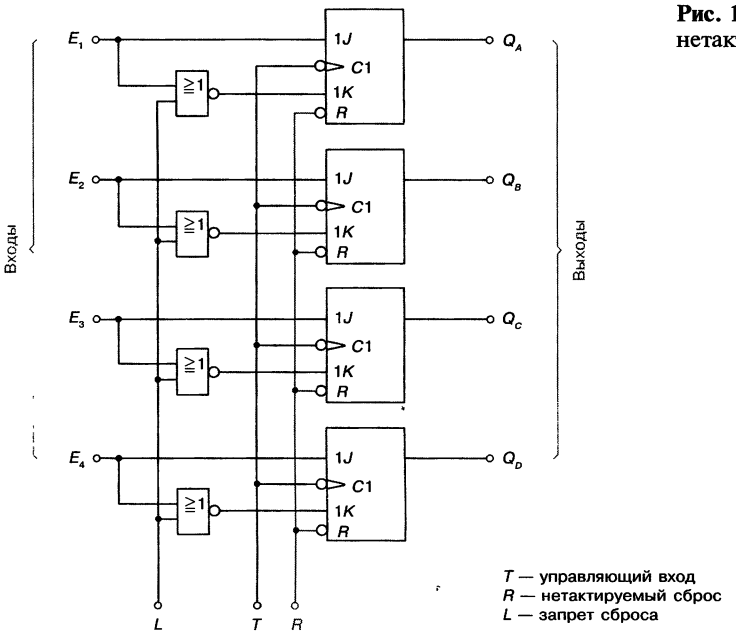


Рис. 12.16. Регистр хранения с нетактируемым входом сброса.

12.3. Оперативные запоминающие устройства (RAM)

Аббревиатурой RAM обозначаются оперативные запоминающие устройства на базе полупроводниковых технологий. Они имеют определенное количество ячеек памяти. Каждая ячейка памяти имеет установленный объем памяти. Она может принимать информацию определенной длины. Каждая ячейка памяти имеет индивидуальный адрес. По этим адресам осуществляется доступ к ячейкам памяти. То есть RAM является памятью с произвольной выборкой.

Обозначение RAM является сокращением от Random Access Memory, англ. — память с произвольной выборкой.

Ячейки памяти выбираются по своим адресам. В ячейках хранится записанная информация. Для вывода данных ячейки также выбираются по своему адресу. Считывание данных не стирает содержание ячейки. Если информация больше не нужна, она может быть удалена, а в ячейку памяти записана новая информация.

ОЗУ выпускаются исключительно в виде интегральных микросхем. Различают статические RAM и динамические ОЗУ. В статическом ОЗУ ячейки состоят из триггеров. Каждый бит записывается в один из триггеров. В динамическом ОЗУ для хранения информации используются внутренние емкости. Каждый бит записывается в маленький конденсатор. Так как имеют место токи утечки, то возникает потеря заряда, поэтому емкости периодически надо подвергать процессу регенерации (восстановление исходных уровней сигнала).

> Статические и динамические ОЗУ являются энергозависимыми. При отключении питания информация теряется.

В целях сохранения информации ставят автономные батареи. Энергонезависимые RAM находятся в стадии разработки. Их ждет большое будущее.

Статические ОЗУ или RAM производятся на базе различных технологий. Схемы принадлежат к различным семействам. Существует ОЗУ в ТТЛ-исполнении, в ЭСЛ-, в *N*-МОП- и в *K*-МОП-исполнении. Динамические ОЗУ строятся на различных МОП-элементах.

12.3.1. Статические ОЗУ (SRAM)

12.3.1.1. Элемент памяти RAM в ТТЛ-исполнении

Статические ОЗУ могут быть построены на биполярных транзисторах. Используется известная ТТЛ-технология (см. гл. 6 «Семейства схем»). Основой ячейки является триггер на двух мультиэмиттерных транзисторах согласно рис. 12.17. Триггер управляется *X*-адресной шиной, *Y*-адресной шиной и двумя разрядными линиями записи—считывания. Такой элемент памяти может запомнить 1 бит. Элемент содержит значение 1, если T_1 открыт, а T_2 заперт. И значение 0, если T_2 открыт, а T_1 заперт.

Активация ячейки памяти

Если на обоих адресных шинах *X* и *Y* действует 0-сигнал (0 В, заземление), то ячейка памяти не активирована. Ток эмиттера открытого транзистора течет на землю. По разрядным линиям SL_1, SL_2 ток не протекает.

Если только на одной адресной шине действует 1-сигнал, то элемент памяти остается неактивным, так как эмиттерный ток открытого транзис-

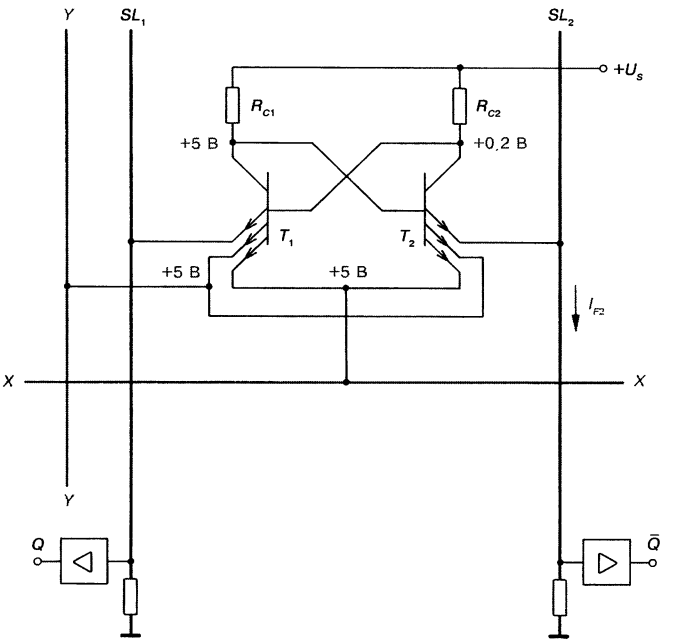


Рис. 12.17. Элемент памяти статической RAM объемом 1 бит в ТТЛ-исполнении (принципиальная схема).

тора может протекать через другую адресную шину. Только когда по обеим адресным шинам проходит 1-сигнал, то есть +5 В, элемент памяти активируется, и эмиттерный ток открытых транзисторов течет по соответствующим линиям записи—чтения.

Чтение

После активации элемента памяти по *SL*-линиям течет ток открытого транзистора. На рис. 12.17 транзистор T_2 открыт. Ток эмиттера протекает через SL_2 и на выходе $\bar{Q}$ через усилитель выдает 1-сигнал. Элемент памяти сохранил значение 0. Если после активации ток протекает по линии SL_1, то элемент памяти сохранил значение 1.

Запись

Для записи 1 в элемент, содержащий 0, необходимо подать 1-сигнал или +5 В на SL_2 и 0-сигнал, или 0 В на SL_1. При хранении 0 проводит транзистор T_2. Он запирается после подачи на его третий эмиттер, подключенный к SL_2, +5 В. Транзистор T_1 открывается, и его эмиттерный ток может течь через SL_1. После завершения активации элемент памяти сохраняет это состояние.

Для записи 0 в элемент, содержащий 1, необходимо подать 1-сигнал или +5 В на SL_1 и 0-сигнал, или 0 В на SL_2. Триггер переключится по алгоритму, описанному выше, и сохранит значение 0.

Элементы памяти на ТТЛ-элементах работают очень быстро. Но зато они отличаются высоким энергопотреблением.

12.3.1.2. Элементы памяти RAM в N-МОП-исполнении

Элементы памяти RAM в *N*-МОП-исполнении имеют по сравнению с ТТЛ-элементами существенные преимущества. Они потребляют меньше энергии и могут производиться с более высокой степенью интеграции. На квадратном миллиметре чипа может размещаться больше МОП-элементов памяти. Однако МОП-схемы имеют большее время переключения, т. е. они медленнее ТТЛ-схем (см. гл. 6 «Семейства схем»).

Структура типичного RAM-элемента представлена на рис. 12.18. Транзисторы T_1 и T_2 собраны по схеме триггера. Транзисторы T_3 и T_4 работают вместо нагрузочных сопротивлений. Если T_1 заперт, а T_2 открыт, то элемент памяти сохранил значение 1. При 0 T_1 открыт, а T_2 заперт.

Активация элемента памяти

Элемент активизируется, если на адресные шины *X* и *Y* подается 1-сигнал. Транзисторы T_5, T_6, T_7 и T_8 открываются и соединяют выходы триггера *Q* и $\bar{Q}$ с линиями чтения—записи SL_1 и SL_2.

Чтение

После активации ячейки памяти из нее можно считывать данные. Если SL_1 проводит 1-сигнал, то элемент сохраняет единицу. Если SL_2 проводит 1-сигнал, то элемент сохраняет 0.

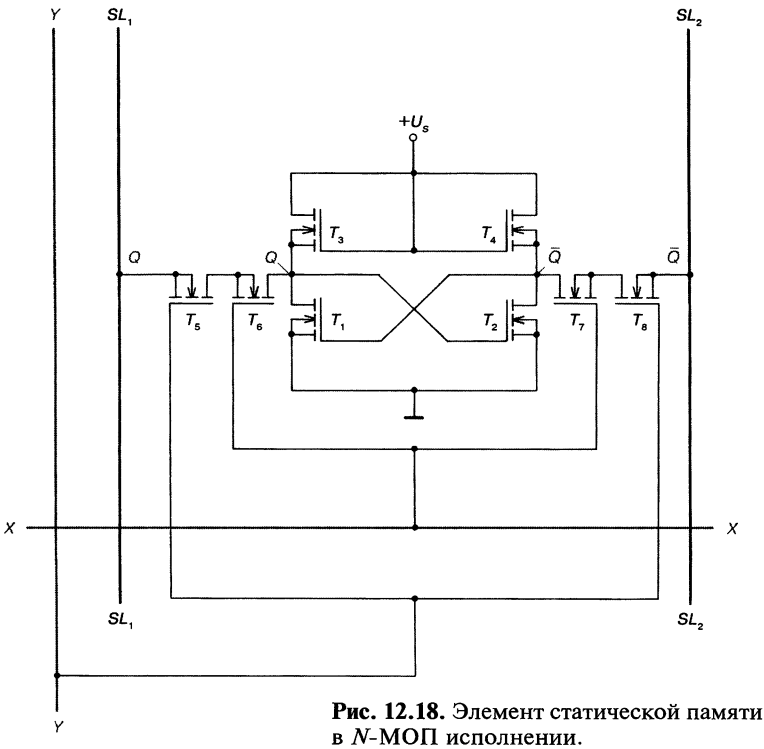


Рис. 12.18. Элемент статической памяти RAM объемом в 1 бит в *N*-МОП исполнении.

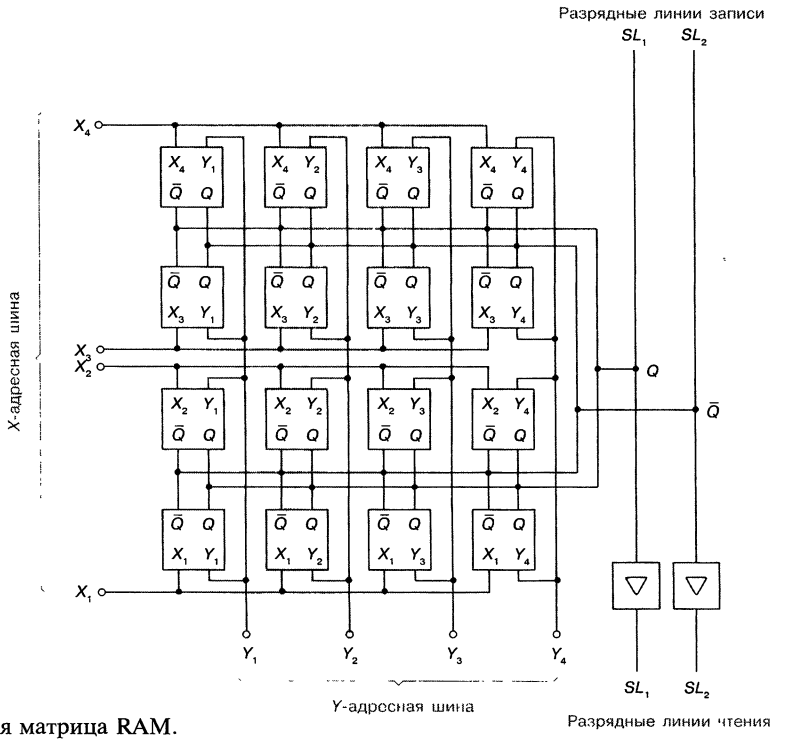


Рис. 12.19. 16-битовая матрица RAM.

Запись

В элемент памяти, содержащий 0, требуется записать 1. При 0 T_1 открыт, а T_2 заперт. Если на SL_2 действует 0-сигнал, то T_1 запирается, а T_2 открывается. Триггер переключается в 1-состояние. Это состояние сохраняется после завершения активации.

12.3.1.3. Структура RAM с двухкоординатной адресацией

RAM-элементы памяти собираются в RAM-матрицы памяти. Матрица на рис. 12.19 имеет объем памяти 16 бит. Каждый элемент доступен индивидуально. Говорят, что каждый бит может быть адресован. Если бы мы хотели узнать, например, содержимое памяти запоминающего элемента 8, то на адресных шинах X_3 и Y_4 должен быть 1-сигнал. На разрядных линиях SL_1 и SL_2 появляются выходные сигналы Q и $\bar{Q}$.

12.3.2. Динамические ОЗУ (DRAM)

12.3.2.1. Элемент памяти динамического ОЗУ

Типичный элемент памяти динамического ОЗУ состоит из трех самозапирающихся MOS-FET-транзисторов согласно рис. 12.20. Информация сохраняется в емкости C. Если C заряжен, то элемент памяти содержит 1. Если C не заряжен, то элемент памяти содержит 0.

Запись

Запоминающий элемент активизируется 1-сигналом на шине чтения X (1-сигнал = +5 В). Вследствие этого транзистор T_1 открывается (низкоомен между истоком и стоком). При подаче на информационный вход A 1-сигнала

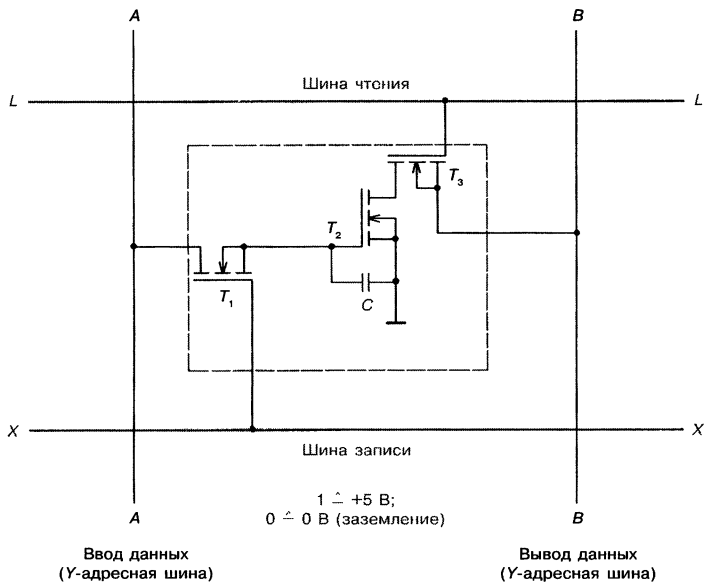


Рис. 12.20. Типичный элемент памяти динамической RAM.

емкость C заряжается. Значение 1 сохранено. При заряженном конденсаторе C транзистор T_2 всегда низкоомен (открыт). Если на адресной шине X действует 0-сигнал, то элемент памяти больше не активизирован. Транзистор T_1 запирается и предотвращает утечку заряда C. Для записи 0 следует активизировать элемент памяти (1-сигнал на адресной шине X). При этом T_1 отпирается. Если 0-сигнал (0 ≙ 0 В, земля) прикладывается на линию входных данных A, то емкость C может разрядиться через транзистор T_1. Значение 0 сохранено. При разряженном конденсаторе C транзистор T_2 всегда заперт.

Чтение

Для считывания данных на шину вывода B прикладывается 1-сигнал (+ 5 В). Элемент памяти активизируется через шину чтения L. На L также прикладывается 1. При этом транзистор T_3 открывается.

Если записана 1, то транзистор T_3 открыт и ток через шину В через T_3 и T_2 течет на землю. Это признак записанной 1.

Если записан 0, то C разряжен, а T_2 заперт. Через шину В ток течь не может. Это признак записанного 0.

Во время чтения содержание элемента памяти не меняется.

Регенерация

Емкость C очень мала. Она составляет в зависимости от степени интеграции от 0,1 пФ до 1 пФ. Соответственно мал и сохраненный заряд. Очень маленький ток утечки быстро понижает заряд. Поэтому заряд должен быть через короткие промежутки времени снова восстановлен. Обычно подзарядка происходит каждые 2 мс.

Цикл регенерации начинается с чтения содержимого элемента памяти. Если оно равно 1, то T_1 открывается и C заряжается. Если содержание памяти 0, то зарядка не происходит.

Для регенерации необходим особенный задающий генератор синхронизирующих импульсов и схема управления. Оба этих элемента включены в состав микросхем.

12.3.2.2. Особенности динамических ОЗУ

Динамические ОЗУ работают очень надежно. Это удивительно, если учесть, что хранимая информация должна быть примерно каждые 2 мс обновлена. Можно было бы предположить, что какой-нибудь бит пропадает. Однако этого не происходит.

Преимуществом динамических ОЗУ является большой объем памяти чипа или микросхемы. МОП-техника позволяет достичь высокой степени интеграции. Запоминающие элементы могут быть очень малы. Предложенное в настоящее время самое большое динамическое ОЗУ имеет объем памяти 4 Мбит (1 Мбит = 1048576 бит, приблизительно 1 млн бит). 4 Мбит могут сохраняться в одной-единственной микросхеме. В разработке находится схема с 16 Мбит.

Недостатком является относительно большое время переключения. Так называемое время доступа лежит в интервале между 100 нс и 300 нс. Под

временем доступа понимают максимальное время, которое проходит от адресации элемента памяти до возможности работать с его данными.

Во время регенерации динамическое ОЗУ должно быть блокировано от записи и чтения, иначе будут возникать ошибки.

С ростом температуры кристалла токи утечки будут расти. Емкости будут разряжаться быстрее. Рекомендованная производителем частота регенерации рассчитана на рабочую температуру 70 °С. Если эта температура будет превышена, то данные могут быть потеряны.

12.3.3. Организация элементов памяти и ее параметры

12.3.3.1. Организация элементов памяти

Статические и динамические ОЗУ предлагаются с различными объемами памяти и различными структурами.

Адресуемая ячейка памяти может состоять из одного элемента или из нескольких элементов. Если она состоит только из одного элемента, то такая память называется памятью с побитовой организацией. Каждый запоминающий элемент, то есть каждый бит, имеет собственный адрес и является, таким образом, адресуемым. Конструктивная схема такой памяти показана на рис. 12.21. Обозначение 16 × 1 значит: совокупная емкость 16 бит, емкость ячейки памяти 1 бит.

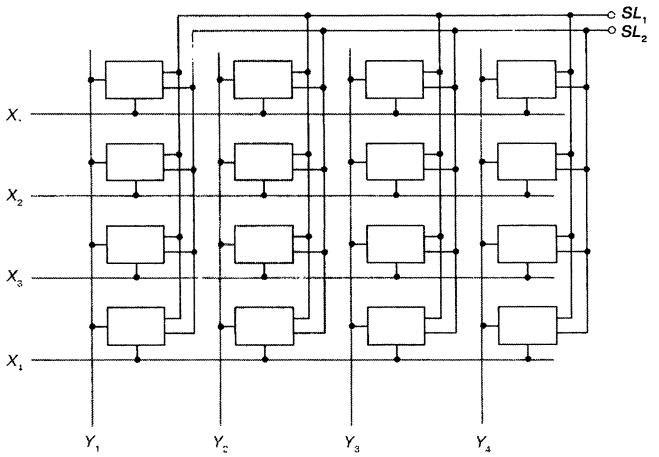


Рис. 12.21. Структура памяти 16 × 1.

Если ячейка памяти состоит из нескольких запоминающих элементов, то память является пословно организованной. На рис. 12.22 показана конструктивная схема 32 × 8-битово элемент является адресуемым. Все 8 бит ячейки памяти всегда записываются и считываются одновременно.

Для 256 × 1-памяти необходимы 16 *X*-адресных шин и 16 *Y*-адресных шин (рис. 12.23). Выводить эти шины наружу в порты микросхемы неудобно. Эта схема имела бы очень много выводов. Поэтому применяются дешифраторы (см. разд. 11.2). Для выбора 16 элементов памяти нужны 4 адресных линии. Адресные линии выводятся на ножки микросхемы.

8 линий SL_1 и 8 линий SL_2 частично не показаны для лучшей наглядности

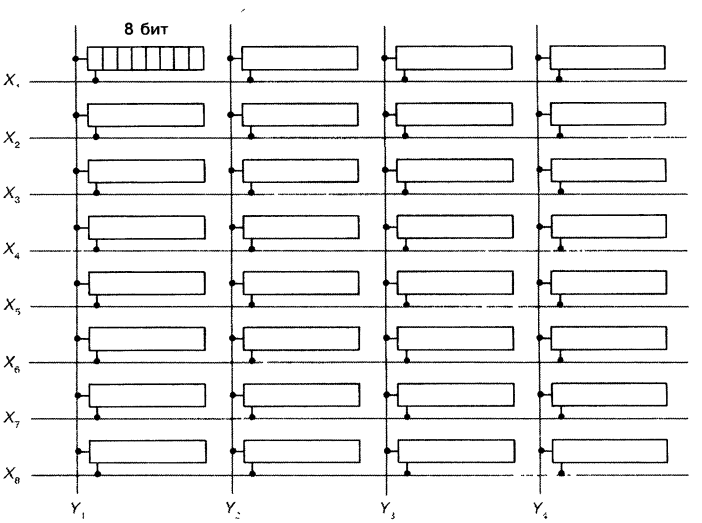
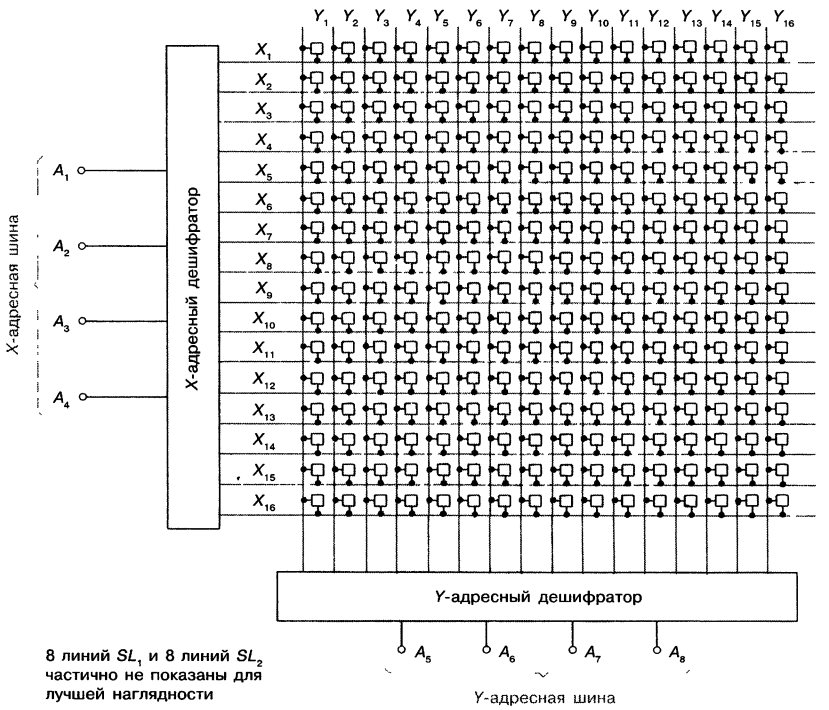


Рис. 12.22. Структура памяти 32 × 8.



8 линий SL_1 и 8 линий SL_2 частично не показаны для лучшей наглядности

Рис. 12.23. Структура памяти 256 × 1 с дешифратором.

Как организованы адресные шины в 16 Кбит × 1-битовой памяти? Должны быть адресованы 16 384 бит. Кроме этого, требуются 128 *X*-адресных линии и 128 *Y*-адресных линии. Для выбора 128 адресных линий необходимо иметь 7 управляющих выводов (рис. 12.24). Всего на выводы микросхемы нужно выводить 14 адресных линий. Так как требуются еще информационные выводы для ввода—вывода данных и для команд управления, то получается очень большое число выводов. Чтобы этого избежать, ставят

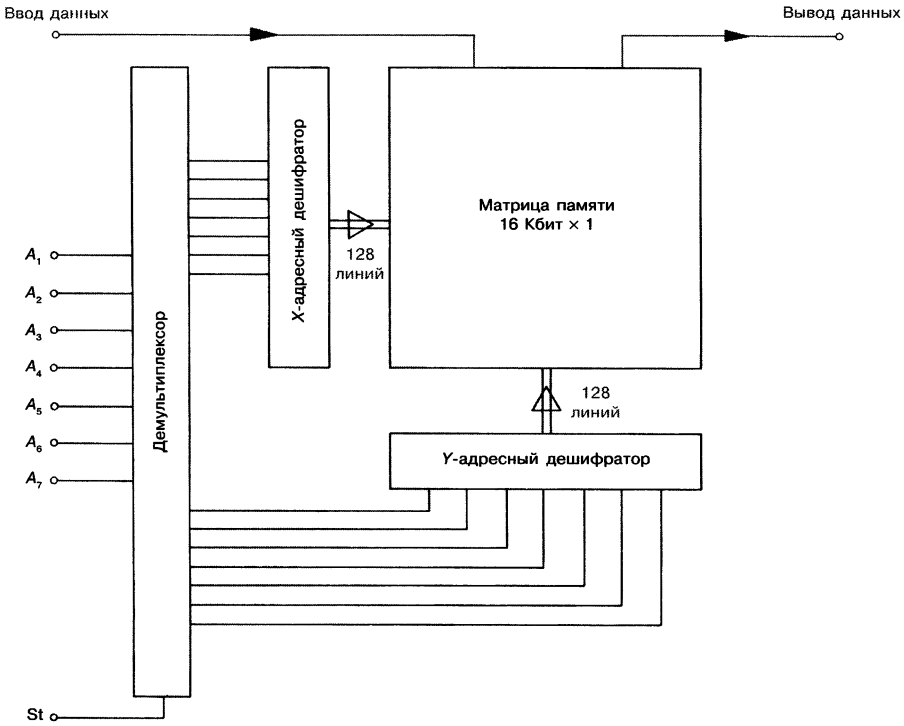


Рис. 12.24. Структура памяти 16 Кбит × 1 с дешифратором и демультиплексором.

демультиплексор (см. разд. 11.1). На входы от A_1 до A_7 сначала подается *X*-адрес, затем *Y*-адрес. Переключение происходит сигналом управления *S*. Мультиплексирование адресов позволяет применять небольшие корпуса ИС.

12.3.3.2. Параметры памяти

Для выбора памяти большое значение имеют ее параметры. В основном это емкость и структура, производительность и энергопотребление. Далее также важны электрические условия эксплуатации и допустимый рабочий диапазон. Рассмотрим по очереди самые важные параметры памяти.

Емкость

Емкость показывает количество элементов памяти, содержащихся в матрице, т. е. число бит, которые могут быть сохранены.

Структура памяти

Характеризует объем памяти одной ячейки и способ адресации.

Время доступа

Время доступа является временем, которое проходит от момента адресации элемента памяти (ЭП) до возможности располагать информацией на выходе данных.

Длительность цикла обработки

Под длительностью цикла обработки понимают минимальное время между двумя следующими друг за другом процессами чтения—записи.

Энергопотребление

Указывается общее энергопотребление всей микросхемы. Оно может варьироваться в зависимости от режима работы.

Электрические условия эксплуатации

Здесь указываются необходимые напряжения питания, необходимые уровни сигнала и диапазоны допустимых значений (см. гл. 6 «Семейства схем»), а также предельные значения других электрических величин.

Диапазон рабочих температур

Диапазон рабочих температур — диапазон температур, в котором память стабильно работает в рамках предписанных электрических условий эксплуатации.

12.3.3.3. Некоторые виды ОЗУ

Рассмотрим некоторые из выпускаемых схем памяти. Память SAB 2102 представляет из себя статическое 1024 × 1-битовое ОЗУ в *N*-МОП-исполнении. Она поставляется в 16-полюсных DIP-корпусах. Схема и расположение портов показаны на рис. 12.25.

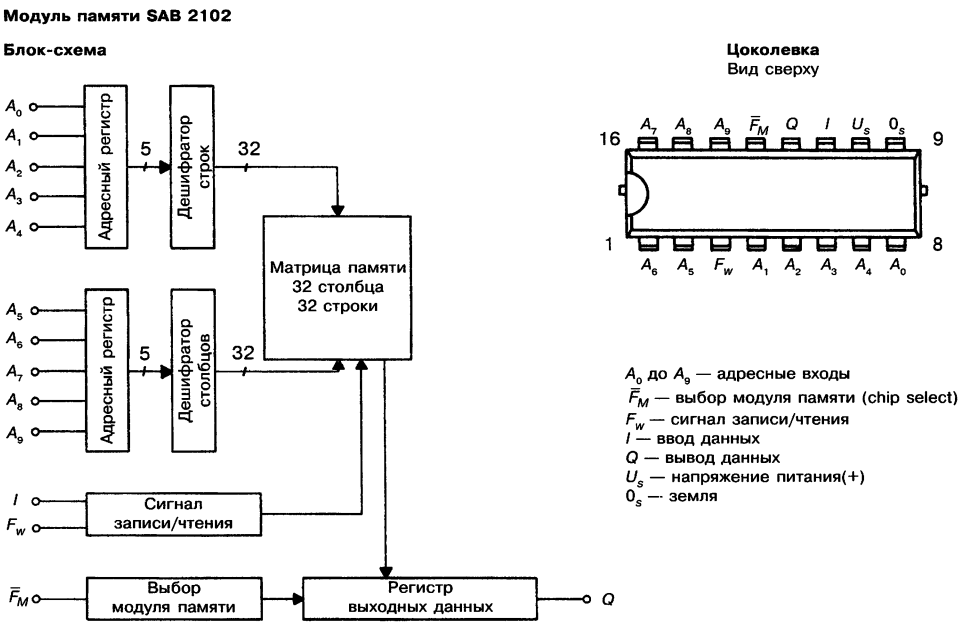


Рис. 12.25. Блок-схема и цоколевка 1024 × 1 битового RAM SAB2102 (Siemens).

X- и *Y*-адреса сохраняются в регистрах и декодируются. Матрица памяти имеет в строке 32 запоминающих элемента и 32 таких строки далее, одна под другой.

Для управления используются входы F_W (разрешить запись) и $\overline{F}_M$ (разрешить запоминание). Для разрешения запоминания требуется 0-сигнал. Если 0 не приложен, то память заблокирована. Для разрешения записи также нужен 0-сигнал. Информация сохраняется в регистре выходных данных.

Таблица данных блока памяти представлена на рис. 12.26. Максимальное время доступа 1000 нс. Память работает относительно медленно.

Статические параметры

При T = от 0 до 70 °C, U_S = +5 В ± 5 %, 0_S = 0 В

		Минимум	Максимум	Условия
Входной ток	I_I		10 мкА	U_I = 0 до 5,25 В
H-выходной ток	I_{QH}		10 мкА	U_Q = 4,0 В, F_M = 2,2 В
L-выходной ток	I_{QL}		−100 мкА	U_Q = 0,45 В, F_M = 2,2 В
Ток питания	$I_{S\,25}$		60 мА	T_U = 25 °C, все входы 5,25 В открыты *Q*
Ток питания	$I_{S\,0}$		70 мА	T_U = 0 °C, все входы 5,25 В открыты *Q*
H-напряжение питания	U_{IH}	2,2 В	U_S	
L-напряжение питания	U_{IL}	−0,5 В	0,65 В	
L-выходное напряжение	U_{QL}		0,45 В	I_{QL} = 19 мА
H-выходное напряжение	U_{QH}	2,2 В		I_{QH} = −100 мкА

Время переключения

T_U = 0 до 70 °C, U_S = 5 В ± 5 %, 0_S = 0 В

		Условия испытаний	Граничные значения		Единицы измерения
			min	max	
Цикл чтения					
Время цикла	t_{CR}		1000		нс
Время доступа (через адрес)	t_Z	U_I = 0,65 до 2,2 В		1000	нс
Время включения модуля памяти (через $\overline{F}_M$)		t_T = 20 нс		500	нс
Время отключения *Q* при смене адреса	t_{HQ}	C_L = 100 пФ	50		нс
Время отключения *Q* ($\overline{F}_M$ тактируемое)	t_{DF}	1 TTL-нагрузка	0		нс

		Условия испытаний	Граничные значения		Единицы измерения
			min	max	
Цикл записи					
Время цикла	t_{CW}		1000		нс
Время подготовки A до F_W	t_{DA}	U_I = 0,65 до 2,2 В	200		нс
Ширина импульса записи	t_{WW}	t_T = 20 нс	750		нс
Время удержания записи	t_{DW}		50		нс
Время подготовки данных	t_{VI}		800		нс
Время сохранения данных	t_{HI}		100		нс
Время подготовки $\overline{F}_M$ до F_W	t_{VF}		900		нс

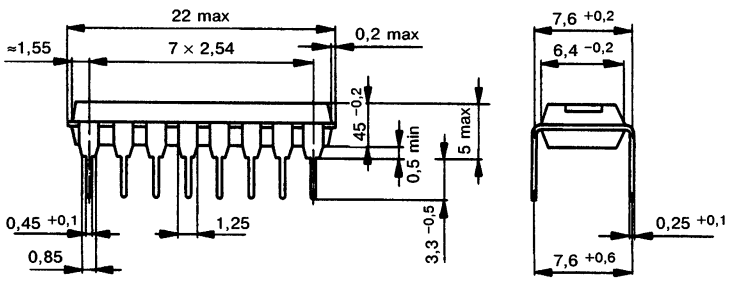
Модуль памяти SAB 2102

Статическое МОП-ОЗУ (RAM) с tri-state выходом (три возможных состояния)

Емкость 1024 бит

- *N*-МОП-технология
- Емкость 1024 × 1 бит
- ТТЛ-совместима, U_S = 5 В
- Выход tri-state, возможность проводного ИЛИ
- Статический режим эксплуатации, нет регенерации данных
- Чтение не меняет содержимого ячеек
- Разнесенный ввод-вывод данных
- Простое расширение с помощью выбора модуля памяти $\overline{F}_M$
- Корпус DIL 16
- Взаимозаменяем с Intersil 7552, Intel, а также AMD 2102 и др.

Чертеж с размерами



Граничные условия

Диапазон рабочих температур	T_U	0 °C до +70 °C
Температура хранения	T_S	−65 °C до +150 °C
Напряжение на каждом выводе по отношению к 0		−0,5 В до +7 В
Максимально допустимая мощность потерь		1 Вт

Емкости

T_U = 25 °C, f = мГц

	Максимум	Условия
Входная емкость C_I	5 пФ	U_I = 0 В (все входы)
Выходная емкость C_Q	10 пФ	U_Q = 0 В

Рис. 12.26. Таблица данных модуля SAB 2102 (Siemens).

Достаточно интересным является ОЗУ GXB 10147 B, которое построено в ЭСЛ-технике. Эта память является статической 128 × 1-битовой памятью с временем доступа 10 нс. Она работает в сто раз быстрее памяти SAB 2102. При напряжении питания −5,2 В эта схема потребляет ток примерно 80 мА, что соответствует рассеиваемой мощности 0,416 Вт. То есть энергопотребление очень высоко. Выдержка из таблицы данных приведена на рис. 12.27.

Статическое ЭСЛ-ОЗУ (RAM) GXB 10147 A

Емкость 128 Бит

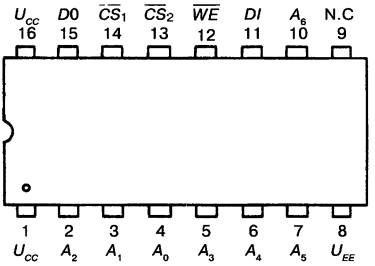
Данные для заказа

Тип	Номер для заказа	Тип корпуса
GXB 10147 A	Q 67000—S38	Рис. 1

- Очень быстрая ЭСЛ-память
- Время доступа 10 нс
- Полностью дешифрован
- Матрица 128 × 1 бит
- Возможность расширения через chip select
- Возможность проводного ИЛИ
- Статический режим эксплуатации, нет регенерации данных
- Стабилизированное потребление тока
- Отрицательный температурный коэффициент токопотребления, самостабилизация
- Металлокерамический корпус с 16 выводами
- Взаимозаменяем с Motorola MCM 10147 AL, Fairchild F 10405
- Совместим с семействами ECL 10 k и Fairchild 95 k

Цоколевка

Вид сверху



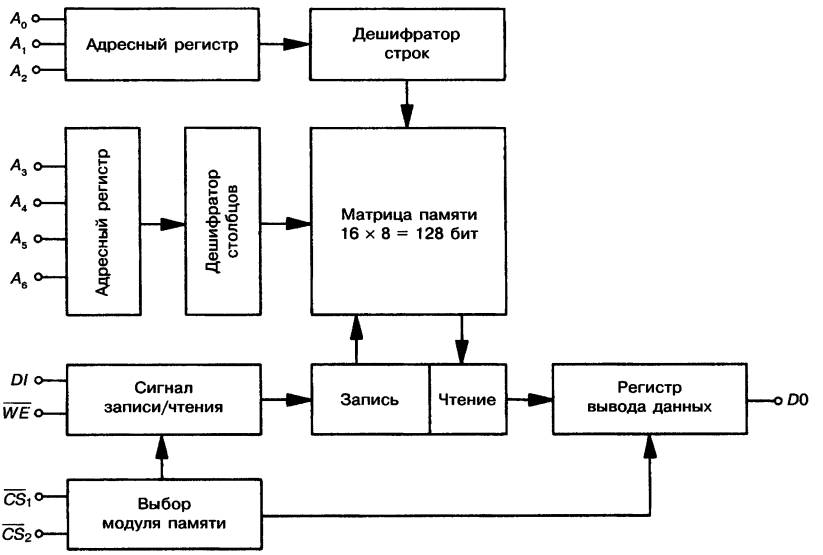
A_0 до A_9 — адресные входы
DI — входы данных
$\overline{CS}_1$, $\overline{CS}_2$ — выбор модуля памяти (chip select)
$\overline{WE}$ — сигнал записи/чтения
$D0$ — ввод данных
U_{EE} — напряжение питания (−5,2 В)
U_{CC} — земля
N.C. — не подключен

Статические параметры

При T_U = 25 °C, U_{EE} = −5,2 В, R_L = 50 Ом по отношению к −2,0 В

Охлаждение воздухом 2,55 м/с

		Условия испытаний	Min	Тип	Max	Ед. измерения
H-напряжение питания	U_{IH}	R_L = 50 Ом по отношению к −2 В; A, DI = $U_{IL\ min}$; $\overline{WE}$ = $U_{IH\ min}$; $\overline{CS}_1$, $\overline{CS}_2$ = $U_{IL\ max}$	−1,105		−0,810	В
L-напряжение питания	U_{IL}		−1,850		−1,475	В
H-выходное напряжение	U_{OH}		−0,960		−0,810	В
L-выходное напряжение	U_{OL}		−1,900		−1,650	В
H-входной ток на A, DI, $\overline{CS}$	I_{IH}				35	мкА
H-входной ток на $\overline{WE}$	I_{IH}				75	мкА
L-входной ток на всех входах	I_{IL}		−6			мкА
Ток питания	I_{EE}			80	100	мА



Время переключения

U_{EE} = 5,2 В ± 10 %

		25 °C			85 °C			Ед. измерения
		min	тип	max	min	тип	max	
Время доступа	t_{ACC}			10	12		14	нс
Время установки сигнала выбора микросхемы	t_E			6,5	8		9	нс
Длительность сигнала записи	t_{WW}	8				8		нс
Задержка перехода выхода данных из активного состояния в состояние отключено в режиме чтения	t_{WR}				8			нс
Время установки *A* до $\overline{WE}$	t_{AWS}	4						нс
Время установки $\overline{CS}$ до $\overline{WE}$	t_{CWS}	1						нс
Время установки *DI* до $\overline{WE}$	t_{IWS}	1						нс
Время удержания $\overline{WE}$ до *A*	t_{WAH}	3						нс
Время удержания $\overline{WE}$ до $\overline{CS}$	t_{WCH}	1						нс
Время удержания $\overline{WE}$ до *DI*	t_{WIH}	1						нс

Граничные условия

Напряжение питания	U_{EE}	−7	В
Входное напряжение	U_I	0 до U_{EE}	В
Выходной ток	I_O	50	мА
Диапазон рабочих температур	T_U	0 до 85	°C
Температура хранения	T_S	−55 до 125	°C

Рис. 12.27. Выдержка из таблицы данных элемента памяти GXB 10147 A (Siemens).

Как пример динамического ОЗУ рассмотрим модуль HYB 4116. Он построен в *N*-МОП-технике и имеет емкость 16384 бит. Каждый бит адресуем индивидуально (16384 × 1-Bit-RAM). Адресные линии переключаются мультиплексором. Структура схемы показана на рис. 12.28.

Динамическое МОП-ОЗУ HYB 4116
Емкость 16384 бит
Предварительные данные

Тип	Номер для заказа	Вид корпуса
HYB 4116 — A 3	Q 67100—Q 186	керамика/рис. 2
HYB 4116 — A 4	Q 67100—Q 187	керамика/рис. 2
HYB 4116 — P 3	Q 67100—Q 219	пластик/рис. 3
HYB 4116 — P 4	Q 67100—Q 220	пластик/рис. 3

HYB 4116 от Siemens — это динамическое ОЗУ в *N*-канальной МОП-технологии с двухслойным поликристаллическим кремнием.

- *N*-канальная МОП-технология
- 16384 × 1 бит, полностью декодирован
- Разделенный ввод—вывод данных
- Все входы ТТЛ-совместимы (включая такт)
- Низкое энергопотребление (462 мВт актив, 20 мВт неактив)
- Промежуточная память адресов и вх. данных
- Время доступа 200 нс, время цикла 375 нс (HYB 4116 — A 3, P 3)
- Время доступа 250 нс, время цикла 410 нс (HYB 4116 — A 4, P 4)
- 3 выходных состояния, 2 ТТЛ-нагрузки
- Может быть заменен MK 4116
- 128 циклов регенерации
- Нет записи выходных данных
- ± 10 % допуск на напряжения питания (все)

Предельные параметры

Макс. наряжение для всех входов и питания U_{DD}, U_{CC} и U_{SS} по отношению к U_{BB}		−0,5 до 20	В
Максимальное наряжение на U_{DD}, U_{CC}, входах по отношению к U_{SS}		−1,0 до 15	В
$U_{BB} - U_{SS}$ ($U_{DD} - U_{SS} > 0$ В)		0	В
Температура окружающей среды в рабочем режиме		0 до 70	°C
Температура хранения	T_U	−65 до 150	°C
Максимально допустимые потери мощности	T_S	1	Вт
Максмальная мощность покоя	P_{tot}	20	мВт

Динамические данные

		HYB 4116				Ед. измерения
		−A 3/−P 3		−A 4/−P 4		
		min	max	min	max	
Время цикла чтения или записи	t_{RC}	375		410		нс
Время цикла чтения—записи	t_{RWC}	375		465		нс
Время цикла чтения—изменения	t_{RMWC}	415		515		нс
Время доступа через $\overline{RAS}$	t_{RAC}		200		250	нс
Время доступа через $\overline{CAS}$	t_{CAC}		135		165	нс

HYB 4116

Принцип действия

Адресация (A_0—A_6)

Для выбора 16384 ячеек памяти требуются всего 14 адресных разрядов, которые последовательно передаются через порты A_0—A_6 двумя тактами (адресное мультиплексирование). Сначала считываются 7 адресных слов и с тактом $\overline{RAS}$ записываются в промежуточную память. В заключение такт $\overline{CAS}$ принимает 7 битов адреса в память выборки битов. При этом необходимо обращать внимание на то, что сигнал адресации к моменту времени отрицательного фронта $\overline{RAS}$ или $\overline{CAS}$ находится в установившемся состоянии.

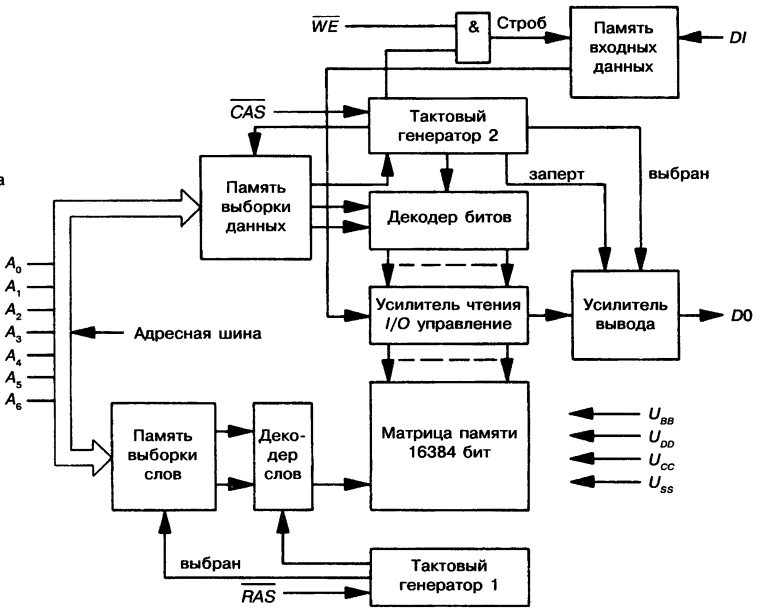
$\overline{RAS}$ и $\overline{CAS}$ определяют момент старта для внутреннего тактирования.

$\overline{RAS}$ вызывает кодировщик слова и активизирует усилитель чтения.

$\overline{CAS}$ управляет кодировщиком битов, а также информационным входом и выводным усилителем.

Цоколевка
Вид сверху

A_0 до A_9 — адресные входы
$\overline{CAS}$ — бит-адресный строб
DI — ввод данных
D0 — вывод данных
$\overline{RAS}$ — строб адресного слова
$\overline{WE}$ — такт записи/чтения
U_{BB} — −5 В
U_{DD} — +12 В
U_{CC} — +5 В
U_{SS} — 0 В



Запись/чтение ($\overline{WE}$)

Цикл запись/чтение запускается, если сигнал $\overline{WE}$ имеет *L*- или *H*-уровень (низкий или высокий уровень сигнала). Информационный вход *DI* заблокирован, если идет процесс чтения.

Самый короткий цикл записи будет в том случае, когда $\overline{WE}$ до или одновременно с $\overline{CAS}$ перейдут на *L* (ранняя запись). Посредством $\overline{CAS}$ записываемые данные передаются в память входных данных.

Замедленная запись, цикл читать/изменить/писать

При замедленной записи или чтении/изменении/письме $\overline{CAS}$ уже стоит на *L*, так что данные записываются со следующим $\overline{WE}$-сигналом в буфер входных данных.

Информационный вход (*DI*)

Данные могут быть введены во время записи и цикла чтения/изменения/записи. Определяющим для времени записи данных является отрицательный фронт $\overline{CAS}$ или $\overline{WE}$ в зависимости от того, какой фронт придет позднее.

Вывод данных (*D0*)

Выход данных может иметь три состояния (Three-State) и предназначен для двух ТТЛ-нагрузок. Выходные данные по отношению к входным данным не инвертируются. Во время цикла чтения данные появляются на входе через соответствующее $\overline{CAS}$ время доступа t_{CAC}. После окончания цикла чтения выход данных посредством $\overline{CAS}$-*H* снова переводится в высокоомное состояние.

Во время цикла чтения/изменения/записи данные подаются на выход, как при цикле чтения. При «ранней записи» выход данных в течение всего цикла высокоомен.

Регенерация

Для сохранения данных в динамической памяти каждая адресная ячейка должна опрашиваться минимум каждые 2 мс. То есть для регенерации всех адресов необходимо в течение каждых 2 мс проводить 128 циклов регенерации. При чтении или записи данные 128 ячеек памяти запрашиваемой линии слов автоматически обновляются.

Цикл зарядки

После подачи напряжения на модуль необходимо подождать несколько циклов, пока не будет обеспечено правильное функционирование. Для этой цели могут быть использованы, например, 8 циклов регенерации.

Условия эксплуатации и испытаний

T_U = 0—70 °C, U_{SS} = 0 В, U_{DD} = +12 В ± 10 %, U_{BB} = −5В ± 10 %, U_{CC} = +5 В ± 10 %

Статические параметры[1]

		Условия испытаний	Min	Тип	Max	Ед. измерения
H-напряжение питания[2] (исключая $\overline{RAS}$, $\overline{CAS}$ и $\overline{WRITE}$)	U_{IH}		2,4		7,0	В
H-напряжение питания[2] $\overline{RAS}$, $\overline{CAS}$ и $\overline{WRITE}$	U_{IHC}		2,7		7,0	В
L-напряжение питания	U_{IL}		−1,0		0,8	В
H-выходное напряжение	U_{OH}	I_O = −5 мА	2,4		U_{CC}	В
L-выходное напряжение	U_{OL}	I_O = 4,2 мА			0,4	В
Токопотребление из U_{DD}[3]	$I_{DD\,1}$				35	мА
Токопотребление покоя из U_{DD}	$I_{DD\,2}$	$\overline{RAS}$ на U_{IH}-уровне $\overline{CAS}$ на U_{IH}-уровне			1,5	мА
Среднее токопотребление из U_{DD} в течение одного цикла регенерации[3]	$I_{DD\,3}$	$\overline{RAS}$ последовательность импульсов $\overline{CAS}$ на U_{IH}-уровне			2,7	мА
Входной ток утечки	$I_{I\,(L)}$		−10		10	мкА
Выходной ток утечки	$I_{O\,(L)}$	$\overline{CAS}$ на U_{IH}-уровне $U_O = U_{SS}$ до U_{CC}	−10		10	мкА
Токопотребление покоя из U_{CC}	I_{CC}	$\overline{RAS}$ на U_{IH}-уровне $\overline{CAS}$ на U_{IH}-уровне	−10		10	мкА
Среднее токопотребление из U_{BB}	$I_{BB\,1}$				200	мкА
Токопотребление покоя из U_{BB}	$I_{BB\,2}$				100	мкА
Емкости						
Входная емкость[4] (A_0—A_6), *DI*	$C_{I\,1}$				5	пФ
Входная емкость[4] $\overline{RAS}$, $\overline{CAS}$ и $\overline{WRITE}$	$C_{I\,2}$				10	пФ
Выходная емкость	C_O	*D*0 — три состояния			7	пФ

Примечания:

[1] При подаче напряжения питания проверить, что U_{DD}, U_{CC} и U_{SS} больше, чем U_{BB} = −0,3 В.

[2] Повышение входных сигналов до уровня 6,5 В или понижение до −2 В длительностью не более 0 нс не приводит к выводу модуля из строя.

[3] I_{DD} зависит от продолжительности рабочего цикла. Максимальный ток измерен при минимальном времени цикла.

[4] Эффективная емкость определяется из уравнения:

$$C = \frac{I \cdot \Delta t}{\Delta U} \quad \text{при } \Delta U = 3 \text{ В}.$$

Рис. 12.28. Выдержка из таблицы данных динамического ОЗУ HYB 4116.

12.4. Постоянные запоминающие устройства (ПЗУ, ROM)

ПЗУ (ROM) содержат нестираемую и неизменяемую информацию. Понятие ROM является сокращением от Read Only Memory.

ПЗУ можно сравнить с книгой. Занесенная в нее информация является доступной в любое время. Однако невозможно заменить или изменить информацию. В ПЗУ сохраняют часто используемые сведения, например команды, программы и таблицы. По мере необходимости могут быть считаны отдельные ячейки таблицы.

> Для организации ПЗУ применяют два типа запоминающих элементов. Запоминающие элементы первого типа должны всегда содержать значение 1. Запоминающие элементы второго типа должны всегда содержать значение 0.

Представление данных в памяти и организация памяти ПЗУ похожи на ОЗУ. Запоминающая матрица состоит из строк и столбцов. Отдельные ячейки памяти выбираются адресацией (рис. 12.29).

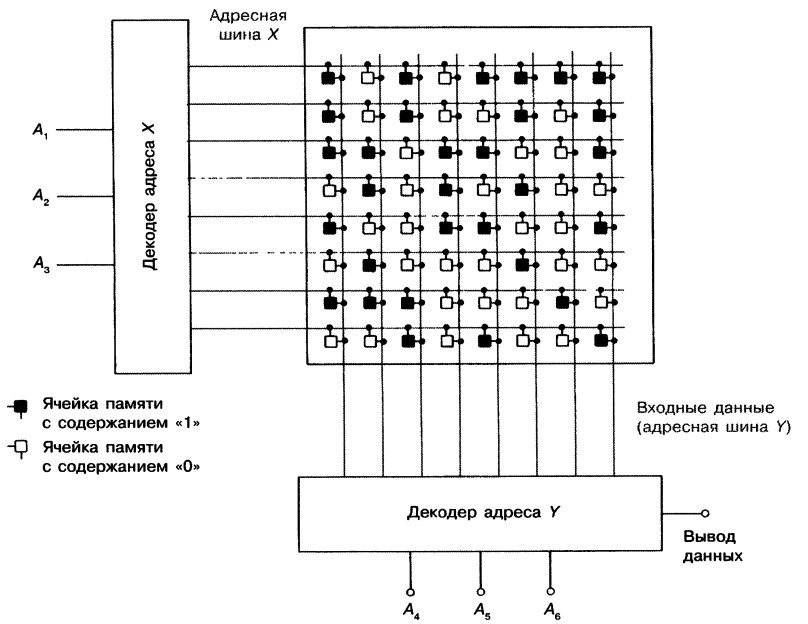


Рис. 12.29. Схема 64 × 1 битового ПЗУ.

ПЗУ производятся чаще всего в *N*-МОП-технике. Возможная степень интеграции велика, а энергопотребление незначительно. Как построен запоминающий элемент, который всегда содержит значение 1? Это можно сделать с помощью отсутствующего *N*-МОП-транзистора. Запоминающий элемент, который всегда имеет значение 0, образуется *N*-МОП-транзистором (рис. 12.30).

Y-адресные шины являются одновременно линиями данных. Если запоминающий элемент считывается, на *Y*-адресную шину прикладывается 1-сигнал.

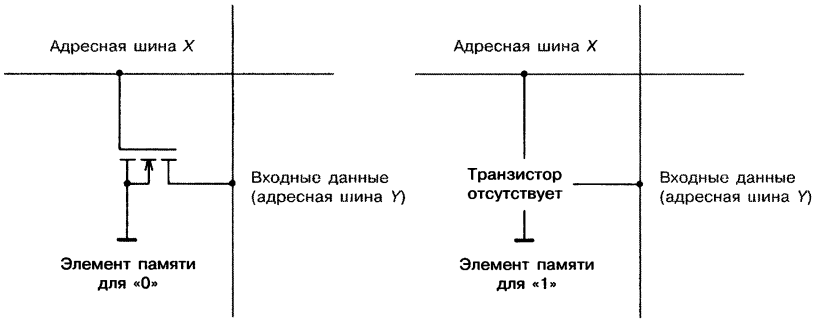


Рис. 12.30. Элемент памяти ПЗУ (*N*-МОП-технология).

Если выбранный запоминающий элемент является 0-элементом, то на линии данных действует 0-сигнал, транзистор открыт и связывает линию данных с землей.

Если выбранный запоминающий элемент является 1-элементом, то на линии данных действует 1-сигнал. На линии данных не может быть 0-сигнал, так как транзистор отсутствует.

Постоянные запоминающие устройства этого вида называются также масочными ПЗУ. Информация записывается при изготовлении. Покрытием (маскированием) на определенных участках предотвращается изготовление полевых транзисторов. Пользователь должен указать заранее, какая информация должна быть записана и соответственно где должны и где не должны быть размещены полевые транзисторы.

Изготовление ПЗУ является экономически обоснованным только при массовом производстве, так как для каждого информационного наполнения должна быть изготовлена соответствующая маска.

На рис. 12.31 приведена сокращенная таблица данных постоянного запоминающего устройства SAB 8316. Это ПЗУ построено в *N*-МОП-технике и содержит 2048 слов по 8 бит, организованных в виде 128 строк и 16 столбцов. Для 128 строк требуются 7 адресных линий. 16 столбцов управляются по 4 адресным линиям. Всего в наличии имеются 11 адресных входов.

Статическое ПЗУ, 16384 бит SAB 8316 A

Структура 2048 × 8 бит
Время доступа макс. 850 нс
Напряжение питания 5 В
ТТЛ-совместим по всем входам-выходам
Программируемый модуль выборки входов для легкого расширения модуля памяти
Три состояния выхода — возможность подсоединения проводного ИЛИ
Полностью дешифрован — адресная дешифровка всего модуля
Все входы защищены против статического разряда
SAB 8316 — статическое ПЗУ со структурой 2048 слов по 8 бит

Входы и выходы полностью ТТЛ-совместимы. Этот модуль использует только одно значение напряжения питания — +5 В. Три входа выборки являются программируемыми. Каждая комбинация *L*- и *H*-уровней активных модулей выборки входов может быть распознана таким образом, что желаемый код выборки модуля в течение процесса масочного программирования может быть зафиксирован.

Эти три программируемых модуля выборки входов, а также возможность проводного ИЛИ облегчают расширение устройства памяти.

SAB 8316 изготовлен в КМОП-технологии.

Статические параметры и условия эксплуатации

T_U = 0 до +70 °C, V_{CC} = 5 В ± 5 % (если не указаны другие данные)

Символ	Обозначение	Предельные значения			Ед. измерения	Условия испытаний
		min	тип[2]	max		
I_{LI}	*L*-входной ток	—	—	10	мкА	V_{IN} = 0 до 5,25 В
I_{LOH}	*H*-выходной ток	—	—	10		CS = 2,2 В, $V_{вых}$ = 4 В
I_{LOL}	*L*-выходной ток	—	—	−20		CS = 2,2 В, $V_{вых}$ = 0,45 В
I_{CC}	Токопотребление	—	40	98	мА	Все входы 5,25 В, выход данных открыт
V_{IL}	*L*-входное напряжение	−0,5	—	0,8	В	—
V_{IH}	*H*-входное напряжение	2	—	V_{CC} + 1 В		
V_{OL}	*L*-выходное напряжение	—	—	0,45		I_{OL} = 2 мА
V_{OH}	*H*-выходное напряжение	2,2	—	—		I_{OH} = −100 мкА

Время переключения

T_U = 0 до +70 °C, V_{CC} = +5 В ± 5 % (если не указаны другие данные)

Символ	Обозначение	Предельные значения			Ед. измерения
		min	тип[1]	max	
t_A	Задержка от адреса к выходу	—	400	850	нс
t_{CO}	Задержка выбора микросхемы в режиме чтения	—	—	300	
t_{DF}	Задержка вывода данных	0	—	300	

Условия испытаний

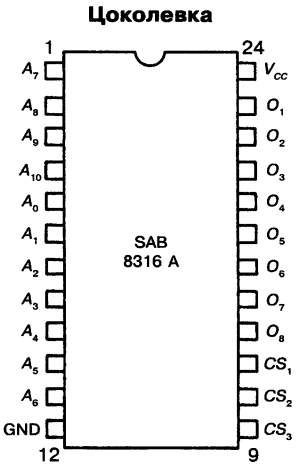
Уровень импульсов на входе	0,8 до 2 В
Передний и задний фронт на входе (от 10 до 90 %)	20 нс
Нагрузка на выходе 1	ТТЛ-нагрузка и C_L = 100 пФ
Уровни проведения измерений:	
для входов	1,5 В
для выходов	0,45 до 2,2 В

Емкость — выборочный контроль

при T_U = 25 °C, f = 1 мГц

Символ	Обозначение	Предельные значения		Ед. измерения
		тип[1]	max	
$C_{вх}$	Входная емкость[2]	4	10	пФ
$C_{вых}$	Выходная емкость[2]	8	15	

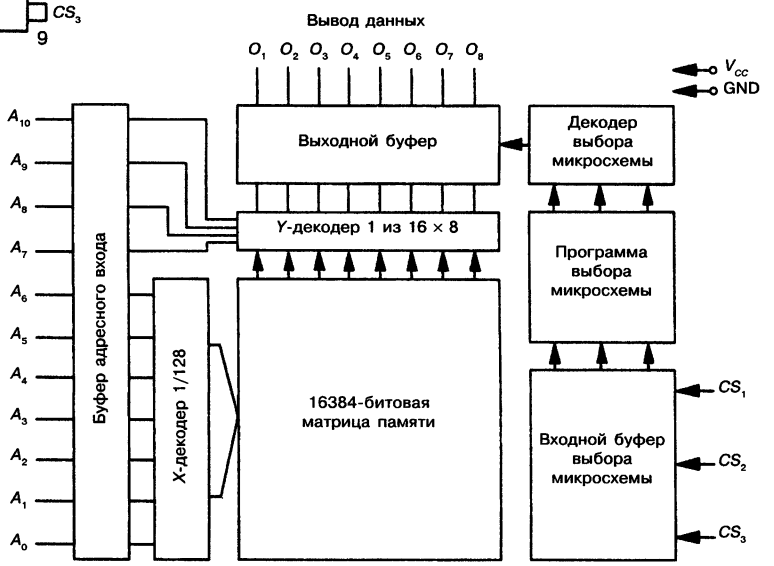
Примечание:
[1] Типовые значения для 25 °C и номинального напряжения питания.
[2] Все выводы, кроме вывода, на котором проводится измерение и который связан с землей.



Обозначения вводов

A_0—A_{10} — адресные входы
O_1—O_8 — вывод данных
CS_1—CS_3 — сигнал выбора микросхемы
V_{CC} — напряжение питания(+5 В)
GND — заземление (0 В)

Блок-схема



Предельные значения[1]

Температура окружающей среды в рабочем режиме	0 до + 70 °C
Температура хранения	−65 до + 150 °C
Напряжение на каждом выводе относительно земли	−0,5 до + 7 В
Потребление мощности	1 Вт

Примечание:
[1] Приведенные данные являются граничными, при их превышении модуль может выйти из строя.
[2] Типичные значения при T_U = 25 °C и номинальном напряжении.

Рис. 12.31. Данные на статическое ПЗУ SAB 8316 A (Siemens).

Интересным является незначительное потребление энергии — 31,4 мкВт на бит и относительно малое время доступа. В этой микросхеме дополнительно содержится модуль схемы выборки. Буфер вывода доступен только при условии, что на входах CS_1, CS_2 и CS_3 действуют определенные бинарные сигналы. Таким образом, облегчается объединение нескольких таких ПЗУ в устройство памяти большего объема.

12.5. Программируемое постоянное запоминающее устройство (ППЗУ, PROM)

Сокращение PROM происходит от Programmable Read Only Memory — программируемое ПЗУ.

Развитие программируемых постоянных запоминающих устройств явилось следствием пожелания пользователей самим вводить данные в постоянное запоминающее устройство. Также они не хотели быть связанными большим количеством экземпляров. Целью было экономически выгодное производство малых партий, до одной штуки.

Представим себе ПЗУ, которое состоит только из элементов памяти для записи 0 согласно рис. 12.30. То есть на перекрестиях находятся полевые транзисторы. Если один из транзисторов будет прожжен, то в этом месте будет сохранена 1. Почему нельзя всегда целенаправленно прожигать транзисторы в тех местах, в которые нужно записать 1?

Таким способом программируется ППЗУ, т. е. производится запись данных. Существуют различные разновидности ППЗУ. Биполярные ППЗУ с диодами и транзисторами в точках пересечения адресных линий имеют в настоящее время большое распространение. На рис. 12.32 представлена структура 8 × 8-битового диодного ППЗУ. Диоды имеют очень тонкие выводы из хромоникелевого сплава (от 20 до 30 нм в ширину, 100 нм толщиной). Если ток превышает определенное значение, то эти выводы перегорают. Для программирования ППЗУ требуется особенное устройство, назы-

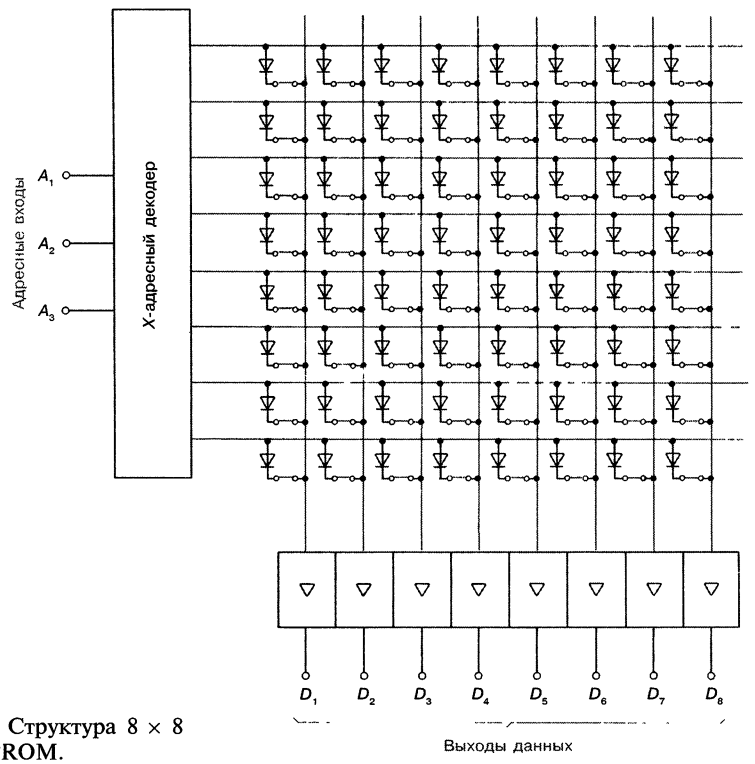


Рис. 12.32. Структура 8 × 8 диодного PROM.

ваемое программатор. Программирование ППЗУ необратимо. В случае ошибки ППЗУ можно выбрасывать. Коррекция возможна только в редких случаях, если можно дополнительно прожечь оставшиеся соединения.

12.6. Перепрограммируемые постоянные запоминающие устройства

Стираемые и программируемые постоянные запоминающие устройства позволяют удалять введенную информацию и перепрограммировать ПЗУ.

Удаление и перепрограммирование может повторяться как угодно часто без повреждения модуля памяти.

Различают две группы перепрограммируемых постоянных запоминающих устройств. В одной группе информация удаляется ультрафиолетовым светом. Постоянные запоминающие устройства этой разновидности называются СППЗУ — стираемое программируемое постоянное запоминающее устройство (EPROM — Erasable Programmable Read Only Memory или REPROM — Re-programmable Read Only Memory).

Стираемые программируемые постоянные запоминающие устройства второй группы перепрограммируются электрическим напряжением. Для них принято сокращение EEROM (Electrically Erasable Read Only Memory = электрически стираемые постоянные запоминающие устройства) и EAROM (Electrically Alterable Read Only Memory — электрически перепрограммируемые постоянные запоминающие устройства).

12.6.1. EPROM и REPROM

EPROM и REPROM незначительно отличаются друг от друга, в основном технологией изготовления. Они идентичны по структуре и принципу действия и поэтому могут рассматриваться вместе. СППЗУ (стираемое программируемое постоянное запоминающее устройство) и REPROM-запоминающий элемент для 1-го бита состоят из двух полевых транзисторов, в основном применяются N-МОП полевые транзисторы. Структура типичного элемента памяти представлена на рис. 12.33. Транзистор T_1 является транзистором выборки, транзистор T_2 — транзистором памяти.

Затвор транзистора памяти T_2 окружен материалом с высоким сопротивлением. Он ни к чему не присоединен. Такой затвор называется плавающим (англ. floating-gate). В стертом состоянии плавающий затвор не заряжен. Транзистор T_2 заперт. Если на адресные шины X и Y помещают +5 В, то транзистор T_1 откроется. Однако транзистор T_2 заперт, так что Y-шина, которая одновременно является шиной данных, не может сброситься на $L \triangleq 0$. Y-шина остается на уровне 1. В стертой памяти типа EPROM и REPROM все элементы памяти имеют содержание H $\triangleq$ 1.

При вводе данных нужные элементы перепрограммируются на 0. Говорят, что программируются «нули».

Запоминающий элемент имеет содержание памяти 0, если транзистор памяти открыт.

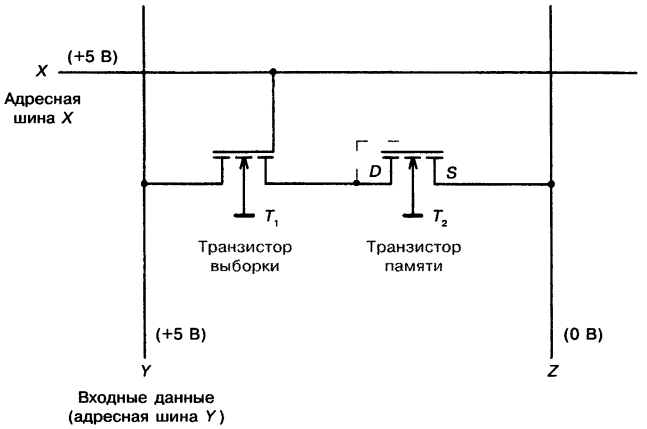


Рис. 12.33. Элемент памяти EPROM-REPROM.

Если адресуется элемент с открытым транзистором памяти T_2, т. е. на X-шину и Y-шину подано +5 В, то T_1 также открывается. Так как на шине Z действует 0 В, то шина Y разряжается до уровня приблизительно 0 В. Как добиться, чтобы транзистор памяти открылся? Нужно зарядить его затвор.

> Плавающий затвор N-МОП-транзистора должен быть заряжен положительно по отношению к подложке, образуя n-проводящий мостик между истоком (Source) и стоком (Drain).

Рассмотрим структуру транзистора памяти (рис. 12.34). Между D и подложкой прикладывается относительно высокое напряжение (+27 В). Так как плавающий затвор и изоляционный слой очень тонкие, то возникает очень сильное электрическое поле. Под влиянием этого сильного поля электроны от плавающего затвора перемещаются к стоку (против линий поля). Изоляционный слой пропускает электроны из-за очень высокой электрической напряженности поля. Можно сказать, что изоляционный слой кратковременно проламывается. На самом деле причиной является туннельный эффект. Этот процесс называется Floating Gate — avalanche-injection (англ. — лавинная зарядка плавающего затвора). МОП-полевой транзистор, заряжающийся по этому принципу, называется FAMOS-транзистор.

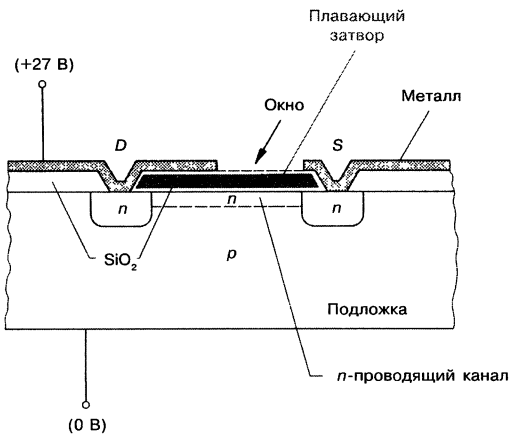


Рис. 12.34. Структура транзистора с плавающим затвором (N-канальный).

Напряжение +27 В называется **напряжением программирования**. После кратковременного действия этого напряжения затвор заряжается. Материал, окружающий плавающий затвор, является высокоомным. Электриче-ский заряд на нем сохраняется. На подложке под плавающим затвором возникает *n*-проводящий мостик. Полевой транзистор низкоомен между *S* и *D*.

Ячейки памяти СППЗУ (стираемое программируемое постоянное запоминающее устройство) или REPROM программируются по очереди после выбора адресными шинами *X* и *Y* (см. рис. 12.33). К *X* и *Y* прикладываются напряжения выборки +5 В. Вследствие этого открывается T_1. Напряжение *Y*-шины кратковременно повышается до +27 В. Программирование может неоднократно повторяться по соображениям надежности. По данным производителя, заряд на плавающем затворе сохраняется в течение многих лет, то есть данные могут сохраняться от 1 года до 100 лет.

Запрограммированные СППЗУ и REPROM сохраняют введенную информацию.

Известные производители дают гарантию от 10 лет на сохранность данных.

Для удаления информации из СППЗУ или REPROM стираемый участок через окно над плавающим затвором облучается ультрафиолетом.

Высокоомный материал ионизируется облучением и начинает проводить. Затвор медленно разряжается. При мощности излучения примерно 10 Вт · с/см^2 затвор разряжается за 20—30 минут. Корпус СППЗУ и REPROM имеет окно, проходящее над всей поверхностью кристаллического чипа (рис. 12.35). Ультрафиолетовый свет облучает все элементы памяти и стирает их все одновременно.

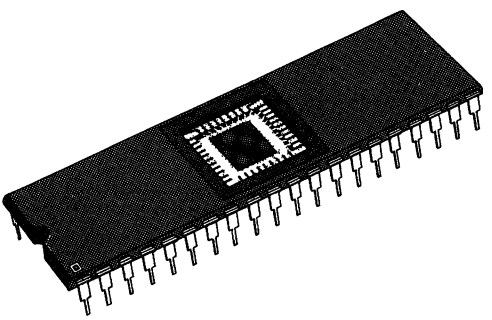


Рис. 12.35. Корпус EPROM-REPROM.

При стирании из СППЗУ и REPROM удаляется вся информация.

После удаления информации модуль должен остыть, так как он заметно нагревается. Прежде всего должна уменьшиться ионизация в изолирующем материале. Материал должен снова стать высокоомным. Только тогда можно снова начинать цикл программирования. Время охлаждения должно составлять по меньшей мере половину от времени программирования, лучше всего час.

Находящиеся на свету СППЗУ и REPROM могут быть случайно стерты.

Лучи солнечного света приводят к стиранию микросхемы в течение примерно 3 дней. Свет люминесцентной лампы удаляет информацию примерно через 3 недели. Чтобы предотвратить случайное стирание, целесообразно заклеить окно темным скотчем.

Процесс стирания не меняет свойства материалов модуля, так что возможно сколь угодно частое удаление и перепрограммирование. Стираемые программируемые постоянные запоминающие устройства и REPROM выпускаются с объемами памяти от 100 бит до 16 Кбит. Модули с 32 Кбит и 64 Кбит находятся в разработке. Часто применяемое устройство REPROM выпускается в виде модуля SAB 8708. На рис. 12.36 воспроизведена сокращенная таблица данных этого модуля.

Стираемое и программируемое устройство памяти ППУ SAB 8708

SAB 8708: Структура 1024 × 8

Быстрое программирование — для всех 8 кбит, 100 с (контрольная величина)
Низкое энергопотребление
Время доступа макс. 450 нс
Напряжение питания ±5 В, ±12 В
Статическая схема, не нужен цикл регенерации
ТТЛ-совместимо по всем входам-выходам во время чтения и программирования
Три состояния выходов, возможность проводного ИЛИ

SAB 8708 — это быстрое 8192-битовое стираемое и электрически программируемое ППУ (REPROM). Особенно хорошо подходит для разработки различных схем и систем.

Производится в DIP-корпусе с 24 выводами и кварцевой крышкой, поэтому у пользователя есть возможность с помощью ультрафиолетового облучения стирать данные и проводить новое электрическое программирование.

Благодаря совместимости портов SAB 8708 и SAB 8308 становится возможным обновление компьютерной системы при использовании возможностей изменения SAB 8708. При этом развитие возможно без изменения проводных линий, что является благоприятным с экономической точки зрения, особенно при массовом производстве.

Время переключения

T_U = 0—70 °C, V_{SS} = +5 В ± 5 %, V_{DD} = +12 В ± 5 %, V_{BB} = −5 В ± 5 %, V_{CC} = 0 В (если не указаны другие данные)

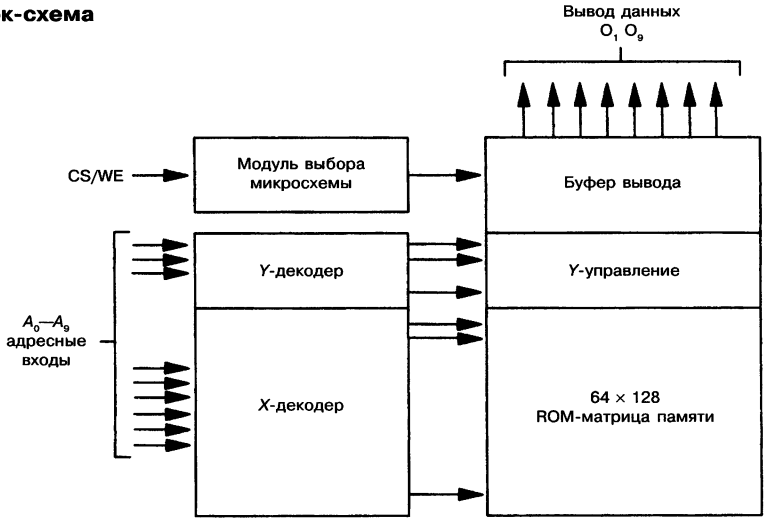
Символ	Обозначение	Предельные значения			Ед. измерения
		min	тип	max	
t_{ACC}	Между началом адресного сигнала и началом сигнала данных	—	280	450	нс
t_{CO}	Между началом сигнала выборки и сигналом данных	—	—	120	
t_{DF}	Между окончанием сигнала выборки и сигналом данных	0	—	120	
t_{OH}	Между окончанием адресного сигнала и сигнала данных	0	—	—	

Емкость — выборочный контроль

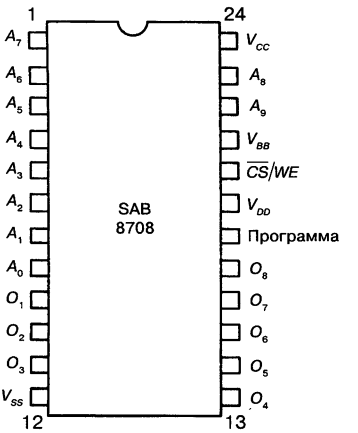
при T_U = 25 °C, f = 1 МГц

Символ	Обозначение	Предельные значения		Единица измерения	Условия испытаний
		тип	max		
$C_{вх}$	Входная емкость	4	6	пФ	$V_{вх}$ = 0 В
$C_{вых}$	Выходная емкость	8	12		$V_{вых}$ = 0 В

Блок-схема



Цоколевка



Обозначения вводов

A_0—A_9 — адресные входы
O_1—O_8 — вывод данных
$\overline{CS}/WE$ — сигнал выборки микросхемы

Предельные значения[1]

Температура окружающей среды в рабочем режиме	−25 до +85 °C
Температура хранения	−65 до +125 °C
Напряжение на каждом входе и выходе по отношению к V_{BB} (кроме программирующего)	+15 до −0,3 В
Программируемый вход на V_{BB}	+35 до −0,3 В
Напряжение питания V_{CC} и V_{SS} по отношению к V_{BB}	+15 до −0,3 В
V_{DD} по отношению к V_{BB}	+20 до −0,3 В
Потребление мощности	1,5 Вт

Программирование

В незапрограммированном модуле, а также после каждого стирания все биты находятся в состоянии «1» (являются выходом High). При записи данных нужные биты становятся нулями.

Для подготовки схемы к программированию на вход $\overline{CS}/WE$ (порт 20) подается +12 В. Адресное слово выбирается так же, как режим чтения. Входные данные поступают на 8-битовый параллельный ввод по шине данных (O_1—O_8). Логический уровень и напряжения питания для адресной шины и шины данных такие же, как и для режима работы «чтение». После того,

как установлены адреса и данные, на порт программирования 18 прикладывается импульс напряжения (V_p) на каждый адрес. Однократное прохождение всех программируемых адресов называется циклом программирования. Количество необходимых циклов N определяется в зависимости от длительности импульса программирования (t_{PW}) по формуле $N \times t_{PW} = 100$ мс. Для контроля программирования циклы программирования и чтения могут быть изменены, как показано на импульсной диаграмме.

Чтение

Статические параметры и условия эксплуатации

T_U = 0—70 °C, V_{SS} = +5 В ± 5 %, V_{DD} = +12 В ± 5 %, V_{BB} = −5 В ± 5 %, V_{CC} = 0 В (если не указаны другие данные)

Символ	Обозначение	Предельные значения			Ед. измерения	Условия испытаний
		min	тип[2]	max		
I_{LI}	Входной ток выборки адреса и микросхемы	—	—	10	мкА	$V_{вх}$ = 5,25 В
I_{LO}	Остаточный выходной ток	—	—	10		$V_{вых}$ = 5,25 В $\overline{CS}/WE$ = 5 В
I_{DD}	Токопотребление на V_{DD}	—	50	65	мА	Токопотребление в худшем случае
I_{CC}	Токопотребление на V_{CC}	—	6	10		Все входы на H-уровне
I_{BB}	Токопотребление на V_{BB}	—	30	45		$\overline{CS}/WE$ = 5 В T_U = 0 °C
V_{IL}	L-входное напряжение	V_{SS}	—	0,65	В	—
V_{IH}	H-входное напряжение	3	—	V_{CC} + 1		
V_{OL}	L-выходное напряжение	—	—	0,45		I_{OL} = 1,6 мА
V_{OH1}	H-выходное напряжение	3,7	—	—		I_{OH} = −100 мкА
V_{OH2}	H-выходное напряжение	2,4	—	—		I_{OH} = −1 мА
P_D	Потребляемая мощность	—	—	800	мВ	T_U = 70 °C

Примечание:
[1] При превышении граничных данных модуль будет выведен из строя.
[2] Типовые значения при T_U = 25 °C и номинальном напряжении питания.
[3] Вход «ввод программ» (Pin 18) может быть соединен с V_{SS} или V_{CC} в течение цикла чтения.

Импульсная диаграмма

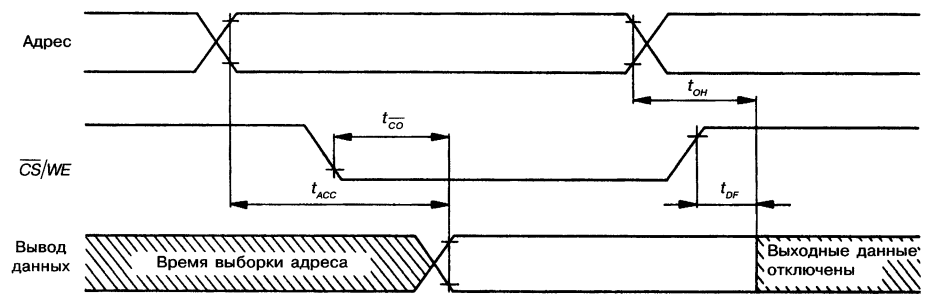


Рис. 12.36. Таблица данных стираемого и программируемого ППУ SAB 8708.

12.6.2. Постоянные запоминающие устройства EEROM (ЭСППЗУ — электрически стираемое программируемое постоянное запоминающее устройство) и EAROM

Запоминающие устройства типа EEROM и EAROM являются, как сказано в предыдущем разделе, постоянными запоминающими устройствами. Их можно стирать и перепрограммировать. Стирание и программирование можно делать часто, например 10 000 раз. Важное различие, тем не менее, существует:

> Постоянные запоминающие устройства типа EEROM и EAROM стираются электрически.

Ячейка памяти построена на двух самозапирающихся MOSFET-транзисторах *n*-канального типа. Структура ячейки памяти соответствует схеме на рис. 12.33. Транзистор T_1 работает как транзистор выборки. Транзистор T_2 является транзистором памяти. В качестве транзистора памяти используется FAMOS-транзистор с плавающим затвором (рис. 12.37).

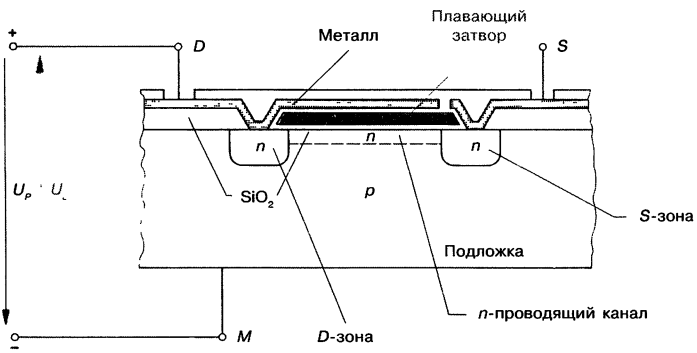


Рис. 12.37. Транзистор памяти с плавающим затвором, стирается электрически.

Программирование происходит так же, как в СППЗУ и REPROM. Металлический сток (*D*) получает положительное напряжение U_p относительно подложки (например +40 В). В очень сильном электрическом поле происходит перемещение электронов от плавающего затвора к стоку (Drain). Плавающий затвор теряет электроны и заряжается вследствие этого положительно. После снятия напряжения программирования U_p остается электрическое поле между затвором и подложкой. В верхней зоне подложки образуется проводящий мост. Транзистор между *S* и *D* низкоомен, т. е. открыт (содержание памяти 0).

Для стирания напряжение между стоком (*D*) и подложкой (*M*) меняет знак. Напряжение стирания U_L возбуждает электрическое поле противоположной направленности. Под действием поля электроны перемещаются от металлического вывода затвора на плавающий затвор и разряжают его. После полной разрядки происходит перезарядка на отрицательный заряд. После снятия напряжения стирания исчезает электрическое поле, направленное от подложки к затвору. *N*-проводящий мост между *D*-зоной и *S*-зоной исчезает. Транзистор запирается (содержание памяти 1).

Электрически стираемые постоянные запоминающие устройства могут строиться таким образом, что вся информация модуля удаляется одновременно. Предложено, чтобы модули с одновременным стиранием информации получили обозначение EEROM.

Можно также построить постоянные запоминающие устройства таким образом, чтобы каждый элемент памяти стирался индивидуально. Такая память позволяет побитовое перепрограммирование. Для памяти этого вида используется обозначение EAROM (Electrically Alterable ROM — электрически программируемое постоянное запоминающее устройство).

Время перепрограммирования составляет от 20 мс до 100 мс.

12.7. Магнитные запоминающие устройства

Магнитные запоминающие устройства являются элементами памяти, которые построены на постоянных магнитах в форме кольца. Каждое магнитное кольцо сохраняет 1 бит. Состояние намагничивания сохраняется при снятии напряжения питания. Магнитное запоминающее устройство является энергонезависимым запоминающим устройством. Время доступа к занесенным в память данным незначительно (около 0,5 мкс). Изготовление является, тем не менее, очень дорогостоящим. Поэтому магнитные кольца являются очень дорогой памятью. Они использовались в большом объеме как быстрая оперативная память в компьютерной технике, однако вытесняются полупроводниковыми запоминающими устройствами.

12.7.1. Магнитные кольца

Магнитные кольца — это кольцеобразные ферритовые сердечники с внешним диаметром от 0,46 мм до 0,8 мм. Много лет назад использовались кольца с внешним диаметром до 4 мм. Феррит — это специальный твердый магнитный материал с почти прямоугольной петлей гистерезиса. Феррит имеет большую намагниченность насыщения и среднюю коэрцитивную силу (рис. 12.38).

Из-за прямоугольной формы петли гистерезиса материал также называется ферритовым прямоугольным. Из-за почти вертикального спуска и подъема петли гистерезиса магнитное запоминающее кольцо может принимать только два стабильных состояния намагничивания. Намагничивание действует либо в области положительной насыщенности, либо в области отрицательной насыщенности (см. рис. 12.38).

> Магнитное кольцо может находиться только в двух различных стабильных состояниях намагничивания.

Состоянию намагничивания в положительной области насыщения ставится в соответствие бинарное состояние 1. Состоянию намагничивания в отрицательной области насыщения ставится в соответствие бинарное состояние 0 (рис. 12.39). Бинарные состояния называются также логическими состояниями.

> С помощью импульсов тока магнитные кольца переключаются из одного состояния в другое.

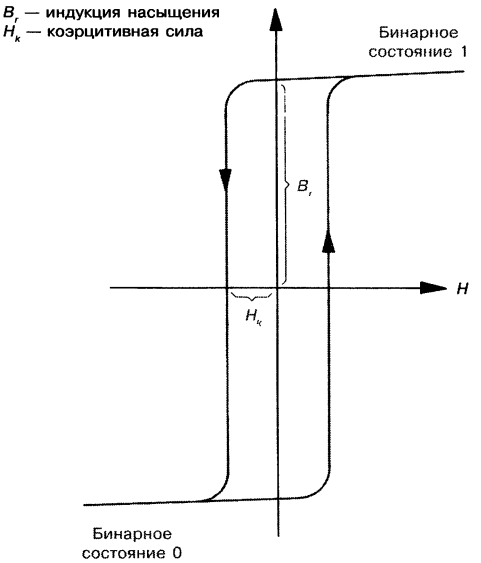


Рис. 12.38. Петля гистерезиса кольца памяти.

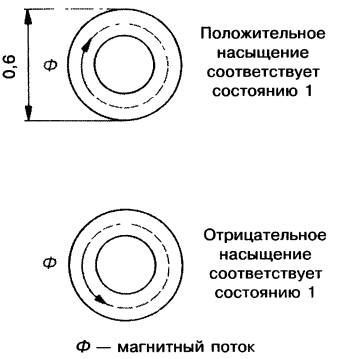


Рис. 12.39. Организация бинарных состояний магнитного кольца.

Импульсы тока производят необходимую для переключения напряженность магнитного поля. Смена состояния происходит примерно за 200 нс. Это время называется **временем переключения**.

12.7.2. Матрица магнитных колец памяти

Структура матрицы представлена на рис. 12.40. Каждый элемент подключен к *X*-адресной шине, *Y*-адресной шине и разрядной шине чтения. Матрица на рис. 12.40 имеет объем памяти 16 бит, организованной в 4 × 4 бита.

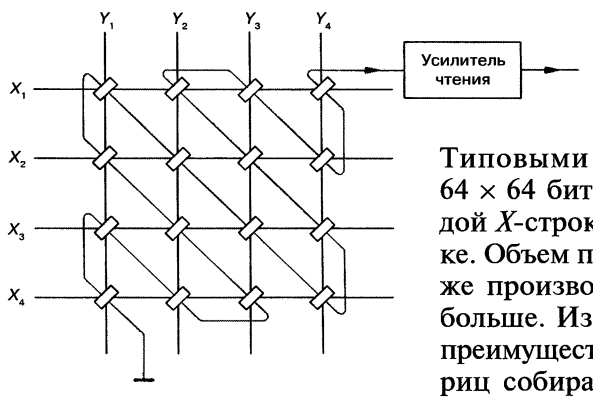


Рис. 12.40. Строение матрицы памяти на магнитных кольцах.

Типовыми являются матрицы объемом 64 × 64 бит. Они имеют 64 кольца в каждой *X*-строке и 64 кольца в каждой *Y*-строке. Объем памяти составляет 4096 бит. Также производятся матрицы 128 × 64 бит и больше. Изготовление матриц происходит преимущественно вручную. Несколько матриц собираются в большой блок магнитной памяти.

12.7.3. Чтение и запись

Запись данных в оперативную память может происходить разными способами. Самый простой — так называемый метод половинного тока. Рассмотрим рис. 12.41. Все элементы матрицы должны быть в состоянии 0. То есть матрица не содержит никакой информации.

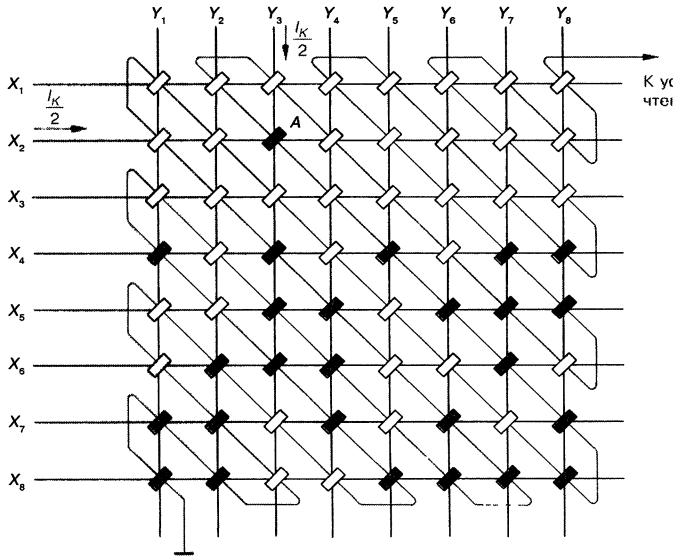


Рис. 12.41. 8 × 8-битовая матрица памяти на магнитных кольцах.

Для записи данных в оперативную память требуется переключить определенные элементы в состояние 1, а остальные оставить в состоянии 0.

Для переключения одного кольца (элемента магнитной памяти) необходим ток 300 мА, значит через элемент должен протекать ток 300 мА в нужном направлении, для того чтобы элемент переключился в состояние 1. Если необходимо переключить определенный элемент в состояние 1, то по каждой координатной шине должен течь ток 150 мА. Если, например, 3-й элемент 2-й строки надо переключить в состояние 1, то по координатным шинам X_2 и Y_3 должен протекать ток 150 мА. Всего через элемент протечет 300 мА, и он переключится. Для надежности по адресным шинам передают ток немного больший, чем половина номинального, в среднем 160 мА. Ко всем другим элементам магнитной памяти, находящимся на координатных шинах X_2 и Y_3, кроме выбранного элемента, будет подведен ток 160 мА. Этого тока недостаточно для переключения. Следовательно, другие элементы памяти переключаться не будут.

По методу половинного тока нужные элементы по очереди переключаются в состояние 1. Для управления током требуется особая схема.

Считывание информации называется **чтением**. При чтении требуется выяснить, какие элементы находятся в состоянии 1 и какие — в состоянии 0. Элементы «опрашиваются» по очереди. Адресные шины посылают полуток в противоположном направлении, чем при записи. Если элемент находится в состоянии 0, он будет этими токами несколько более отрицательно насыщен. Его магнитное поле не изменится. Если элемент находится в состоянии 1, он переключается в состояние 0. Поле инвертируется. В процессе чтения индуцируется импульс напряжения. Импульс напряжения усиливается в усилителе чтения и обрабатывается дальше.

При чтении все элементы, которые находятся в состоянии 1, переключаются в состояние 0. Вследствие этого информация удаляется.

Удаление информации при чтении является существенным недостатком. Если информация нужна и позже, она должна быть сохранена в промежуточном буфере и после чтения обновлена, т. е. снова записана.

12.8. Память на магнитных доменах

Запоминание методом магнитных доменов позволяет сохранять большие объемы данных в небольшом пространстве с малым временем доступа. Данные доступны относительно быстро. Этот вид памяти энергонезависим, т. е. она сохраняется при снятии напряжения питания.

12.8.1. Цилиндрические магнитные домены

В определенных магнитных слоях, таких, как Gadolinium-Gallium-Granat ($Cd_3Ga_5O_{12}$). можно производить малые зоны, которые имеют намагниченность с обратным знаком по отношению к слою, в котором они находятся (рис. 12.42). Эти малые зоны называются **цилиндрическими магнитными доменами**. Они могут производиться с различными диаметрами. Типовые диаметры находятся в пределах между 1,5 и 4 мкм.

> Цилиндрические магнитные домены являются малыми зонами, которые намагничены всегда противоположно окружающему слою.

Они принимают форму цилиндров, если внешнее поле вертикально, например в поле постоянного магнита, полюса которого расположены выше и ниже магнитного слоя.

Цилиндрические магнитные домены производятся в генераторе цилиндрических магнитных доменов. Это электромагнит, при протекании тока через который отдельные зоны кристалла намагничиваются со знаком, противоположным полю окружающей среды.

Если поместить поблизости от этих цилиндрических магнитных доменов постоянный магнит, они начнут перемещаться в слое в зависимости от полярности магнита либо к нему, либо от него.

> Цилиндрические магнитные домены можно передвигать с помощью внешнего магнитного поля.

При движении материя не перемещается. Кристаллы магнитного слоя лишь перемагничиваются (быстрое изменение магнитных свойств внутри кристаллического слоя).

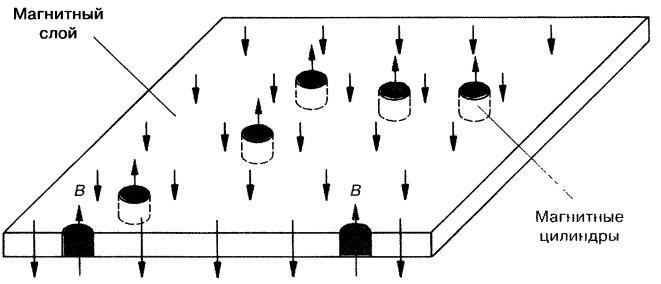


Рис. 12.42. Магнитные цилиндры в магнитном слое из гадолиниум-галлиум-граната ($Cd_3Ga_5O_{12}$).

Цилиндрические магнитные домены не исчезают, если для этого не приложена энергия. Однажды сгенерированные, они остаются до их целенаправленного уничтожения или удаления.

Цилиндрические магнитные домены очень стабильны. Они могут оставаться в магнитном слое в течение многих лет.

12.8.2. Магнитные дорожки

Неупорядоченное движение цилиндрических магнитных доменов в слое нерационально, так как нельзя идентифицировать отдельные цилиндрические магнитные домены и присвоить им логический уровень.

> Цилиндрические магнитные домены должны передвигаться в слое по определенным траекториям.

Производитель магнитной памяти должен спроектировать своеобразную дорогу для доменов, по которой они будут передвигаться.

На магнитоактивный слой напыляется так называемый образец. Во многих случаях это угол (рис. 12.43), однако также широко применяется форма круга и другие формы. Напыляется обычно мягкий железо-никелевый сплав с хорошими магнитными свойствами.

Посредством катушек, которые окружают магнитноактивный слой, производится вращающееся магнитное поле. Уголки образца намагничиваются этим полем. В определенный момент времени, например, угол 1 является углом с максимальной намагниченностью. Эта высшая намагниченность переходит к углу 2, затем к углу 3 и т. д. Цилиндрический магнитный домен перемещается всегда к самому сильно намагниченному уголку (при условии верного направления намагничивания). Если цилиндрический магнитный домен находится при угле 1, он переходит к углу 2, затем к углу 3 и так далее (см. рис. 12.43).

> Цилиндрический магнитный домен всегда передвигается вдоль напыленной «дороги». Вращающееся поле определяет скорость движения.

Так как вращающееся поле действует на все имеющиеся цилиндрические магнитные домены, они перемещаются с равномерной скоростью вдоль заданной траектории.

> Все цилиндрические магнитные домены передвигаются синхронно.

На активный магнитный слой накладываются различные «дороги». Чаще всего траектории имеют форму петли. По ним перемещаются цилиндрические магнитные домены.

> Каждому цилиндрическому магнитному домену присваивается логическое состояние 1.

Цикл движения цилиндрического магнитного домена (см. рис. 12.43) по принципу действия похож на кольцевой сдвиговый регистр. Тактирование обеспечивается вращающимся полем.

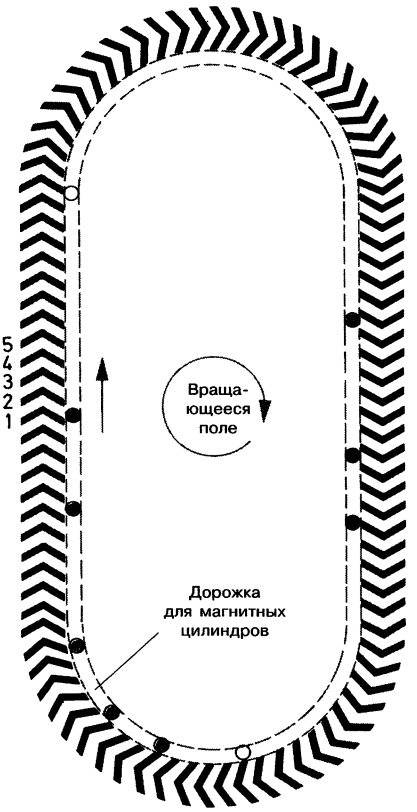


Рис. 12.43. Магнитная дорожка.

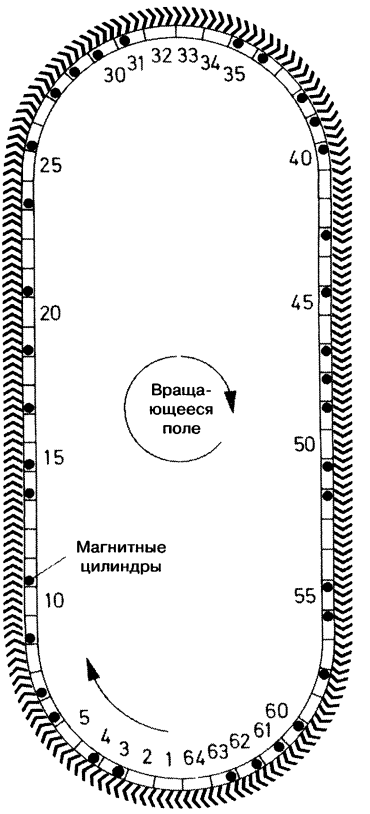


Рис. 12.44. Магнитная дорожка емкостью 64 бита.

Цилиндрические магнитные домены устанавливаются на определенное расстояние. Оно должно быть больше четырехкратного диаметра домена во избежание взаимного влияния. При диаметре домена 2 мкм рационально расстояние 10 мкм. То есть каждые 10 мкм может находиться 1 домен. Если очередного домена нет, значит это место считается 0 битом.

Каждому недостающему цилиндрическому магнитному домену присваивается логическое состояние 0.

На дорожке, по которой перемещаются магнитные домены, умещается определенное число доменных мест. На этих местах могут быть размещены магнитные домены, но могут и не быть.

Каждое доменное место на магнитной дорожке представляет объем памяти емкостью в 1 бит.

На рис. 12.44 представлена магнитная дорожка с объемом памяти 64 бит. Типичные магнитные дорожки имеют емкость 4096 бит и больше.

12.8.3. Запись информации

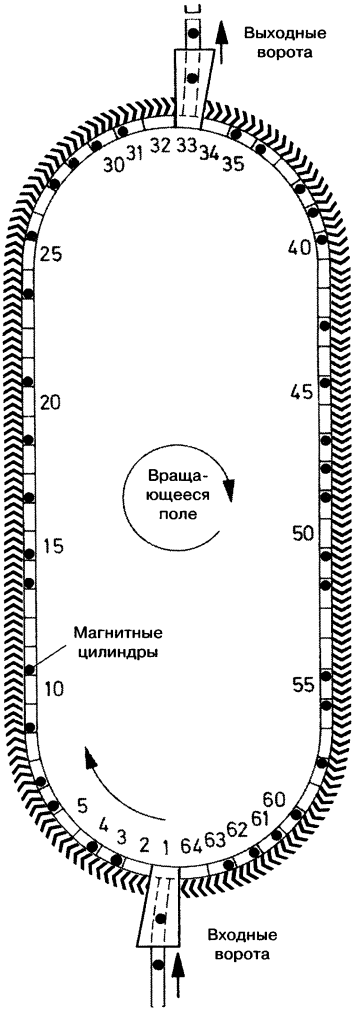


Рис. 12.45. Магнитная дорожка с входными и выходными воротами.

Можно производить и стирать магнитные домены внутри магнитной дорожки. Однако это не принято. Обычно домены вводят и выводят через так называемые ворота. Каждая магнитная дорожка имеет входные ворота и выходные ворота (рис. 12.45).

> Цилиндрические магнитные домены вводятся и выводятся через ворота.

Ворота образуются с помощью проводника. Он действует как маленький электромагнит. При определенном направлении тока возникает магнитное поле, которое блокирует вход для доменов.

При противоположном направлении тока возникает поле, которое пропускает домены внутрь дорожки.

Перед записью новой информации должна быть удалена старая информация. Магнитное кольцо должно быть очищено, т. е. все домены должны быть выведены через выходные ворота.

Необходимые для записи домены и свободные места поступают из кольца ввода данных, рассматриваемого позже.

12.8.4. Чтение информации

При чтении домены выводятся через выходные ворота и попадают в кольцо вывода. Там они проходят мимо детектора доменов. Детектор доменов состоит из крохотной полосы металла на доменной дорожке. Полоса металла находится на пути доменов. Она является частью согласованной мостовой схемы.

Если домен проходит через полосу металла, мост кратковременно разбалансируется. На выходе появляется импульс напряжения. Он означает информационное содержание 1. Если домен не проходит через детектор, то импульса напряжения не возникает. Отсутствие напряжения обозначает информационное содержание 0.

> Цилиндрические магнитные домены считываются детектором доменов.

Информация считывается из магнитной дорожки и стирается оттуда. Это во многих случаях нежелательно. Можно делать ворота, которые увеличивают магнитные домены, вытягивают их и делят на две части. Количе-

ство магнитных доменов таким способом удваивается. Один магнитный домен отправляется на выход, другой остается в магнитной дорожке, называемой **буфером памяти**. В буфере памяти информация сохраняется.

Воротами этой разновидности невозможно очистить буфер памяти. Ворота должны позволять в зависимости от выбора пользователя удалять магнитные домены полностью или удваивать их количество. И то и другое можно достигнуть различной продолжительностью воздействия поля и силой магнитного поля.

12.8.5. Структура магнитной памяти

Магнитная память на базе магнитных доменов содержит большое количество магнитных дорожек. Обычно это 128, 256 или 512 дорожек. В разработке находится память на 1024 и больше дорожек. Каждая дорожка содержит примерно 4096 бит. Обычно 128 дорожек объединяются в блок памяти. Каждый блок памяти имеет генератор доменов и вводную дорожку данных (рис. 12.46).

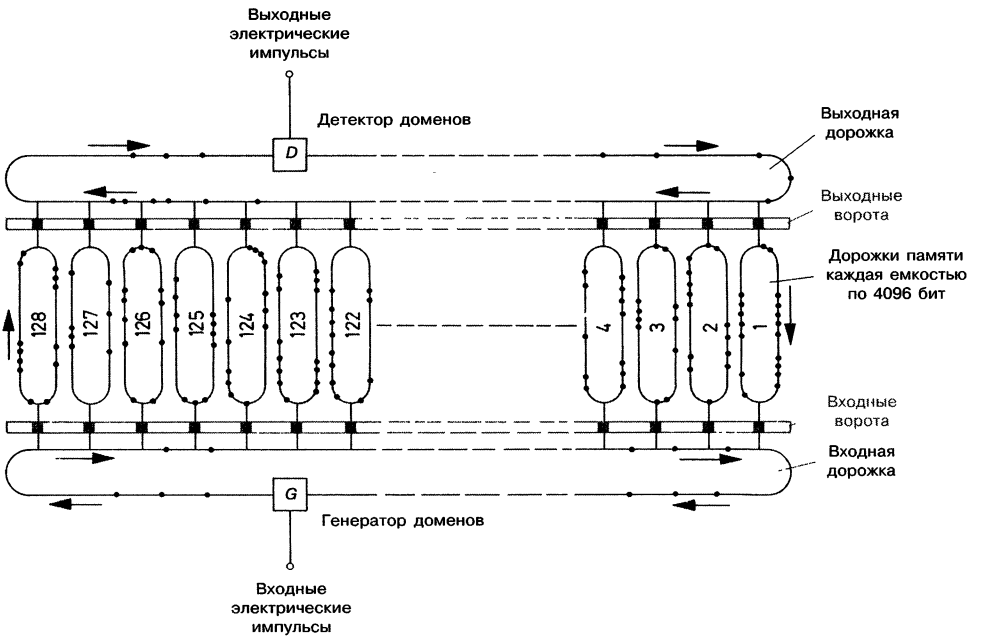


Рис. 12.46. Принципиальная схема памяти на магнитных доменах (128 магнитных дорожек).

На генератор доменов подаются электрические импульсы, которые переводятся в магнитные домены. Эти магнитные домены прежде всего поступают в дорожку ввода данных. В точно установленные моменты времени управляющим импульсом на входные ворота они переводятся в основные магнитные дорожки.

Итак, происходит параллельный ввод данных. В момент ввода данных 128 ячеек памяти находятся перед 128 входными воротами. Каждая ячейка памяти может содержать информацию 0 или 1. Если ячейка памяти не со-

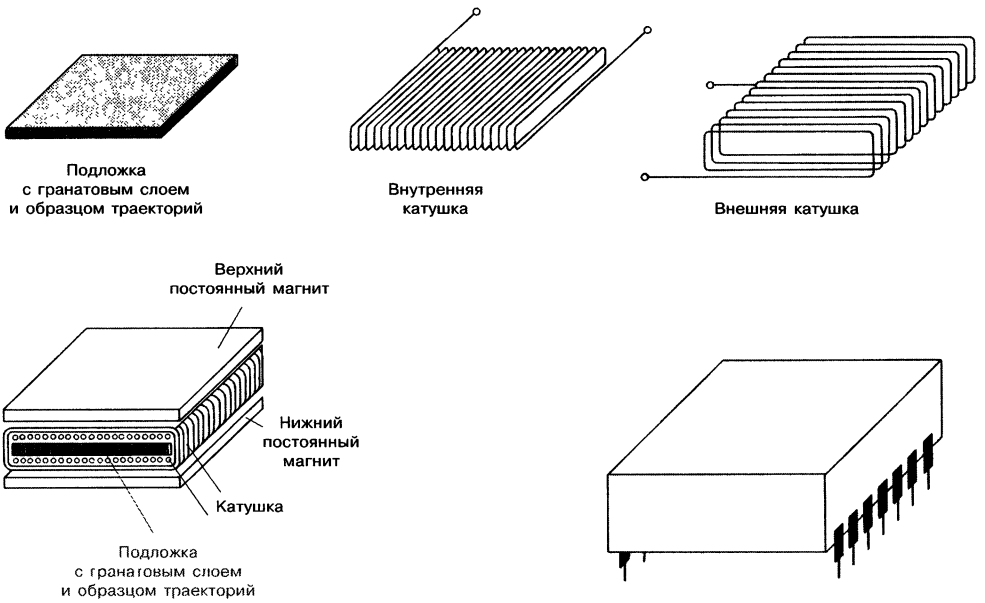


Рис. 12.47. Строение памяти на магнитных доменах.

держит магнитный домен, то она имеет содержание 0, если содержит цилиндрический магнитный домен, то она имеет содержание 1. Соответственно одновременно параллельно передается информация объемом 128 бит.

При выводе данных также всегда выводятся параллельно 128 бит. Цилиндрические магнитные домены кружат в магнитных дорожках. В определенный момент 128 ячеек памяти находятся перед выходными воротами: перед каждой — свои выходные ворота. Если управляющий сигнал подается на выходные ворота, то находящиеся на выходе магнитные домены проходят через ворота и принимаются на выходную магнитную дорожку. Далее они циркулируют по выходной дорожке. Каждый домен производит в детекторе импульс напряжения. После этого они могут быть удалены.

Структура магнитной памяти представлена на рис. 12.47. На подложку нанесен тонкий слой из магнитного материала. Этот материал является слоем, в котором находятся магнитные домены. Магнитные дорожки образуются напылением образцов на магнитный слой. Образцы состоят из магнитномягкого железо-никелевого сплава. Для придания магнитным дорожкам формы нужны два постоянных магнита. Вращающее поле производят две катушки, повернутые относительно друг друга на 90°. Маленькие электромагниты ворот на рис. 12.47 не показаны. Также не видно генератор и детектор доменов. Готовая память похожа на маленький трансформатор в корпусе из пластика (приблизительные габариты — 50 мм × 40 мм × 15 мм).

Контрольный тест

1. Изобразите схему 6-битового сдвигающего регистра для последовательного ввода и вывода данных. При построении схемы должны использоваться *RS*-триггеры.

2. Что понимают в сдвиговом регистре под параллельным вводом, что — под параллельным выводом?

3. Как работает кольцевой регистр?

4. Объясните понятия ОЗУ и ПЗУ.

5. Какие различия состоят между статической оперативной памятью и динамической оперативной памятью?

6. Нарисуйте схему стационарной RAM для 1 бита в *N*-МОП-технологии. Объясните принцип действия этой схемы.

7. Расскажите о достоинствах и недостатках статической RAM в ТТЛ-технологии и в *N*-МОП-технологии.

8. Оперативная память имеет 4 *X*- и 4 *Y*-адресные шины и декларируется как 256 × 4-битовый элемент памяти. Нарисуйте схему этой памяти.

9. Чем отличается ПЗУ от ППЗУ?

10. Что такое масочное программирование?

11. Как устроен EPROM-элемент и как он работает?

12. Что такое плавающий затвор в FAMOS-транзисторе?

13. Имеются в наличии блоки памяти с обозначениями EEROM и EPROM. Чем отличаются эти блоки памяти?

14. Изобразите структуру магнитной матрицы 6 × 6 бит.

15. Что такое цилиндрический магнитный домен и какие свойства он имеет?

16. Что понимают под магнитной дорожкой?

17. Как устроена память на базе магнитных доменов?

18. Объясните процесс записи в память на базе магнитных доменов

ГЛАВА 13

ЦИФРОАНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ, АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ

13.1. Цифроаналоговый преобразователь (ЦАП)

Цифроаналоговый преобразователь, также называемый ЦАП, преобразует цифровую информацию в аналоговую.

13.1.1. Принцип цифроаналогового преобразования

Рассмотрим таблицу, содержащую синусоидальную функцию, или информацию в цифровой форме. По этой таблице может быть изображена синусоида. Она содержит информацию в аналоговой форме. Преобразование таблицы в синусоиду является цифроаналоговым преобразованием.

В цифровой технике управления информация существует чаще всего в бинарном виде, который кодируется по определенному коду. Для этого кода должен быть свой цифроаналоговый преобразователь.

Цифроаналоговый преобразователь может превращать в аналоговые только сигналы определенного двоичного кода.

Некоторые двоичные коды не подходят для цифроаналогового преобразования. Это так называемые **невесовые коды**. Невесовым называют код, элементам которого не присвоены определенные числовые значения. Двоичный код является, например, **весовым кодом**. Каждому элементу двоичного кода, т. е. каждому разряду поставлен в соответствие вес, или степень числа 2. Двоично-десятичный код также является весовым кодом. Код Грея, напротив, является невесовым кодом. Элементам этого кода не соответствует никакое числовое значение (см. гл. 8).

Невесовые коды перед цифроаналоговым преобразованием должны преобразовываться в весовой код.

Преобразование соответствующими преобразователями кода не представляет никаких проблем. Только для кодов с исправлением ошибок могут возникнуть проблемы с избыточными разрядами. Если код является весовым (например код Хемминга), то избыточные разряды должны быть исключены перед преобразованием. Если код невесовой, он должен быть преобразован перед цифроаналоговым преобразованием в весовой код.

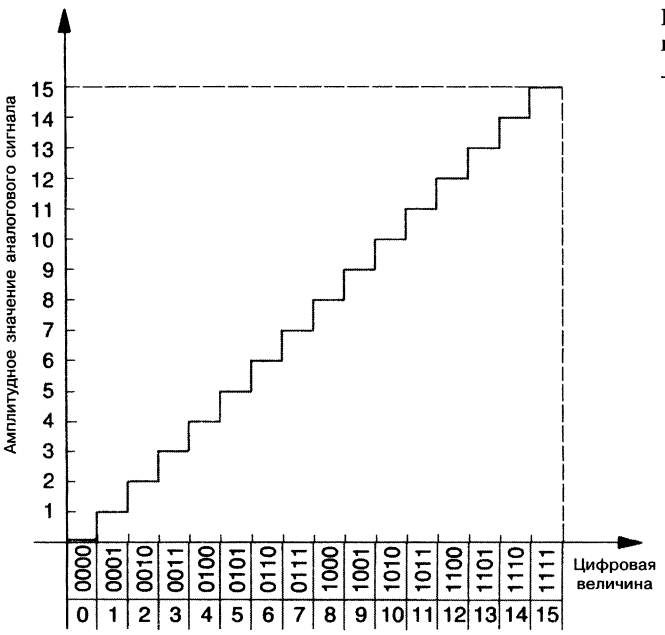


Рис. 13.1. Принцип цифроаналогового преобразования.

Количество бит	Амплитуды
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768

Рис. 13.2. Связь между числом битов и амплитудой.

Принцип цифроаналогового преобразования показан на рис. 13.1. С 4-битовым словом можно образовать 16 числовых значений. Аналоговый сигнал получается в виде ступенчатого напряжения. С помощью 4 бит образуются 16 различных значений амплитуд. Соответственно с 5 битами возможны 32 амплитудных значения, с 6 битами — 64 и т. д. (рис. 13.2).

Получающийся в результате цифроаналогового преобразования аналоговый сигнал является ступенчатым сигналом с определенным количеством возможных амплитудных значений.

Ступеньки могут быть сколь угодно малыми. Они будет тем меньше, чем больше бит в цифровом сигнале.

Ступени сглаживаются фильтрами, так что возникает непрерывно протекающий аналоговый сигнал.

Преобразование цифровых сигналов в аналоговые сигналы возможно разнообразными методами. Самые важные методы представлены ниже.

13.1.2. Цифроаналоговые преобразователи с весовыми резисторами

Принципиальная схема цифроаналогового преобразователя с весовыми резисторами представлена на рис. 13.3. На четыре входа *A*, *B*, *C* и *D* подается 4-битовый цифровой сигнал. Резисторы от R_0 до R_3 подобраны по значимости бит в двоичном коде согласно уравнению:

$$R_n = \frac{R}{2^n}.$$

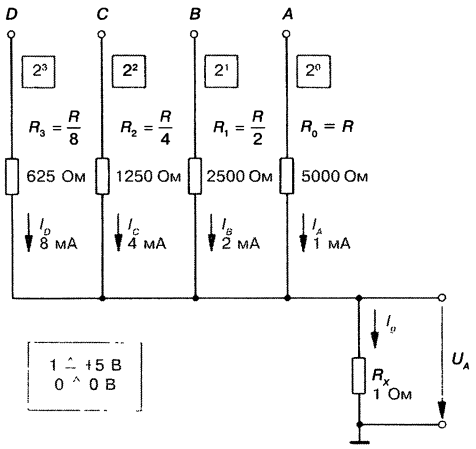


Рис. 13.3. Принципиальная схема ЦАП с весовыми резисторами.

Значение R может выбираться любое в разумных пределах. В этом примере оно равно 5000 Ом. Для R_1 получается 2500 Ом, для R_2 — 1250 Ом. Каждое следующее сопротивление всегда наполовину меньше предыдущего.

В таблице на рис. 13.3а приведены токи, которые получаются, если 1-сигнал соответствует +5 В, и 0-сигнал соответствует 0 В. Получается выходное напряжение, которое тарировано в 1-мВ-шагах. Оно всегда имеет столько милливольт, сколько может представить 4-битовое двоичное число. В итоге получается аналоговый сигнал.

Цифроаналоговый преобразователь на рис. 13.3 работает не очень точно. Сигналы высокого уровня схемы имеют в редких случаях значение точно +5 В. Отклонения в рамках допустимых допусков приводят к ошибочным аналоговым сигналам. Значительно более точно работает схема на рис. 13.4. Здесь используется стабилизированное постоянное напряжение $U_{\text{стаб}}$. Оно подается через транзисторный каскад на резисторы от R_0 до R_3. Операционный усилитель работает как усиливающий сумматор. Резисторы R_V служат ограничителями базового тока. Транзисторы работают в инверсном режиме. В этом режиме работы напряжение насыщения коллектор—эмиттер $U_{CE\,\text{sat}}$ открытого транзистора очень мало (около 20 мВ). На входы от A до D должны подаваться 1-сигналы высокого уровня, которые немного больше, чем 2,6 В.

Цифроаналоговые преобразователи для других весовых двоичных кодов построены аналогичным образом. В зависимости от значимости отдельного бита подбираются сопротивления от R_0 до R_n.

Десятичное значение	D 2^3	C 2^2	B 2^1	A 2^0	I_D мА	I_C мА	I_B мА	I_A мА	I_0 мА	U_A мВ
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	1
2	0	0	1	0	0	0	2	0	2	2
3	0	0	1	1	0	0	2	1	3	3
4	0	1	0	0	0	4	0	0	4	4
5	0	1	0	1	0	4	0	1	5	5
6	0	1	1	0	0	4	2	0	6	6
7	0	1	1	1	1	4	2	1	7	7
8	1	0	0	0	8	0	0	0	8	8
9	1	0	0	1	8	0	0	1	9	9
10	1	0	1	0	8	0	2	0	10	10
11	1	0	1	1	8	0	2	1	11	11
12	1	1	0	0	8	4	0	0	12	12
13	1	1	0	1	8	4	0	1	13	13
14	1	1	1	0	8	4	2	0	14	14
15	1	1	1	1	8	4	2	1	15	15

Рис. 13.3а. Таблица частичных токов I_A, I_B, I_C, I_D общего тока I_g и выходных напряжений U_A по схеме ЦАП с рис. 13.3 для входных сигналов от 0000 до 1111.

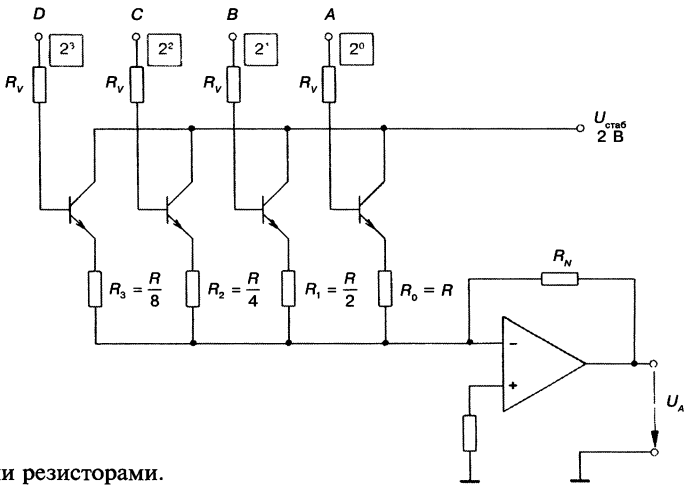


Рис. 13.4. ЦАП с эмиттерными резисторами.

13.1.3. *R*/2*R*-ЦАП

ЦАП можно построить с помощью цепочки резисторов. Для такой цепочки требуются только два различных значения сопротивления — R и его удвоенный номинал $2R$. Такой ЦАП называется $R/2R$ преобразователь. Схема $R/2R$-ЦАП показана на рис. 13.5.

По схеме (рис. 13.5) трудно понять принцип действия. Поэтому для упрощения рассмотрим схему на рис. 13.6, работающую только с двумя переключателями. Номиналы сопротивлений составляют 1 кОм и 2 кОм, напряжение питания $U_{стаб}$ составляет 12 В.

Схема преобразовывает 2-битовые двоичные сигналы в аналоговые сигналы. При $S_0 = 0$ переключатель связан с землей. При $S_0 = 1$ переключатель связан с $U_{стаб}$. Аналогично для переключателя S_1.

Если оба переключателя открыты (заземлены), то на выходе появится напряжение $U_A = 0$ В. Если только переключатель S_0 закрыт, получается схема на рис. 13.8 с общим сопротивлением 3,2 КОм и общим током $I_g = 3{,}75$ мА. Между точкой P и землей находится сопротивление 1,2 КОм, на котором падает напряжение 4,5 В. Оно делится между сопротивлениями R_3 и R_4. На R_4 падение напряжения составляет 3 В.

Если закрыт только переключатель S_1, то напряжение $U_{стаб}$ делится пополам. Если оба переключателя закрыты, то получается выходное напряжение 9 В (рис. 13.7). Шаг напряжения U_A составляет, таким образом, 3 В.

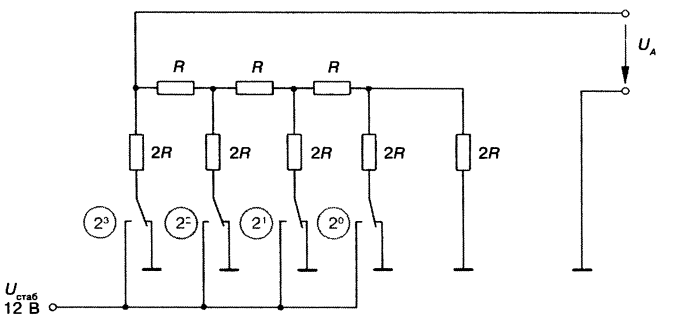


Рис. 13.5. Принципиальная схема R-$2R$-ЦАП.

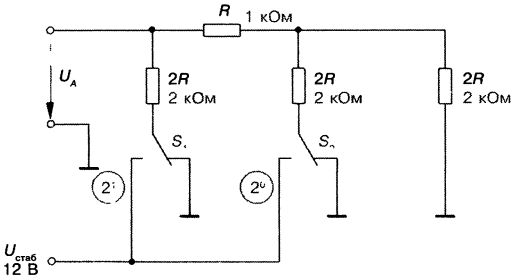


Рис. 13.6. *R*/2*R*-ЦАП для 2-битовых сигналов.

Десятичное значение	S_1 (2^1)	S_0 (2^0)	U_A
0	0	0	0 В
1	0	1	3 В
2	1	0	6 В
3	1	1	9 В

Рис. 13.7. Таблица для схемы с рис. 13.6.

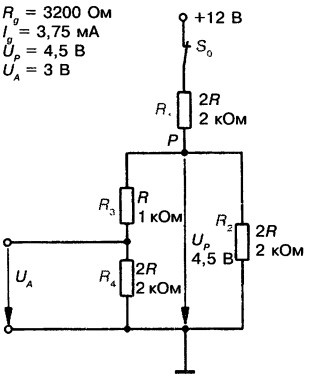


Рис. 13.8. Схема на сопротивлениях.

Шаг U_A получается из количества переключателей и соответственно количества бит. Они рассчитываются уравнением:

$$\Delta U_A = \frac{U_{\text{стаб}}}{2^n}.$$

Для схемы на рис. 13.5 оно составляет, следовательно:

$$\Delta U_A = \frac{U_{\text{стаб}}}{2^n} = \frac{12\ \text{В}}{2^4} = \frac{12\ \text{В}}{16} = 0{,}75\ \text{В}.$$

На практике ключи S_0—S_n заменяются транзисторными каскадами. *R*/2*R*-метод особенно хорошо подходит для микросхем. Должны производиться только два различных номинала сопротивлений.

ЦАП производятся преимущественно в виде микросхем. Из-за незначительного энергопотребления ЦАП чаще выпускаются в виде КМОП-микросхем.

13.2. Аналого-цифровой преобразователь (АЦП)

Аналого-цифровые преобразователи, сокращенно АЦП, преобразуют аналоговые сигналы в соотвествующие цифровые сигналы.

13.2.1. Принцип аналого-цифрового преобразования

Аналоговый сигнал, например сигнал на рис. 13.9, может представляться определенным количеством амплитудных значений. Амплитуда сигнала измеряется, например, каждые 10 мкс. Измеренные цифровые значения будут по очереди сохранены в нужной последовательности. Все цифровые значения вместе образуют цифровой сигнал.

> Цифровой сигнал изменяющейся во времени величины состоит из последовательности чисел.

Числа могут представляться в любой системе счисления или коде. На рис. 13.9 представлены десятичные и двоичные числа.

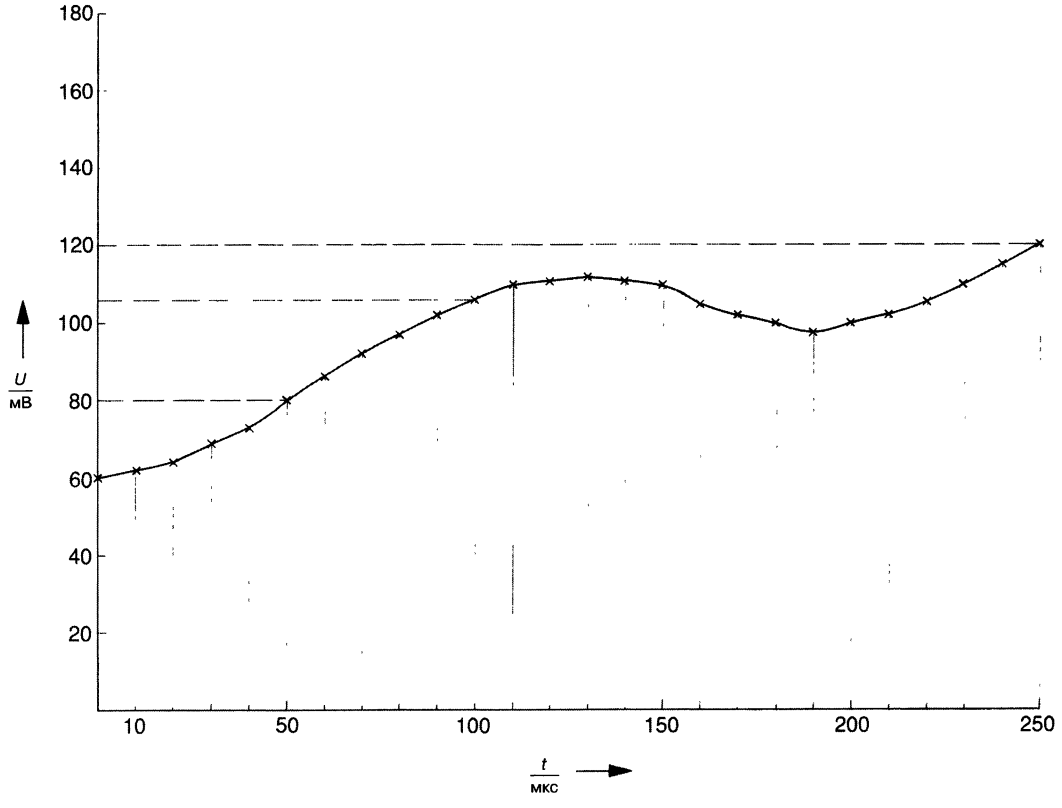


Рис. 13.9. Аналоговый сигнал, представленный амплитудными значениями.

$\frac{t}{\text{мкс}}$	$\frac{U}{\text{мВ}}$	$\frac{U}{\text{мВ}}$						
0	60	0	1	1	1	1	0	0
10	62	0	1	1	1	1	1	0
20	64	1	0	0	0	0	0	0
30	69	1	0	0	0	1	0	1
40	73	1	0	0	1	0	0	1
50	80	1	0	1	0	0	0	0
60	86	1	0	1	0	1	1	0
70	92	1	0	1	1	1	0	0
80	97	1	1	0	0	0	0	1
90	102	1	1	0	0	1	1	0
100	106	1	1	0	1	0	1	0
110	110	1	1	0	1	1	1	0
120	111	1	1	0	1	1	1	1
130	112	1	1	1	0	0	0	0
140	111	1	1	0	1	1	1	1
150	110	1	1	0	1	1	1	0
160	105	1	1	0	1	0	0	1
170	102	1	1	0	0	1	1	0
180	100	1	1	0	0	1	0	0
	:							
	десятичный	двоичный						

АЦП выдают результат чаще всего в двоичной системе счисления или в двоично-десятичном коде.

Амплитуды представляются в определенном масштабе, например в мВ. Если напряжение 4 В должно быть преобразовано с точностью до 1 мВ, требуется 4000 ступеней. Для получения 4000 ступеней напряжения необходимо 12-разрядное двоичное число. Каждое амплитудное значение будет тогда представлено 12 битами. Точность представления числа зависит от количества бит. Она называется **разрешающей способностью АЦП**.

Разрешающая способность АЦП тем выше, чем больше бит он имеет для отображения результата.

Не следует путать разрешающую способность с точностью аналого-цифрового преобразователя. Точность зависит от правильности выданного результата. АЦП с высоким разрешением, то есть с большим количеством бит, могут быть неточны.

Каждый аналого-цифровой преобразователь работает с определенной точностью.

Точность показывает, насколько результат преобразования может отличаться от правильной величины в большую и меньшую сторону. При точности от 10^{-3} результат может отклоняться от правильного результата не более чем на 1/1000. Наиболее дорогие АЦП имеют точность 10^{-5}.

Изменяющийся во времени аналоговый сигнал должен измеряться с определенной частотой. Амплитуды должны измеряться и сохраняться каждые 10 мкс или каждую мс. Частота измерения амплитуд должна быть тем выше, чем быстрее изменяется аналоговый сигнал.

Частота измерения аналогового сигнала должна быть по меньшей мере в два раза выше несущей частоты аналогового сигнала.

Если преобразуется аналоговый звуковой сигнал в частотном диапазоне от 50 Гц до 20 кГц в соответствующий цифровой сигнал, то требуются минимум 40 000 измерений в секунду. Так называемая частота дискретизации составляет 40 кГц. Она может выше, но не меньше. Если она будет меньше, преобразованный сигнал будет искажен.

АЦП производятся в основном в виде микросхем в *K*-МОП-технологии. Производство с помощью дискретных элементов экономически нецелесообразно. Схемы в ТТЛ- и ЭСЛ-исполнении из-за высокого энергопотребления применяются только там, где требуется высокая скорость.

АЦП характеризуются следующими параметрами:

- *разрешающая способность* — количество бит;
- *точность* — ошибка в % от результата, или % от максимального значения;
- *скорость* — длительность преобразования;
- *напряжение* — диапазон от мин. до макс. преобразуемого напряжения.

Существует множество различных методов и схем преобразователей. Они будут рассмотрены в следующих разделах.

13.2.2. АЦП последовательного счета

АЦП последовательного счета оцифровывает аналоговый сигнал линейно-нарастающим эталонным напряжением (пилообразной формы). Передний фронт начинается в отрицательной области напряжения (рис. 13.10). К моменту времени ①, если эталонное напряжение переходит через нуль, запускается двоичный счетчик. Он считает импульсы генератора. Если напряжение аналогового сигнала достигнет эталонного напряжения (точка ②), то счетчик остановится.

Счетчик считает в течение времени Δt. Он измеряет время. Так как пилообразное напряжение растет с заранее определенным углом α, то можно рассчитать по Δt и углу α значение амплитуды u:

$$\tan\alpha = \frac{u}{\Delta t};$$

$$u = \Delta t \cdot \tan\alpha.$$

Двоичное число, которое показывает счетчик, является значением напряжения u. Частоту генератора импульса можно выбирать так, чтобы счетчик показывал напряжение в требуемых единицах измерения, например в мВ.

Максимальное значение пилообразного сигнала определяет максимально возможное значение аналогового сигнала. Если амплитуда пилообразного сигнала равна, например 10 В (рис. 13.10), то могут быть оцифрованы аналоговые напряжения только до максимального значения 10 В.

На рис. 13.11 показана принципиальная схема АЦП последовательного счета. Пилообразное напряжение подается на две схемы компаратора. На выходе компаратора бинарный сигнал 1 действует только тогда, когда оба входа одинаковы по величине (аналоговый компаратор или компаратор с аналоговыми входами).

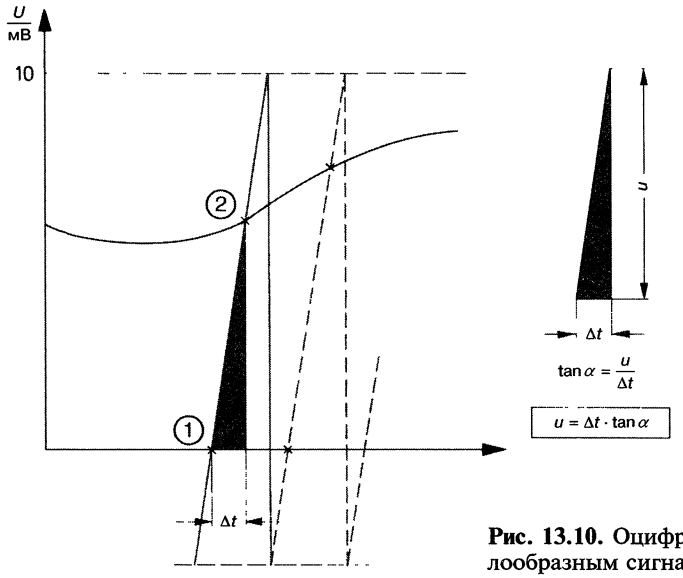


Рис. 13.10. Оцифровка аналогового сигнала пилообразным сигналом.

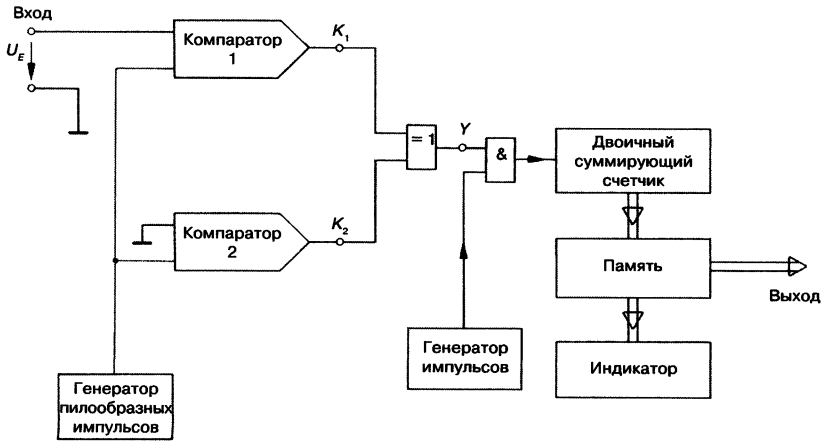


Рис. 13.11. Структура АЦП по методу пилообразного напряжения.

Если пилообразное напряжение имеет отрицательное значение, то на выходе компаратора 2 действует 0-сигнал ($K_2 = 0$).

Если пилообразное напряжение переходит через линию нуля (точку 1), $K_2 = 1$, так как оба входных напряжения равны. На выходе Y ИСКЛЮЧАЮЩЕГО ИЛИ теперь появляется 1-сигнал. И-элемент включает генератор импульсов на счет. Как только пилообразное напряжение становится равно аналоговому напряжению, компаратор 1 также выдает 1-сигнал ($K_1 = 1$). Теперь $Y = 0$. И-элемент блокирует генератор импульсов. Цифровой сигнал счетчика подается в память и далее обрабатывается. Затем он отображается, например, на цифровой индикатор. Точность схемы очень сильно зависит от линейности пилообразного напряжения.

АЦП на рис. 13.11 работает только для положительного измеряемого напряжения. Однако неизвестное аналоговое напряжение можно сложить с постоянным, чтобы поднять его в положительную полуплоскость. После преобразования постоянное напряжение вычитается из результата.

Схема на рис. 13.11 может быть изменена таким образом, чтобы было возможно измерять как положительные, так и отрицательные сигналы.

13.2.3. АЦП с двойным интегрированием (Dual Slope)

АЦП с двойным интегрированием работают в два этапа. Поэтому этот метод также еще называется двухшаговым методом. Dual Slope — английский термин. Он значит «два импульса с разным наклоном». Основным узлом аналого-цифрового преобразователя является интегратор (рис. 13.12). Выходное напряжение интегратора представляет собой два фронта с различным наклоном.

На 1-м шаге интегрируется положительное аналоговое напряжение в течение заданного времени t_1. Конденсатор C интегратора заряжается.

На 2-м шаге на вход интегратора подается отрицательное опорное напряжение. Конденсатор C разряжается до нулевого выходного напряжения интегратора. Время, за которое происходит разрядка интегратора до нуля, характеризует величину аналогового напряжения. Это время t_2.

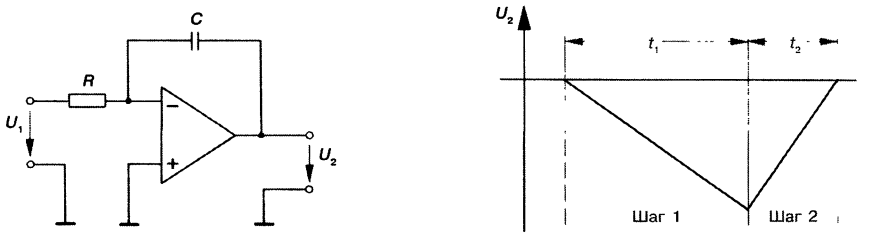


Рис. 13.12. Интегратор с вводом напряжения во время 1-го и 2-го шагов.

В течение времени t_2 суммирующий счетчик производит подсчет импульсов. Импульсы поступают с выхода генератора импульсов. По истечении времени t_2 счетчик показывает бинарную величину. Это значение амплитуды аналогового сигнала, выраженной в цифровой форме. Если аналоговый сигнал преобразуется в двоичный код, то счетчик выбирается двоичным. Счетчик всегда должен считать в коде, в который преобразуется аналоговый сигнал.

Из-за принципа двойного интегрирования этот метод называется АЦП с двойным интегрированием.

Блок-схема АЦП с двойным интегрированием представлена на рис. 13.13. Схема управления включает электронный переключатель *S*. Перед началом процесса преобразования к моменту времени t_1 переключатель переводится в положение 1. В этом положении он остается в течение времени t_1. Время t_1 задается генератором импульсов через делитель частоты. Это четко заданное время, например 100 мкс. В течение времени t_1 *C* заряжается через *R*.

По истечении времени t_1 электронный переключатель переключается в положение 2. Опорное напряжение подается на вход интегратора. Одновременно с выхода элемента И от генератора импульсов начинают поступать импульсы, которые считаются счетчиком. Начинается отсчет времени t_2. Из интегратора выходит ток, и *C* разряжается. В течение времени t_2 сигналы на разных входах компаратора не равны. На выходе *X* действует 0-сигнал. Он инвертируется и появляется как 1-сигнал на входе И-элемента.

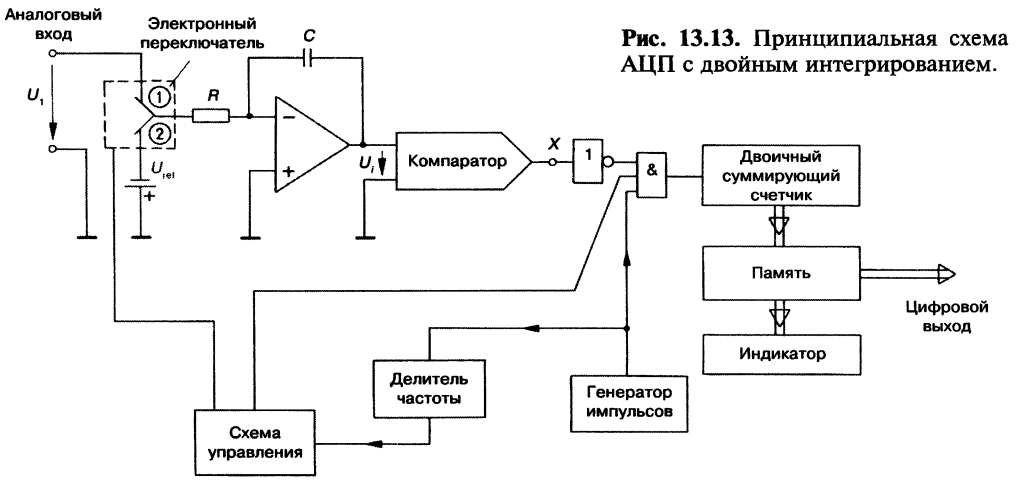


Рис. 13.13. Принципиальная схема АЦП с двойным интегрированием.

В момент времени t_2 выходное напряжение интегратора U_2 станет равным нулю. Оба входных напряжения компаратора теперь равны 0 В. На выходе X появляется 1-сигнал. Он инвертируется в 0-сигнал, который блокирует И-элемент. Подсчет импульсов закончен и получен цифровой результат.

Аналого-цифровые преобразователи с двойным интегрированием работают медленнее, чем АЦП последовательного счета. Однако при одинаковой стоимости они более точные. Значения R и C не влияют на точность результата. Точность АЦП зависит прежде всего от отношения t_1/t_2 и от величины опорного напряжения U_{ref}:

$$\frac{U_1}{U_{ref}} = \frac{t_2}{t_1}.$$

Так как времена t_1 и t_2 в равной мере зависят от частоты генератора импульсов, эта частота также не влияет на результат. Разумеется, она должна быть постоянна в течение времени $t_1 + t_2$.

13.2.4. АЦП компенсационного типа

В АЦП компенсационного типа преобразуемое аналоговое напряжение U_1 сравнивается с компенсационным напряжением U_K. Компенсационное напряжение характеризует выходное напряжение цифроаналогового преобразователя (ЦАП) и поэтому называется также напряжением преобразователя. На ЦАП подаются цифровые значения до тех пор, пока напряжение преобразователя U_K не сравняется с преобразуемым аналоговым напряжением U_1. Принципиальная схема изображена на рис. 13.14.

Перед началом процесса преобразования счетчик устанавливается схемой управления на нулевое значение. На выходе компаратора действует $X = 0$. Этот сигнал инвертируется и подается на вход И-элемента. И-элемент пропускает импульсы генератора импульсов к счетчику. Счетчик начинает считать на увеличение от нуля.

Цифровой сигнал счетчика превращается цифроаналоговым преобразователем в соответствующий аналоговый сигнал. Из-за суммирующего счет-

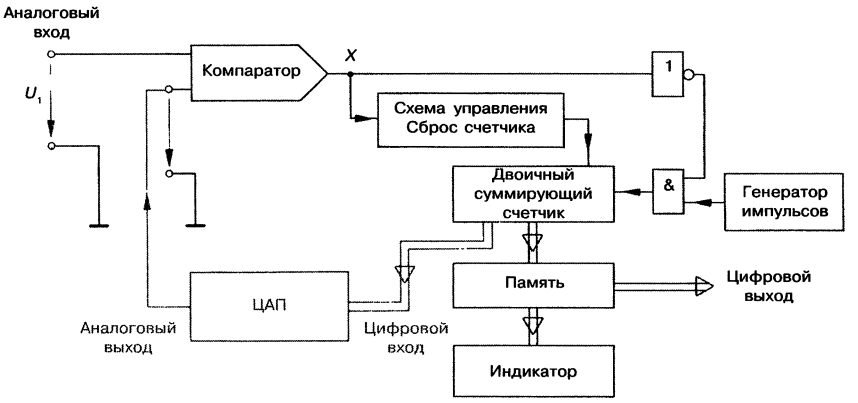


Рис. 13.14. Принципиальная схема АЦП компенсационного типа.

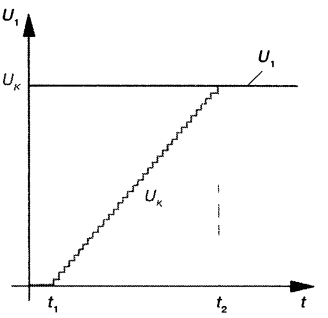


Рис. 13.15. Вид напряжения в процессе преобразования (t_1 = начало, t_2 = завершение преобразования, U_1 = аналоговое напряжение).

чика возникает ступенчато поднимающийся выходной сигнал U_K. Когда напряжение U_K достигнет напряжения U_1 (рис. 13.15), на выходе компаратора появится 1-сигнал. Этот сигнал инвертируется. 0-сигнал блокирует И-элемент. Следующие импульсы не доходят до счетчика, и счетчик перестает считать.

Сигнал $X = 1$ подводится к другой схеме управления. Она отвечает за ввод показаний счетчика в память. Показание счетчика и является результатом аналогового преобразования. Результат готов для дальнейшей обработки и может быть выведен на индикатор.

Схема управления сбрасывает счетчик на нуль. Начинается новый цикл преобразования. АЦП, построенные по этому принципу, довольно медленны, так как каждый цикл преобразования начинается с нуля.

Число возможных циклов преобразования в единицу времени может быть сильно увеличено при применении реверсивного счетчика. Структурная схема такого АЦП приведена на рис. 13.16. Компаратор заменен дифференциальным усилителем. На выходе усилителя действует положительное напряжение до тех пор, пока U_1 меньше, чем U_K. Положительное выходное напряжение переключает счетчик на счет в сторону увеличения. Кроме того, положительное выходное напряжение увеличивает частоту колебаний управляемого напряжением осциллятора. Импульсы приходят чаще. Счетчик считает быстрее. Напряжение U_K растет быстрее. Чем ближе напряжение U_K к напряжению U_1, тем меньше положительное выходное напряжение усилителя, тем меньше частота колебаний осциллятора, тем мед-

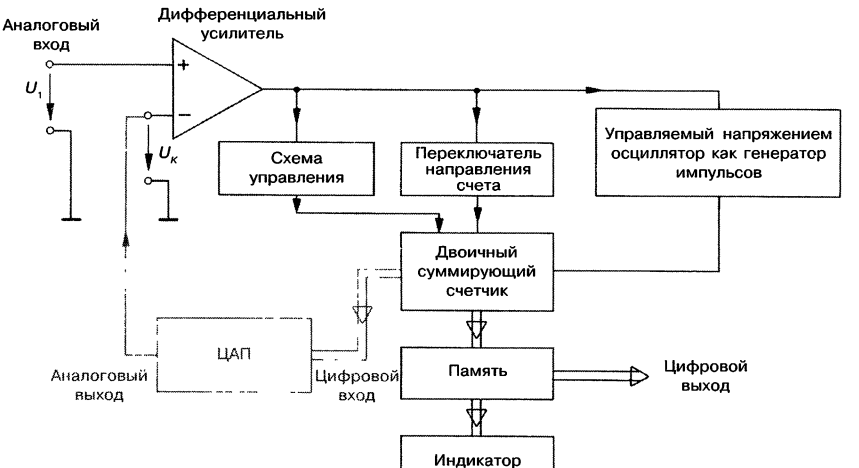


Рис. 13.16. Принципиальная схема АЦП компенсационного типа с непрерывным сравнением.

леннее считает счетчик. Напряжение U_K постепенно приближается к значению напряжения U_1. При $U_K = U_1$ счетчик останавливает счет, полученный результат записывается в память, и его можно обрабатывать дальше.

Если аналоговое напряжение U_1 продолжает изменяться, то процесс преобразования продолжается. На выходе усилителя появляется положительное напряжение, счетчик считает до тех пор, пока U_K не станет равным U_1.

Если U_1 меньше, чем U_K, то на выходе усилителя появляется отрицательное напряжение. Отрицательное выходное напряжение переключает счетчик на счет на уменьшение. Счетчик считает обратно, U_K уменьшается до тех пор, пока не сравняется с U_1. После этого счет прекращается, и показания счетчика сохраняются.

Показания счетчика остаются всегда поблизости от значения U_1. При изменении U_1 требуется немного импульсов в прямом или обратном направлении до равенства напряжений U_K и U_1. Поэтому процесс преобразования происходит очень быстро.

Управляемый напряжением осциллятор реагирует на напряжение без учета знака. Он колеблется при больших положительных или отрицательных напряжениях быстрее и при малых положительных или отрицательных напряжениях медленнее.

13.2.5. АЦП по принципу напряжение—частота

Преобразуемый аналоговый сигнал преобразуется в АЦП в переменное напряжение определенной частоты. Для этого существует множество схем. Простейшей схемой был бы LC-осциллятор, к конденсатору колебательного контура которого параллельно подключен емкостной диод. Емкость диода управляется аналоговым напряжением преобразователя. Вследствие этого изменяется резонансная частота колебательного контура и выходная частота LC-осциллятора (см. «Электроника», ч. 3). Такая схема в принципе пригодна, но она слишком неточна.

АЦП этого типа должен иметь хорошую линейную связь между напряжением и частотой. Если значение напряжения возрастает, например, в 2 раза, то и частота должна также возрастать в 2 раза. Частота измеряется цифровым способом. Соответствующее измеренной частоте значение напряже-

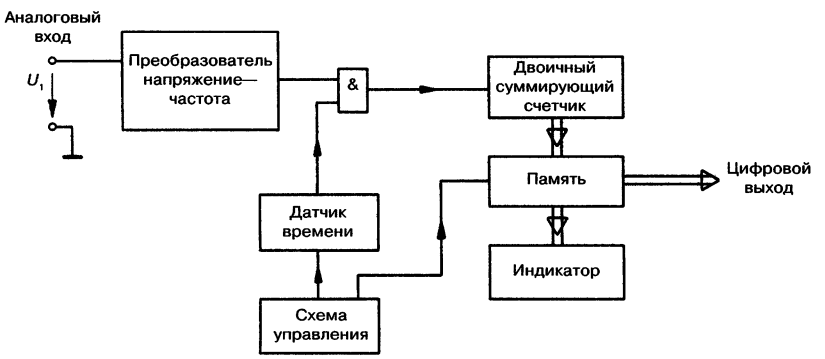


Рис. 13.17. Принципиальная схема АЦП по принципу напряжение—частота.

ния сохраняется как цифровой сигнал. Он может быть выведен на индикатор или обработан.

На рис. 13.17 показана принципиальная схема АЦП. Схема управления запускает процесс преобразования. Датчик времени подает 1-сигнал на вход И-элемента. От преобразователя напряжение—частота приходят импульсы, полученные от сигнала, например положительные полуволны напряжения. Они проходят через И-элемент и считаются счетчиком.

По истечении заданного времени датчик времени останавливает счет. Он подает на И-элемент 0-сигнал. Управляющая электроника разрешает запись результата измерения в память. После этого может начинаться следующий цикл преобразования. Точность результата изменения зависит прежде всего от линейности преобразователя и от точности датчика времени.

13.2.6. АЦП прямого преобразования

АЦП прямого преобразования работает с компараторами, которые всегда переключают выходное состояние 0 в выходное состояние 1, если на положительном входе напряжение равно или больше, чем на отрицательном.

Каждый выходной бит привязан к отдельному компаратору. Схема на рис. 13.18 работает с 8 компараторами. Каждый компаратор получает заданное фиксированное опорное напряжение, которое подается на отрицательный вход. Опорные напряжения стандартизованы (например 0,5 В, 1,5 В, 2,5 В, 3,5 В, 4,5 В, 5,5 В, 6,5 В, 7,5 В).

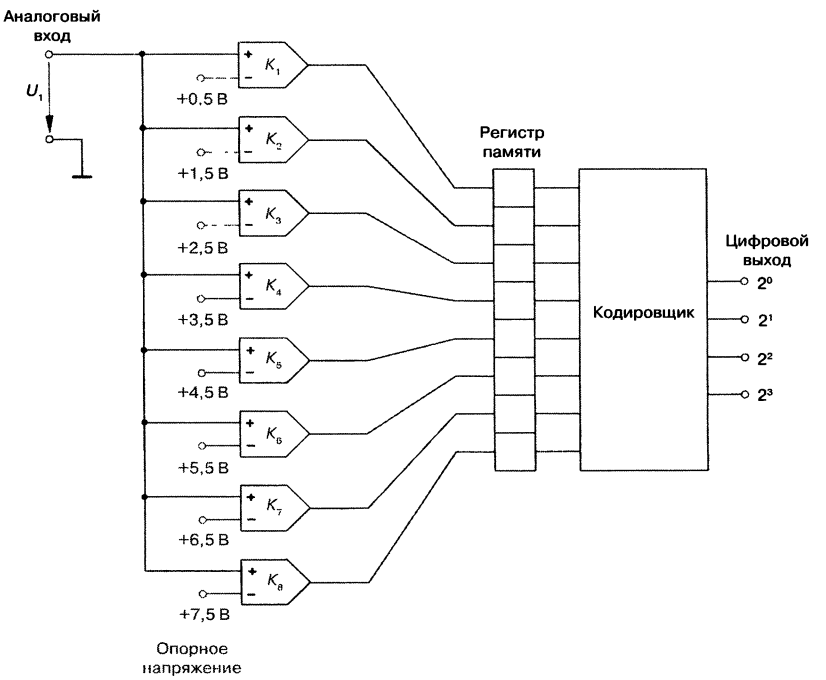


Рис. 13.18. Принципиальная схема АЦП прямого преобразования.

$\frac{U_i}{B}$	K_8	K_7	K_6	K_5	K_4	K_3	K_2	K_1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
3	0	0	0	0	0	1	1	1
4	0	0	0	0	1	1	1	1
5	0	0	0	1	1	1	1	1
6	0	0	1	1	1	1	1	1
7	0	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
Напряжения	7,5 В	6,5 В	5,5 В	4,5 В	3,5 В	2,5 В	1,5 В	0,5 В

Рис. 13.19. Рабочая таблица АЦП по схеме с рис. 13.18.

Если преобразуемое напряжение увеличивается с нуля, то при достижении самого малого опорного напряжения (0,5 В) компаратор выдает на выход 1-сигнал. При достижении следующего опорного напряжения (1,5 В) следующий компаратор выдает сигнал 1 на выход и так далее. Выходные сигналы представлены в рабочей таблице на рис. 13.19. С 8 компараторами возможны 8 уровней напряжения. Выходной код АЦП прямого преобразования по мере необходимости преобразуется в другой код, например в двоичный.

АЦП прямого преобразования работают быстрее, чем все другие виды АЦП. Время преобразования определяется временем переключения компараторов. Оно равняется примерно от 40 до 50 нс. Значит, при 100 нс на одно преобразование каждую секунду возможны 10 миллионов преобразований. Стоимость такого АЦП, разумеется, очень большая. Для аналого-цифрового преобразователя с 128 уровнями напряжения требуются 128 компараторов. 256 уровней напряжения нужны для 8-битового преобразователя. Для 10-битового АЦП с 1024 шагами напряжения необходимо иметь 1024 компаратора.

Высокая степень интеграции современных микросхем позволяет создавать АЦП прямого преобразования с хорошим разрешением. Точность преобразования зависит от точности опорных напряжений и от степени точности компараторов.

Контрольный тест

1. Как работает ЦАП с весовыми резисторами?
2. Цифровые сигналы, представленные в невесовом коде, должны быть преобразованы в аналоговый сигнал. Что нужно делать?
3. Что понимают под разрешающей способностью аналого-цифрового преобразователя?
4. Какие характеристики следует учитывать при выборе аналого-цифрового преобразователя?
5. Перечислите различные виды АЦП.
6. Объясните принцип действия АЦП последовательного счета.
7. Какие преимущества и недостатки имеет АЦП прямого преобразования?

ГЛАВА 14

СЧЕТНЫЕ СХЕМЫ

С помощью цифровых схем могут производиться арифметические операции, например сложение и вычитание. Такие схемы называются **счетными схемами**.

> Счетные схемы производят над входными переменными логические операции, которые соответствуют нужной арифметической операции.

Входные числа должны быть закодированы в определенном двоичном коде. В таком же коде будет выведен результат.

> Каждая счетная схема предназначена только для одного кода или соответствующей системы счисления.

Часто используется двоичный код, т. е. двоичная система счисления, и двоично-десятичный код (см. гл. 8).

14.1. Полусумматор

Полусумматор является самой простой счетной схемой.

> Полусумматор может складывать два двоичных числа.

Действуют следующие правила:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

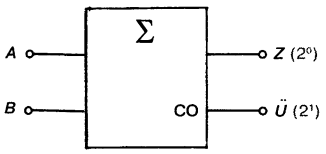
$$1 + 1 = 10.$$

Прибавляемая цифра обозначается переменной A. Другая прибавляемая цифра обозначается переменной B. Схема должна иметь два выхода. Выход с весом 2^0 назовем Z, выход с весом 2^1 назовем U (перенос). Если двоичное число 0 ставится в соответствие состоянию 0 и двоичное число 1 ставится в соответствие состоянию 1, то получается таблица истинности на рис. 14.1.

Из таблицы истинности легко могут быть определены через нормальные формы ИЛИ логические функции полусумматора. Полные конъюнкции выделены серым на рис. 14.1. Полусумматор обозначается знаком суммирования (DIN 40900, часть 12).

$$Z = \left(A \wedge \overline{B}\right) \vee \left(\overline{A} \wedge B\right);$$

$$\ddot{U} = A \wedge B.$$



Вар.	B	A	Ü	Z	
1	0	0	0	0	
2	0	1	0	1	→ A ∧ B̄
3	1	0	0	1	→ Ā ∧ B
4	1	1	1	1	
					→ A ∧ B

Рис. 14.1. Полусумматор с таблицей истинности:
CO — выход переноса (от англ. carry out). Σ — знак суммы.

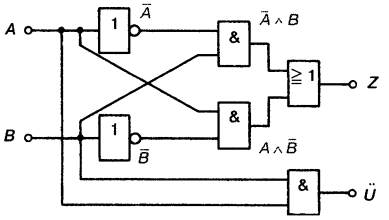


Рис. 14.2. Схема полусумматора на базе основных элементов.

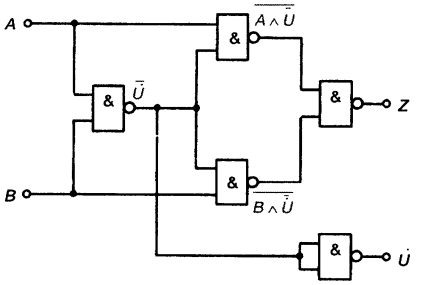


Рис. 14.3. Схема полусумматора на элементах И-НЕ.

Получающаяся по уравнениям схема изображена на рис. 14.2. Схема может быть пересчитана в базисе И-НЕ. Особенно простая схема получается при многократном использовании $\overline{\overline{U}}$ (рис. 14.3).

$$Z = \left(A \wedge \overline{B}\right) \vee \left(\overline{A} \wedge B\right) = \left(A \vee \overline{A}\right) \wedge \left(A \vee B\right) \wedge \left(\overline{B} \vee \overline{A}\right) \wedge \left(\overline{B} \vee B\right);$$

$$Z = \left(A \vee B\right) \wedge \left(\overline{A} \vee \overline{B}\right) = \left(A \vee B\right) \wedge \overline{A \wedge B} = \left(A \wedge B\right) \wedge \overline{\overline{U}};$$

$$Z = \left(A \wedge \overline{\overline{U}}\right) \vee \left(B \wedge \overline{\overline{U}}\right);$$

$$Z = \overline{\overline{\left(A \wedge \overline{\overline{U}}\right)} \vee \left(B \wedge \overline{\overline{U}}\right)} = \overline{\overline{A \wedge \overline{\overline{U}}} \wedge \overline{B \wedge \overline{\overline{U}}}}.$$

14.2. Полные сумматоры

Для построения полных сумматоров нужны схемы, которые могут складывать три двоичных числа, так как при сложении двух двоичных чисел необходимо уметь складывать переносы.

Пример

	1	1	1	1		
		1	0	1	1	
+		0	1	1	1	
	1	0	0	1	0	

Полным сумматором называется схема, которая может складывать три двоичных числа.

Рис. 14.4. Полный сумматор и его таблица истинности. Выход CO выдает знак переноса, на вход CI подается знак переноса (от англ. carry in input). Σ — знак суммы.

Вар.	C	B	A	Ü	Z
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

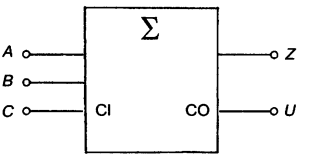


Схема полного сумматора проектируется по правилам синтеза схем (см. гл. 5). Полный сумматор имеет три входа — для каждого складываемого двоичного числа. Назовем их A, B и C. Выходы называются, как и в полусумматоре, Z и $\ddot{U}$.

Таблица истинности полного сумматора получается из правил сложения. Она представлена на рис. 14.4. В случае 1 выходы Z и $\ddot{U}$ равны нулю, так как все входные слагаемые являются нулями. В случае 2 в результате сложения 0 + 0 + 1 $Z = 1$ и $\ddot{U} = 0$. В случае 4 в результате сложения 0 + 1 + 1 получается $Z = 0$ и $\ddot{U} = 1$. Рассмотрим еще вариант 8. Сложение 1 + 1 + 1 дает в результате $Z = 1$ и $\ddot{U} = 1$. Нормальная форма ИЛИ для Z состоит из четырех полных конъюнкций:

$$Z = \left(A \wedge \bar{B} \wedge \bar{C}\right) \vee \left(\bar{A} \wedge B \wedge \bar{C}\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C\right) \vee \left(\bar{A} \wedge \bar{B} \wedge C\right).$$

Нормальная форма ИЛИ для $\ddot{U}$:

$$\ddot{U} = \left(A \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge C\right) \vee \left(\bar{A} \wedge B \wedge C\right) \vee \left(A \wedge B \wedge C\right).$$

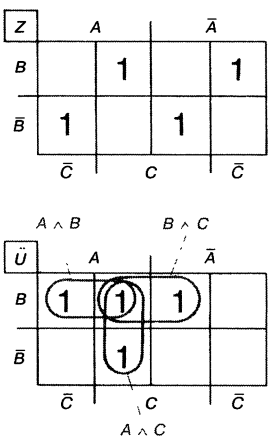


Рис. 14.5. Карта Карно полного сумматора.

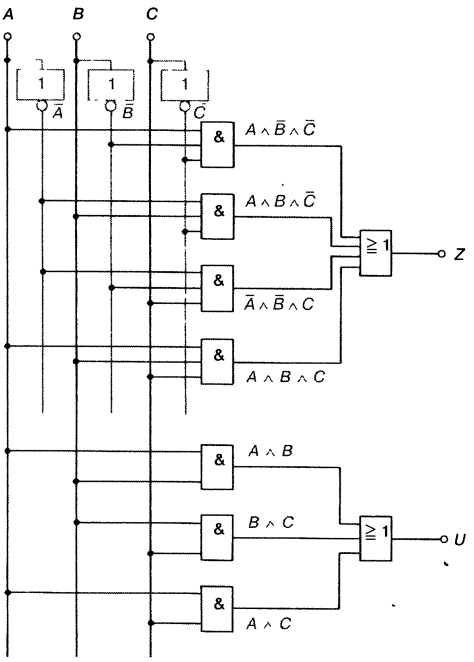


Рис. 14.6. Принципиальная схема полного сумматора.

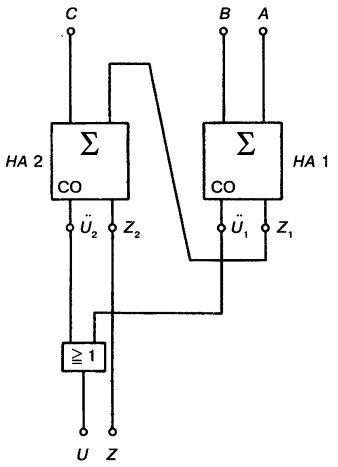


Рис. 14.7. Полный сумматор, построенный на базе двух полусумматоров.

Уравнение для Z дальше не упрощается (рис. 14.5). Для $\ddot{U}$ с помощью карты Карно получают упрощенное уравнение:

$$\ddot{\ddot{U}} = (A \wedge B) \vee (B \wedge C) \vee (A \wedge C).$$

Эти уравнения приводят к схеме на рис. 14.6.

> Полный сумматор может быть построен из двух полусумматоров и элемента ИЛИ.

Схема показана на рис. 14.7. Эту схему также называют **полусумматором**. Два полусумматора образуют полный сумматор. Только дополнительно нужно подключить элемент ИЛИ.

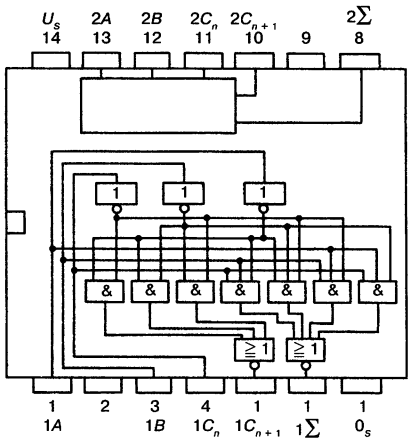
Полные сумматоры производятся в основном в виде микросхем. Они еще называются 1-битовые полные сумматоры, так как при сложении, как показано в примере в начале разд. 14.2, можно сложить только один столбец чисел (серая рамка).

В качестве примера приведем популярную микросхему FLH 451-74H183. Она содержит два 1-битовых полных сумматора в ТТЛ-исполнении. Полная таблица данных приведена на рис. 14.8.

Два быстрых 1-битовых полных сумматора
FLH451-74H183
FLH455-84H183

Тип	Номер для заказа
FLH 451-74H183	Q67000-H495
FLH 455-84H183	Q67000-H551

Цоколевка
Вид сверху



Нет перевода

Вход			Выход	
C_n	B	A	$\Sigma = T$	$C_{n+1} = \ddot{U}$
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

Данные поступают в модуль FLH 451/455 через входы *A*, *B* и C_n (перенос от предыдущего младшего разряда), результат сложения выдается через выход сумматора $\sum$ и выход C_{n+1} (перенос для старшего разряда).

Статические параметры в температурных зонах 1 и 5		Условия испытаний		Нижний предел *B*	Тип	Верхний предел *A*	Ед. измерения
Напряжение питания	U_S			4,74	5,0	5,25	В
Входное напряжение *H*-уровня	U_{IH}	U_S = 4,75 В		2,0			В
Входное напряжение *L*-уровня	U_{IL}	U_S = 4,75 В				0,8	В
Входное напряжение на клеммах	$-U_I$	U_S = 4,75 В, $-I_I$ = 8 мА				1,5	В
Выходное напряжение *H*-уровня	U_{QH}	U_S = 4,75 В, U_{IH} = 2,0 В, $-I_{QH}$ = 1 мА		2,4	3,5		В
Выходное напряжение *L*-уровня	U_{QL}	U_S = 4,75 В, U_{IL} = 0,8 В, I_{QL} = 20 мА			0,2	0,4	В
Входной ток на канал	I_I	U_I = 5,5 В	U_S = 5,25 В			1	мА
H-входной ток на канал	I_{IH}	U_{IH} = 2,4 В				150	мкА
L-входной ток на канал	$-I_{IL}$	U_{IL} = 0,4 В				6	мА
Ток короткого замыкания на канал выхода	$-I_Q$			40		100	мА
Ток питания *H*-уровня	I_{SH}	U_{IQ} = 4,5 В			40		мА
L-уровня	I_{SL}	U_{IL} = 0 В			48	75	
Время переключения, при U_S = 5 В, T_U = 25 °C							
Время прохождения сигнала (быстродействие)	t_{PLH}	C_L = 25 пФ, R_L = 280 Ом			10	15	нс
	t_{PHL}				12	18	нс
Логические параметры							
H-коэффициент разветвления по выходу на канал	F_{QH}					24	
L-коэффициент разветвления по выходу на канал	F_{QL}					12	
Входной нагрузочный коэффициент на *T*	F_I					3,75	

Рис. 14.8. Таблица данных интегральной микросхемы FLH 451-74H183 (Siemens).

14.3. Параллельный сумматор

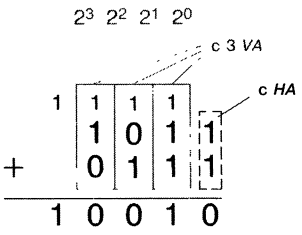


Рис. 14.9. Сложение двух четырехзначных чисел.

Если нужно сложить два двоичных числа за один шаг, потребуется один полусумматор и три полных сумматора. Первый столбец справа (вес 2^0) может складываться полусумматором, так как в этом столбце никогда не возникнет перенос. В других трех столбцах с весом 2^1, 2^2 и 2^3 могут встречаться переносы. Для сложения этих столбцов необходимы полные сумматоры (рис. 14.9).

Сложение за один шаг называется **параллельным сложением**. 4-битовый параллельный сумматор изображен на рис. 14.10. На входы полусумматора *HA* поступают первые цифры обоих справа складываемых чисел (вес 2^0). Выход Z_0 подключен к регистру результата. Выход переноса U_0 связан с входом полного сумматора *VA*1 для второго столбца, так как в этом столбце должен складываться возникающий перенос.

Полный сумматор *VA*1 для второго столбца получает, кроме переноса полусумматора, вторые цифры складываемых чисел (вес 2^1). Выход Z_1 полного сумматора выдает одну цифру результата. Выход переноса U полного сумматора подключен на вход полного сумматора *VA*2 для третьего столбца (вес 2^2). Этот полный сумматор содержит, кроме того, третьи цифры справа складываемых чисел.

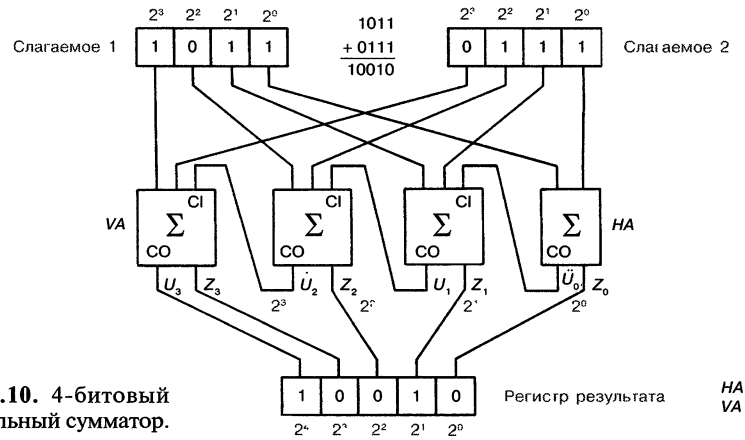


Рис. 14.10. 4-битовый параллельный сумматор.

Полный сумматор *VA*3 для четвертого столбца (вес 2^3) подключен так, как изображено на рисунке. Выход переноса поступает в регистр результата.

Параллельный сумматор для сложения двух 8-битовых чисел состоит из одного полусумматора и семи полных сумматоров.

14.4. Последовательный сумматор

При последовательном сложении столбцы двоичных чисел складываются по очереди. Сначала происходит сложение в столбце с самым низким весом (крайний столбец справа). Затем происходит сложение в столбце с боль-

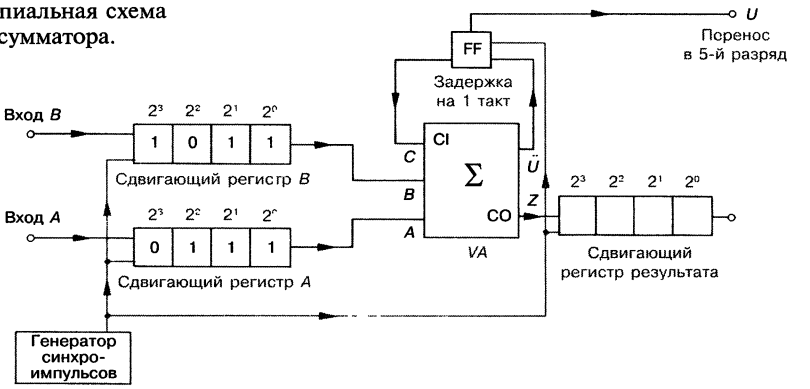


Рис. 14.11. Принципиальная схема последовательного сумматора.

шим весом. Затем складывается столбец со следующим весом и так далее, пока не будут сложены все столбцы. Перенос из результата сложения предыдущего столбца переходит в сложение, производимое в текущий момент. Методика последовательного сложения идентична выполнению сложения двух двоичных чисел от руки в столбик.

Принципиальная схема последовательного сумматора показана на рис. 14.11. Первое складываемое число, т. е. первое слагаемое, сохранено в сдвигающем регистре *A*. Второе складываемое число, т. е. второе слагаемое, сохранено в сдвигающем регистре *B*. Последовательные выходы сдвиговых регистров выведены на полный сумматор. *Z*-выход полного сумматора выдает сигналы результата, которые поступают в результирующий сдвигающий регистр. Сигнал, который находится на выходе переноса полного сумматора, задерживается на такт и прибавляется при следующем сложении столбцов. Задержка на такт обеспечивается MASTER-SLAVE-триггером *FF*.

С 1-м тактом на полный сумматор подаются, например, два 1-сигнала весом 2^0 согласно содержанию регистров *A* и *B* на рис. 14.11. На выходе *Z* появляется 0, на выходе *U* появляется 1. *Z*-сигнал переходит в результирующий сдвигающий регистр. *U*-сигнал записывается в *MS*-триггер *FF*. Со вторым тактом подаются два сигнала с весом 2^1 и сигнал переноса на входы полного сумматора (например $A = 1$, $B = 1$, $C = 1$). *Z*-сигнал (например $Z = 1$) сохраняется в результирующем сдвигающем регистре. *U*-сигнал (например $U = 1$) принимается триггером *FF*.

С 3-м тактом складывается третий столбец с весом 2^2. Затем следует сложение четвертого столбца на 4-м такте. Последовательное сложение закончено. Четыре бита результата с весами 2^0, 2^1, 2^2, 2^3 занесены в результирующий сдвигающий регистр. Пятый бит результата с весом 2^4 находится в триггере *FF* и может быть считан оттуда. Следующие такты не оказывают влияния на работу схемы.

> Последовательное сложение требует больше времени, чем параллельное сложение.

Результирующий сдвигающий регистр можно не ставить. Сдвигающий регистр *A* (или сдвигающий регистр *B*) может принимать данные со сдвигающего регистра результата. Сдвигающий регистр *A* будет во время сложения в столбик тактироваться вхолостую. Сигналы выхода результата *Z* пол-

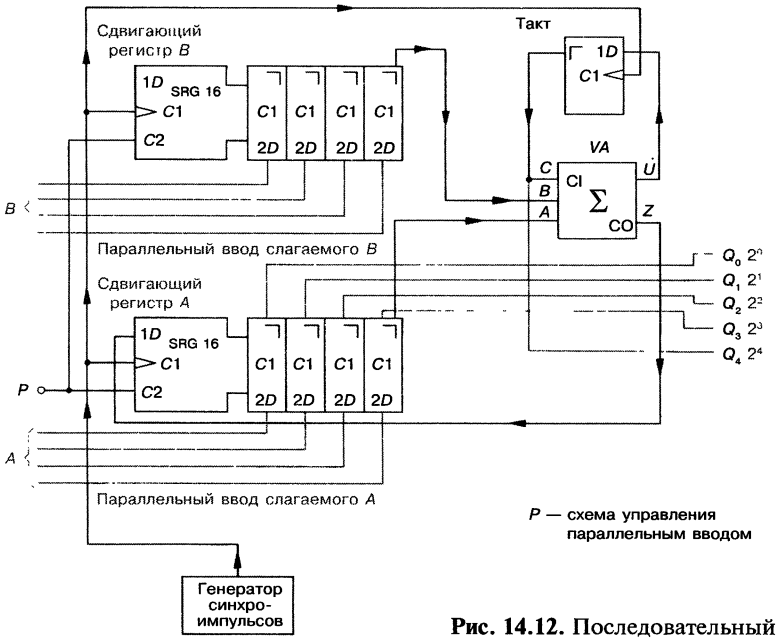


Рис. 14.12. Последовательный 4-битовый сумматор.

ного сумматора могут подаваться на вход сдвигающего регистра A и там сохраняться. После завершения сложения их можно оттуда считать.

Составленный по этим рассуждениям последовательный сумматор изображен на рис. 14.12. Два сдвигающих регистра построены на D-триггерах. Они имеют параллельный ввод данных, т. е. слагаемые A и B вводятся параллельно. Параллельный ввод разрешен, если на входе P действует 1-сигнал. Тогда через входы C_2 сдвигающих регистров разрешается ввод через входы D_2-триггеров сдвигающего регистра. При $P = 0$ параллельный ввод заблокирован. Дальнейший ввод сохраненной информации происходит через C_1-входы. Сдвигающий регистр A работает одновременно как результирующий сдвигающий регистр. Результат сложения находится после завершения сложения на выходах от Q_0 до Q_4.

14.5. Схемы вычитания

Схемы вычитания могут строиться аналогично схемам сложения. В таблицах истинности сопоставляются возможные входные сигналы и затем желаемые выходные сигналы. По правилам синтеза схем проектируется нужная схема.

Вычитание можно производить на базе сложения (см. гл. 8, подразд. 8.2.6.2). Состоящая из полных сумматоров схема сложения может быть преобразована путем небольших изменений в схему вычитателя.

14.5.1. Полувычитатель

Очень простым вычитателем является **полувычитатель**.

Полувычитатель может вычитать одно двоичное число из другого.

Соблюдаются следующие правила счета:

$$0 - 0 = 0;$$

$$0 - 1 = -1;$$

$$1 - 0 = 1;$$

$$1 - 1 = 0.$$

Двоичное число, из которого вычитается другое двоичное число (уменьшаемое), обозначим переменной A.

Вычитаемое — переменной B. Цифра 0 будет соответствовать бинарному состоянию 0, цифра 1 — бинарному состоянию 1.

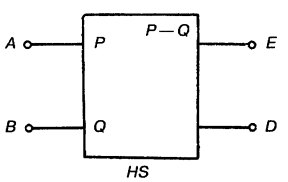
Полувычитатель имеет, следовательно, два входа A и B и один выход результата D. Результат (−1) представляет некоторую трудность. Поэтому предусмотрен второй выход. При результате вычитания (−1) на этом втором выходе появляется дополнительно сигнал 1. Второй выход получает обозначение E. Он также называется выходом займа.

Если в результате вычитания получается (−1), то $D = 1$ и $E = 1$. Соответствующая этому утверждению таблица истинности показана на рис. 14.13. Согласно таблице истинности на рис. 14.13 можно записать следующие уравнения:

$$D = \left(\bar{A} \wedge B\right) \vee \left(A \wedge \bar{B}\right);$$

$$E = \bar{A} \wedge B.$$

Уравнения приводят к схеме, рис. 14.14. 1-сигнал на выходе E, кроме идентификации (−1), служит прежде всего для многоразрядного вычитания, т. е. для вычитания многоразрядных двоичных чисел, осуществляя так называемый заем.



P + вход уменьшаемого
Q вход вычитаемого

$D = A \cdot B$

D выход для разности
E выход заема
HS полувычитатель

Вар.	A	B	E	D
1	0	0	0	0
2	0	1	1	1
3	1	0	0	1
4	1	1	0	0

Рис. 14.13. Полувычитатель и его таблица истинности.

A — уменьшаемое
B — вычитаемое
E — заем
D — разность(результат)

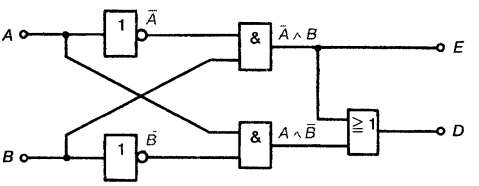


Рис. 14.14. Принципиальная схема полувычитателя.

14.5.2. Полный вычитатель

Полный вычитатель необходим для многоразрядного вычитания.

Пример

	2^4	2^3	2^2	2^1	2^0	
	1	1	0	1	1	27
	−1	0	0	1	0	− 22
Заем →		1				
	0	0	1	0	1	5

Вычитание в 3-м столбце (вес 2^2) требует займа. Полувычитатель будет выдавать $D = 1$ и $E = 1$. 1-сигнал с E-выхода должен в следующем столбце (вес 2^3) прибавиться к вычитаемому числу. Для этого требуется полный вычитатель.

> **Полный вычитатель** является схемой, которая может к значению вычитаемого прибавить сигнал займа (1-сигнал) и такое увеличенное вычитаемое вычесть из уменьшаемого.

Полный вычитатель должен иметь три входа (рис. 14.15). На вход A подается уменьшаемое число. На вход B подается вычитаемое число. На входе E находится сигнал займа от предыдущего столбца. Цифры в E_X и B складываются. Сумма вычитается из A. Разница дает в итоге D.

$$D = A - (B + E_X).$$

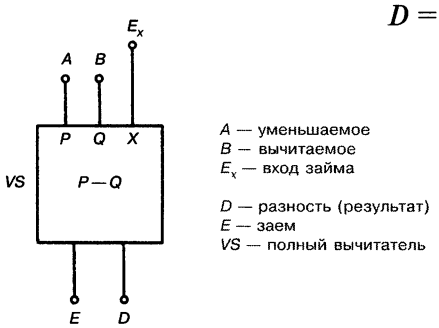


Рис. 14.15. Полный вычитатель.

Если требуется новый заем, на E появляется 1-сигнал.

Полный вычитатель можно собрать из полусумматора и полувычитателя. Полусумматор *HA* на рис. 14.16 складывает цифры в B и E_X в общее вычитаемое Z. Если получается перенос, он передается на выход E.

Полувычитатель *HS* вычитает Z из A. То есть он вычисляет $A - Z$. Если будет необходим заем, то на выходе E_1 появится 1-сигнал. Он передастся через элемент ИЛИ на E-выход (рис. 14.16).

Полный вычитатель может строиться также на двух полувычитателях (рис. 14.17). В полувычитателе HS_1 сначала образуется разность $A - B$. От этого результата вычитается заем в предыдущем столбце полувычитателем HS_2. При осуществлении займов как при вычитании в HS_1, так и при вычитании в HS_2 на выходе E появляется сигнал 1.

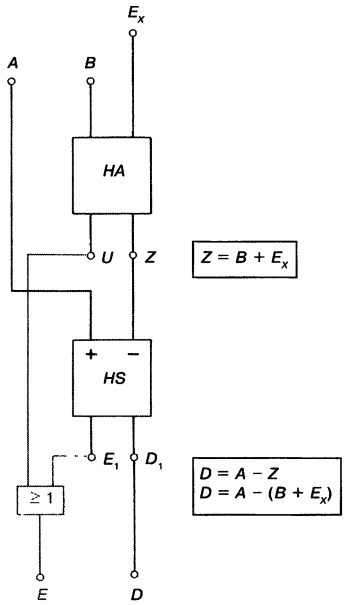


Рис. 14.16. Полный вычитатель, построенный на полусумматоре и полувычитателе.

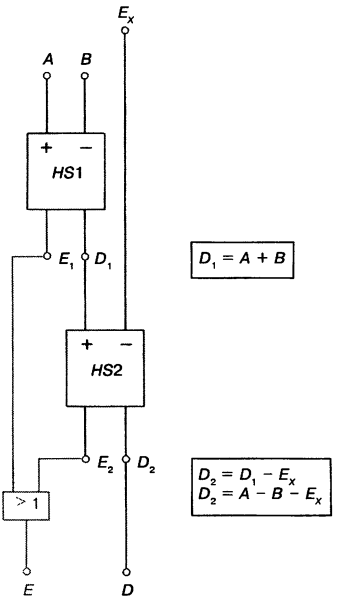


Рис. 14.17. Полный вычитатель, построенный на двух полувычитателях.

14.5.3. 4-битовый вычитатель

4-битовый вычитатель может вычитать из четырехзначного двоичного числа максимально четырехзначное двоичное число.

Схема состоит из трех полных вычитателей и полувычитателя (рис. 14.18).

Полувычитатель *HS* вычитает цифру наименьшего разряда вычитаемого от цифры наименьшего разряда уменьшаемого. Если будет необходим заем, то $E = 1$. Заем будет учтен при вычитании цифр с весом 2^1. Будет вычтено

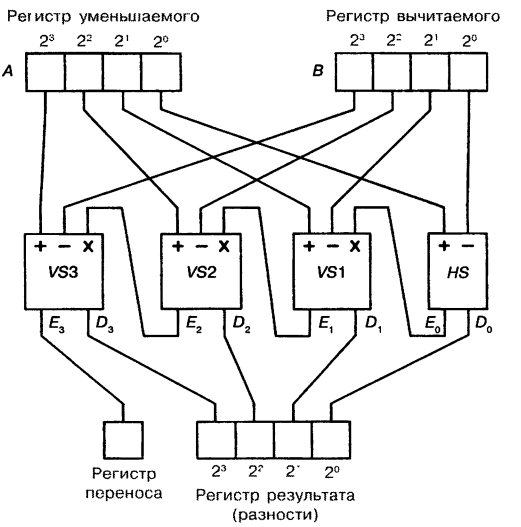


Рис. 14.18. 4-битовый вычитатель (параллельный).

число на 1 больше. Если заем будет необходим снова, то на E опять появится 1-сигнал. При вычитании цифр с весом 2^2 этот обновленный заем учитывается тем, что снова вычитается число на 1 больше. То же самое происходит при вычитании цифр с весом 2^3.

Если при последнем вычитании будет необходим заем, то в регистре переноса появится 1. Это значит, что вычитаемое больше, чем уменьшаемое. Результат является отрицательным числом. Отрицательное число представляется неправильно.

> Если вычитаемое больше, чем уменьшаемое, то возникающее отрицательное число представляется в выходном регистре неправильно.

Содержимое регистра должно дополняться. Организация дополнения разъясняется подробно в гл. 8.

> В случае отрицательного результата его инвертируют и прибавляют 1.

14.5.4. Вычитатель на полных сумматорах

В гл. 8, подразд. 8.2.6.2 показано, что вычитание двоичных чисел может быть произведено путем прибавления дополнения вычитаемого двоичного числа. 4-битовый вычитатель можно построить также и на базе 4-битового сумматора. Принцип такого вычитателя показан на рис. 14.19.

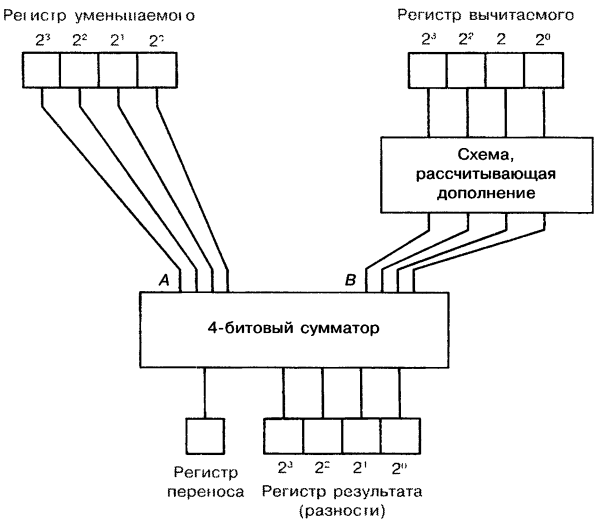


Рис. 14.19. Принцип действия вычитателя.

4-битовый сумматор — это параллельный сумматор согласно разд. 14.3. Схема получения дополнения должна инвертировать отдельные биты вычитаемого и прибавлять 1.

4-битовый параллельный сумматор может строиться на 4 полных сумматорах. Тогда просто необходимо прибавлять 1. Полному сумматору для сложения цифр с весом 2^0 нужно только два входа. На третий вход может подаваться 1, которая прибавляется к инвертированному вычитаемому. Для

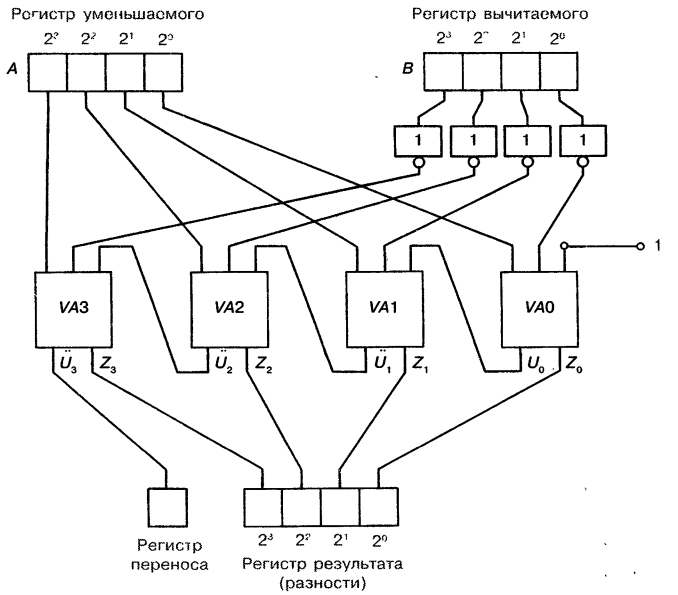


Рис. 14.20. 4-битовый вычитатель на базе полных сумматоров.

инвертирования вычитаемого используются только 4 элемента НЕ. Такая схема представлена на рис. 14.20.

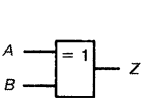
14.6. Универсальный сумматор-вычитатель

Рассмотренный в предыдущем разделе 4-битовый вычитатель на полных сумматорах может быть легко изменен так, чтобы по выбору работать либо как сумматор, либо как вычитатель. Для использования в качестве сумматора необходимо только выполнять два условия:

1. Инверсия вычитаемого должна быть отменена.
2. Не должно происходить прибавление 1 на входе C полного сумматора $VA1$.

Инверторы заменяются ИСКЛЮЧАЮЩИМИ ИЛИ (рис. 14.21). B-вход используется для управления. При $B = 0$ не происходит инвертирования, при $B = 1$ происходит. Получившийся сумматор-вычитатель представлен на рис. 14.22. Если на вход управления S поступает 0-сигнал, то проводится сложение $Z = A + B$. Если на вход управления S поступает 0-сигнал, схема работает как вычитатель. Вычисляется разность $Z = A - B$.

4-битовый сумматор-вычитатель можно сделать еще более универсальным. Если присоединить выходы A-регистра к ИСКЛЮЧАЮЩИМ ИЛИ, при соответствующем управлении можно считать также $B - A$. Если сделать к тому же еще выходы A-регистра и B-регистра выборочно запираемыми через элемент И, то получается еще больше возможностей. Можно, например, преобразовать A в $-A$.



Вар.	B	A	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

A инвертируется, если B = 1

Рис. 14.21. Условное обозначение и таблица истинности ИСКЛЮЧАЮЩЕГО ИЛИ.

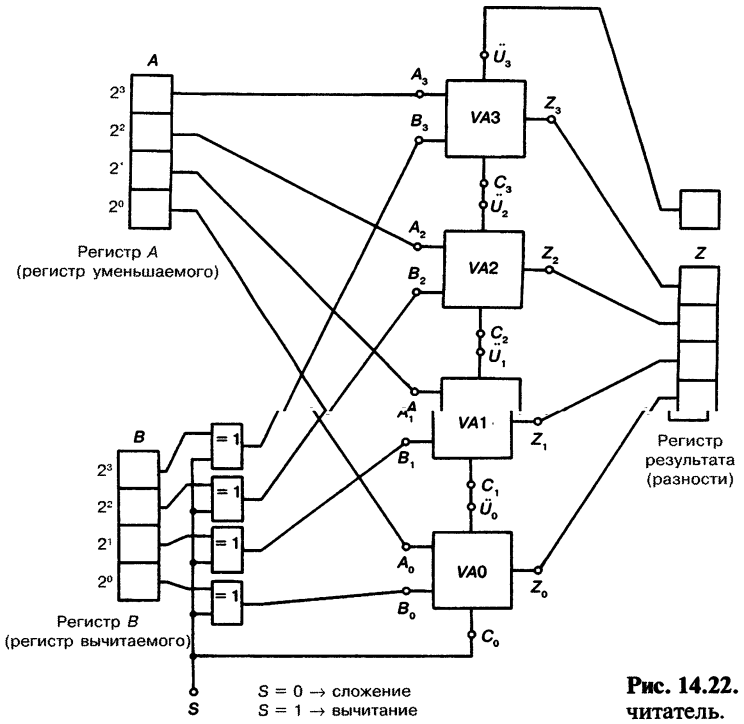


Рис. 14.22. 4-битовый сумматор-вычитатель.

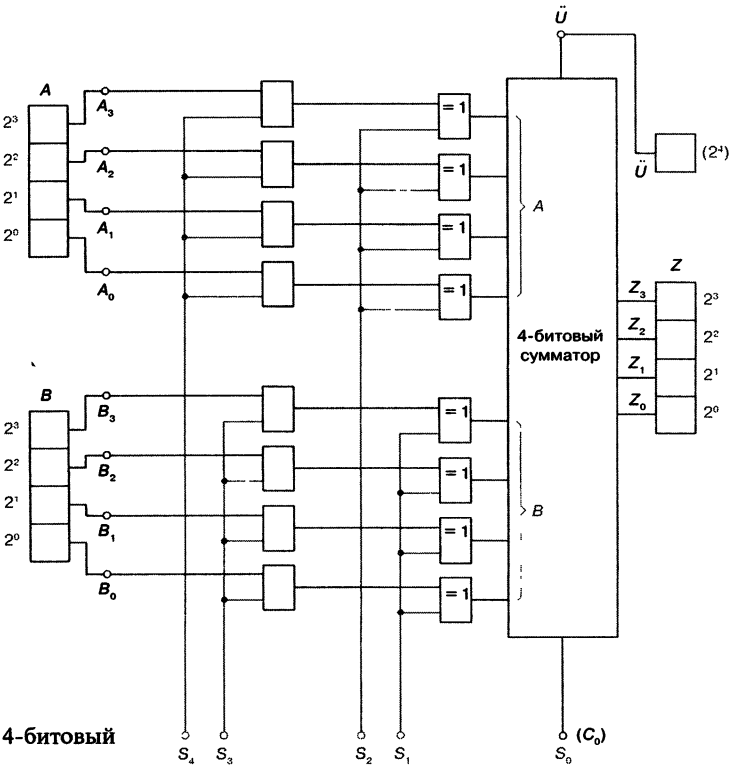


Рис. 14.23. Расширенный 4-битовый сумматор-вычитатель.

Рис. 14.24. Возможности управления 4-битового сумматора-вычитателя схемы рис. 14.23.

Вар.	Команды					Функции
№	S_4	S_3	S_2	S_1	S_0	
1	0	0	0	0	0	0
2	0	0	0	0	1	1
3	0	0	0	1	0	−1
4	0	0	0	1	1	0
5	0	0	1	0	0	−1
6	0	0	1	0	1	0
7	0	0	1	1	0	−2
8	0	0	1	1	1	−1
9	0	1	0	0	0	B
10	0	1	0	0	1	$B+1$
11	0	1	0	1	0	$-B-1=\bar{B}$
12	0	1	0	1	1	$-B$
13	0	1	1	0	0	$B-1$
14	0	1	1	0	1	B
15	0	1	1	1	0	$-B-2$
16	0	1	1	1	1	$-B-1=\bar{B}$
17	1	0	0	0	0	A
18	1	0	0	0	1	$A+1$
19	1	0	0	1	0	$A-1$
20	1	0	0	1	1	A
21	1	0	1	0	0	$-A-1=\bar{A}$
22	1	0	1	0	1	$-A$
23	1	0	1	1	0	$-A-2$
24	1	0	1	1	1	$-A-1=\bar{A}$
25	1	1	0	0	0	$A+B$
26	1	1	0	0	1	$A+B+1$
27	1	1	0	1	0	$A-B-1$
28	1	1	0	1	1	$A-B$
29	1	1	1	0	0	$B-A-1$
30	1	1	1	0	1	$B-1$
31	1	1	1	1	0	$-A-B-2$
32	1	1	1	1	1	$-A-B-1$

Расширенный 4-битовый сумматор-вычитатель показан на рис. 14.23. Четыре полных сумматора собраны в блок 4-битового параллельного сумматора. Входы от S_0 до S_4 являются управляющими входами.

Пять управляющих входов дают в итоге 32 различные возможности управления. Они представлены на рис. 14.24. Каждый 5-битовый блок может пониматься как 5-битовое управляющее слово.

Команда на сложение содержимого регистра A с содержимым регистра B выглядит как 11000.

Если $S_4 = 1$, содержимое регистра A пропускается. При $S_2 = 1$ содержимое регистра A инвертируется. Команда 10100 означает, следовательно, $Z = \bar{A}$.

Если нужно получить дополнение к A, то подается команда 10101. Регистр A пропускается, инвертируется и складывается с 1. Схема выдает тогда $-A$, так как дополнение к A равно $-A$. Отрицательные двоичные числа рассмотрены в гл. 8, разд. 8.2.7.

Команда 00010 должна приводить согласно рис. 14.24 к выводу −1. Это не очень ясно по схеме. По этой команде блокируются регистры A и B. На выходах всех И-элементов действует 0-сигнал. Так как $S_1 = 1$, инвертируются четыре 0-сигнала И-элементов от B. B-входы 4-битового параллельного сумматора равны, следовательно, 1111, A-входы — 0000. Производится следующее сложение:

$$\begin{array}{lcccccc} A & \rightarrow & & 0 & 0 & 0 & 0 \\ B & \rightarrow & + & 1 & 1 & 1 & 1 \\ \hline Z & \rightarrow & & 1 & 1 & 1 & 1 \end{array}$$

Это значение должно рассматриваться не как 15, а как −1, так как оно является также дополнением от 0001. (Области определения положительного и отрицательного двоичного числа см. в разд. 8.2.7.)

Как выглядит команда для разности ($B - A$)? Содержание регистров A и B должно быть пропущено ($S_4 = 1$, $S_3 = 1$). Содержание A-регистра инвертируется ($S_2 = 1$). Должна быть прибавлена 1 ($S_0 = 1$). Команда будет 11101.

Расширенный 4-битовый сумматор-вычитатель может благодаря возможности вычитания и сложения использоваться для многих целей.

14.7. Умножители

Для умножения двоичных чисел действуют следующие правила:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

Если цифре 0 ставится в соответствие бинарное состояние 0, а цифре 1 — бинарное состояние 1, то получается таблица истинности согласно рис. 14.25. Это таблица истинности И-элемента. Базой схем умножения является И-элемент.

> И-элемент является 1-битовым умножителем.

Вар.	B	A	X
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

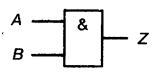


Рис. 14.25. Таблица истинности 1-битового умножителя (элемент И).

Умножение проводится за один шаг, т. е. параллельно. Для этого разработаны схемы параллельного умножения. Побитовое умножение, т. е. последовательно бит за битом, также возможно. Схемы последовательного умножения многоразрядных чисел дешевле схем параллельного умножения, но работают медленнее.

14.7.1. Параллельные умножители

При умножении используются понятия **множимое** и **множитель**. Множимое — это базовое число. Множитель является числом, на которое увеличивается множимое. В результате получается произведение.

Множимое		Множитель		Произведение
2	·	3	=	6

Для начала перемножим двухразрядные двоичные числа:

$$2 \cdot 3 = 6$$

$$\begin{array}{rl} \underline{10 \cdot 11} & \\ 10 & \rightarrow \text{1-е слагаемое} \\ \underline{10} & \rightarrow \underline{\text{2-е слагаемое}} \\ 110 & \rightarrow \text{Результат} \end{array}$$

Первое слагаемое получается из двух 1-битовых умножителей ($1 \cdot 0$, $1 \cdot 1$). Для этого требуются два И-элемента. Второе слагаемое получается также после двух 1-битовых умножителей ($1 \cdot 0$, $1 \cdot 1$), т. е. также двух И-элемен-

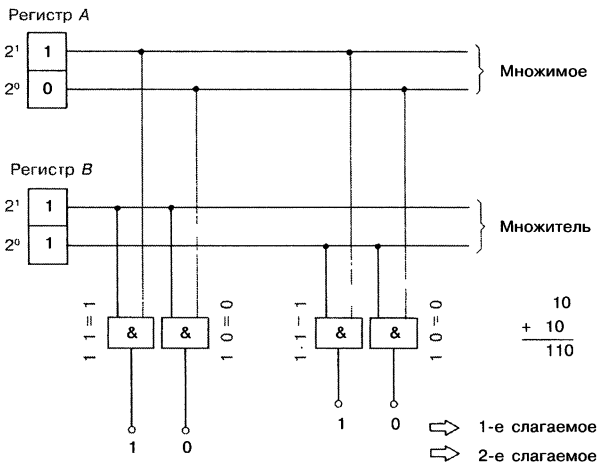


Рис. 14.26. Схема умножителя для получения слагаемых.

тов. Множимое и множитель находятся в двух регистрах (рис. 14.26). На выходах И-элементов находятся слагаемые.

Оба слагаемых должны теперь складываться поразрядно. Сложение происходит в 2-битовом параллельном сумматоре. И-элементы должны быть включены таким образом, чтобы второе слагаемое прибавлялось к первому слагаемому со смещением на одну позицию влево (рис. 14.27).

Для умножения больше чем 2-разрядных двоичных чисел схему нужно дополнить соответственно рис. 14.27. Если требуется перемножить два 4-разрядных двоичных числа, требуются 16 И-элементов, так как должно быть выполнено 16-битовое умножение

$$9 \cdot 11 = 99$$

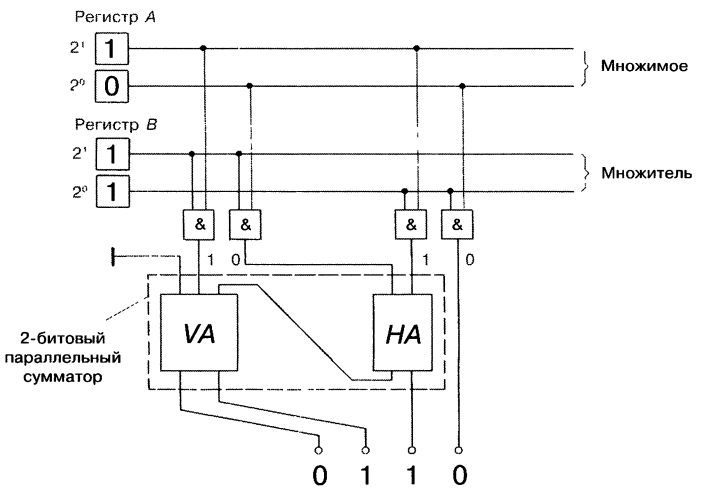


Рис. 14.27. 2-битовый параллельный умножитель.

Пример

```
                1 0 0 1 · 1 0 1 1
                ---------
                1 0 0 1          →  1-е слагаемое
              1 0 0 1            →  2-е слагаемое
            0 0 0 0              →  3-е слагаемое
          1 0 0 1                →  4-е слагаемое
        -------------
        1 1 0 0 0 1 1            →  Результат
```

Получается четыре 4-разрядных слагаемых, которые нужно складывать поразрядно. 4-битовый параллельный умножитель представлен на рис. 14.28.

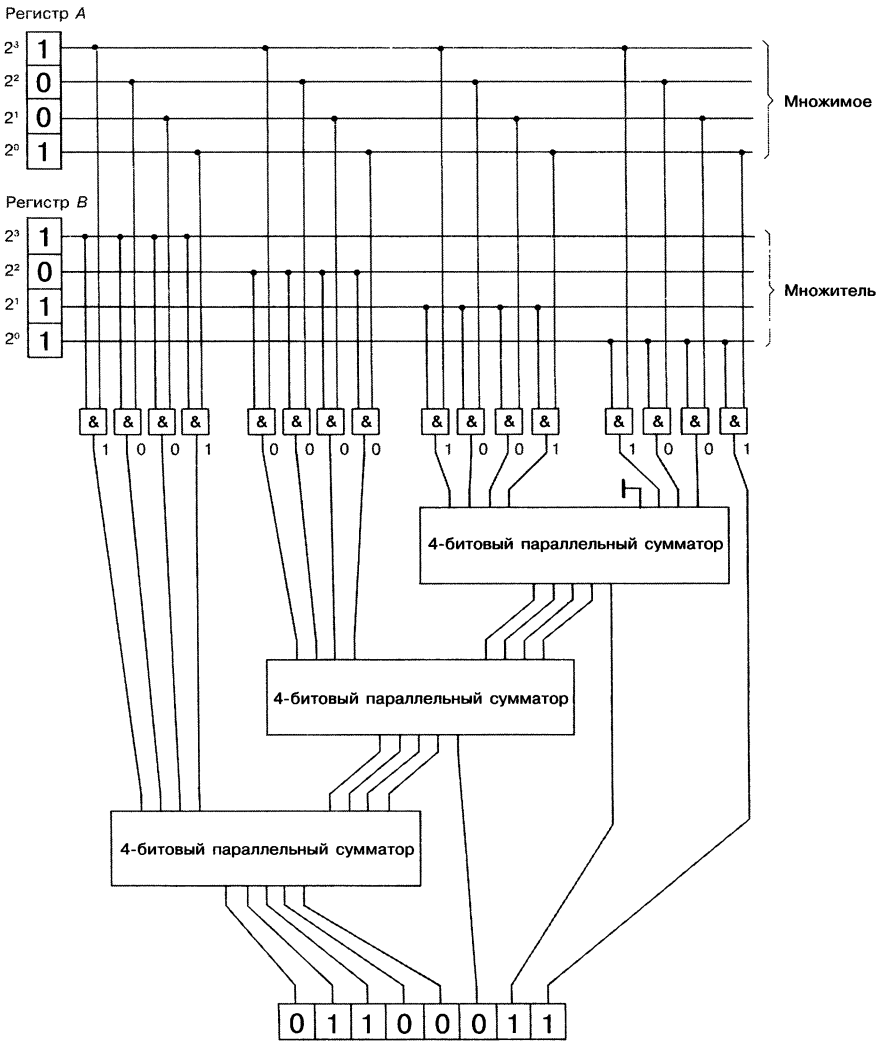


Рис. 14.28. 4-битовый параллельный умножитель.

Эта схема является относительно дорогой. С ростом разрядности чисел стоимость резко возрастает. Для умножения двух 8-битовых чисел требуются 64 И-элемента и восемь 8-битовых параллельных сумматоров.

14.7.2. Последовательный умножитель

Структура последовательного умножителя представлена на рис. 14.29. Схема предназначена для умножения 4-разрядных двоичных чисел. Множимое и множитель находятся каждый в своем регистре. Собственно умножение происходит посредством И-элемента, который работает как 1-битовый умножитель. 1-битовый полный сумматор прибавляет результат умножения поразрядно к уже полученному промежуточному результату, который находится в регистре результата.

Перед началом умножения регистр результата *E* пуст. Разряд с наименьшим весом (2^0) подается на нижний вход И-элемента. Этот разряд содержит 1 на рис. 14.29. С этой 1 перемножаются по очереди разряды множимого, начиная с разряда 2^0. Содержание регистра множимого (*A*) тактируется дальше сигналом синхронизации T_1. Отдельные цифры поступают по очереди в верхний вход обозначенного серым И-элемента и перемножаются. Регистр множимого включен по схеме кольцевого регистра. После четырех тактов в регистре множимого восстанавливается первоначальное положение цифр. Первое слагаемое (1001) получено и находится в регистре результата.

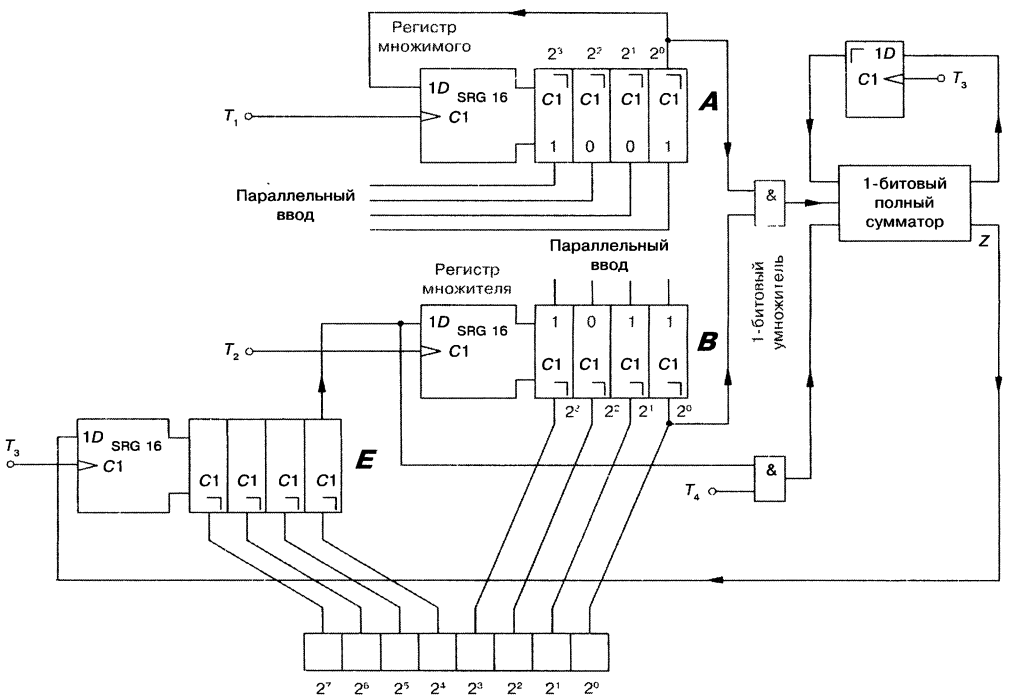
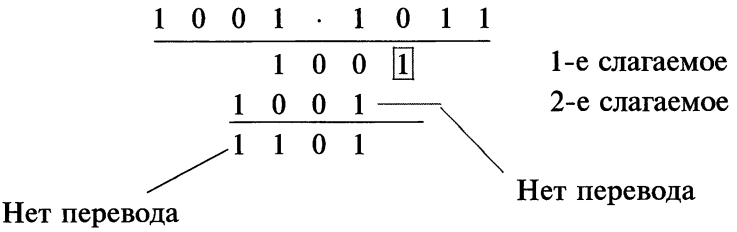


Рис. 14.29. Структура последовательного 4-битового умножителя.

Пример



Теперь подается такт T_2. Содержимое регистра множителя (*B*) перемещается на один разряд вправо. Затем происходит умножение множимого на 2-й разряд множителя (2^1) за 4 такта. Одновременно подается такт T_3 на регистр результата *E*. Содержимое регистра результата перемещается на разряд вправо. Разряд регистра результата с наименьшим весом переходит в регистр *B*.

Теперь происходит умножение множимого со 2-м разрядом множителя (2^1) за 4 такта. Получившийся результат умножения прибавляется к содержимому регистра результата. Полученное 2-е слагаемое прибавляется к уже найденному 1-му слагаемому. При этом разряд с наименьшим весом 1-го множителя, который находится *B*-регистре, не записывается.

Следующими тактами T_2 и T_3 множитель перемещается на один разряд вправо, затем на следующих тактах содержимое наименьшего разряда *E*-регистра подается в *B*-регистр.

Затем происходит умножение множимого с 3-м разрядом множителя (2^2) за 4 такта. Получившийся результат умножения прибавляется к регистру результата.

Процесс продолжается аналогично выше изложенному. Множитель и содержание регистра результата снова сдвигаются на 1 разряд. Наименьший разряд *E*-регистра переходит в *B*-регистр. Умножение множимого с 4-м разрядом множителя (2^3) за 4 такта. Получившийся результат умножения прибавляется к регистру результата.

Следующими тактами T_2 и T_3 содержимое регистров *B* и *E* сдвигается на один разряд вправо, и одновременно содержимое наименьшего разряда регистра *E* подается в *B*-регистр.

Умножение завершено. Конечный результат находится в регистрах *E* и *B* и может выдаваться на выходы. Вес поразрядно показан на рис. 14.29.

Контрольный тест

1. Изобразите таблицу истинности полусумматора и разработайте по таблице истинности схему на базе основных логических элементов.
2. Чем отличается полный сумматор от полусумматора?
3. Из двух полусумматоров и ИЛИ-элемента соберите полный сумматор.
4. Объясните понятие 8-битовый параллельный сумматор.
5. Как построен последовательный сумматор?
6. Нарисуйте схему полувычитателя и объясните принцип его работы.

7. На трех полных сумматорах и любых логических элементах спроектируйте 3-битовый сумматор-вычитатель. Нарисуйте схему.

8. Какие операции должен производить 1-битовый умножитель? Изобразите таблицу истинности 1-битового умножителя.

9. Как работает 3-битовый параллельный умножитель? Объясните принцип действия. Сколько требуется 1-битовых умножителей и схем сложения? Какие должны быть схемы сложения?

10. Объясните принцип действия последовательного умножителя.

ГЛАВА 15

МИКРОПРОЦЕССОРЫ И МИКРОКОМПЬЮТЕРЫ

15.1. Микропроцессор как универсальная схема

Можно ли спроектировать схему, которая умеет складывать, вычитать, умножать и выполнять любые логические операции с бинарными сигналами? Введенные сигналы, также называемые входными данными, должны быть в любое время обработаны по заданному алгоритму. Порядок обработки, например порядок логических операций, задается перед началом работы схемы в программе.

Такая схема была бы универсальной в применении. Она смогла бы заменять логические схемы любого вида. Не надо было бы каждый раз паять необходимую комбинационную схему из различных элементов. Можно было бы брать такую универсальную схему и программировать ее таким образом, чтобы выполнялись желаемые логические функции.

Структура этой универсальной схемы сложна, и схема, очевидно, получилась бы дорогой. Однако современные технологии изготовления микросхем дают возможность экономически выгодно изготовлять сложные схемы.

Соображения такого рода стояли на заре проектирования первых универсальных схем, которые сегодня называются **микрокомпьютерами**. Основной частью микрокомпьютера является **микропроцессор**. Микропроцессоры различных типов предлагаются в настоящее время в виде микросхем относительно недорого. Сложные схемы управления, довольно дорогие в случае их сборки из триггеров и логических элементов, можно заменить недорогими микрокомпьютерами.

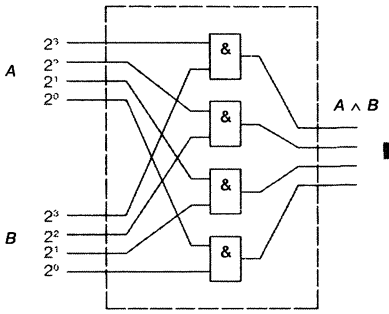
15.2. Арифметико-логическое устройство (АЛУ)

При разработке универсальных схем целесообразно брать за основу расширенные 4-битовые сумматоры-вычитатели (см. рис. 14.23), которые были рассмотрены в предыдущей главе. С этой схемой входные сигналы A и B по выбору могут складываться и вычитаться.

Дополнительно требуется, чтобы с сигналами A и B можно было производить операции:

- логического умножения И;
- логического сложения ИЛИ;
- ИСКЛЮЧАЮЩЕГО ИЛИ.

Схема для проведения 4-битового И-умножения показана на рис. 15.1. Также изображены схемы для 4-битовых ИЛИ, и 4-битовых ИСКЛЮЧАЮЩИХ ИЛИ, логических функций (рис. 15.2).



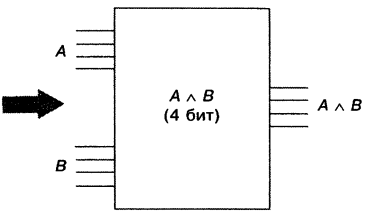


Рис. 15.1. Схема для логического умножения И двух 4-битовых слов.

Пример

A	1011
B	1101
$A \wedge B$	1001

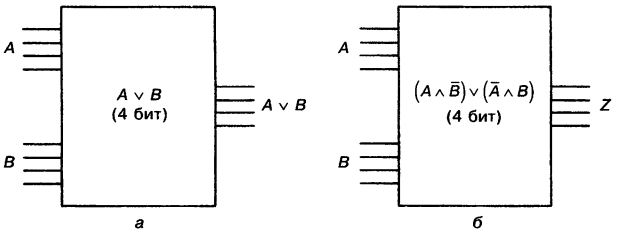


Рис. 15.2. Схема для логического умножения ИЛИ (*а*) двух 4-битовых слов и 4-битового ИСКЛЮЧАЮЩЕГО ИЛИ (*б*).

Схема, которая может складывать и вычитать два N-битовых слова, а также проводить над ними операции логического умножения И, логического сложения ИЛИ и ИСКЛЮЧАЮЩЕГО ИЛИ, называется **арифметико-логическое устройство** (АЛУ).

АЛУ для 4-битовых слов состоит из расширенного сумматора-вычитателя согласно рис. 14.23, схемы для проведения 4-битового И-умножения, схемы для проведения 4-битового ИЛИ-сложения и схемы для проведения операции 4-битового ИСКЛЮЧАЮЩЕГО ИЛИ. Четыре 4-битовых выхода выборочно через четыре мультиплексора (см. гл. 11) подаются на 4-битовый Z-выход. Сумматор-вычитатель имеет также выход переноса U, который выводится наружу (рис. 15.3).

АЛУ производятся в виде микросхем на 4 бита, 6 битов, 8 битов и 16 битов. Чаще всего используются 8-битовые АЛУ. Условное обозначение — в виде блока (рис. 15.4). Так как 8-битовые АЛУ имеет 8 A-входов, 8 B-входов и 8 Z-выходов, то 8 проводников могут представляться 8-проводной шиной. Схемы получаются более наглядными (рис. 15.4).

Через семь входов управления от S_0 до S_6 могут подаваться всего $2^7 = 128$ различных команд управления. Из этих команд нужны только 13. В этом случае рационально произвести перекодировку. Она происходит с помощью ПЗУ (см. гл. 12, разд. 12.4). Используют 4 входа управления (рис. 15.5). С ними можно представлять 16 различных команд. 3 возможных команды остаются неиспользованными. Команды представлены на рис. 15.6. Несколько команд требуют подавления переноса U. Для этой цели ПЗУ имеет выход S_7, всегда равный нулю, если на выходе U нет переноса. Выходы S_8 и S_9 используются для дополнительного управления (см. разд. 15.3 и 15.4).

Блок-схема 8-битового АЛУ с кодировочным ПЗУ и блокированием переноса показана на рис. 15.7.

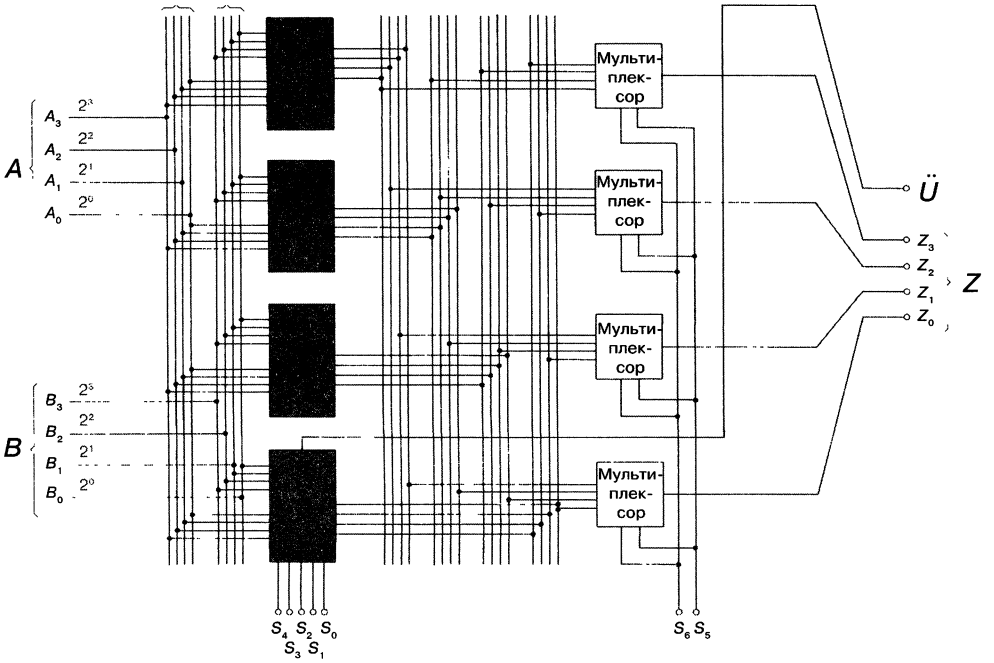


Рис. 15.3. Схема 4-битового АЛУ.

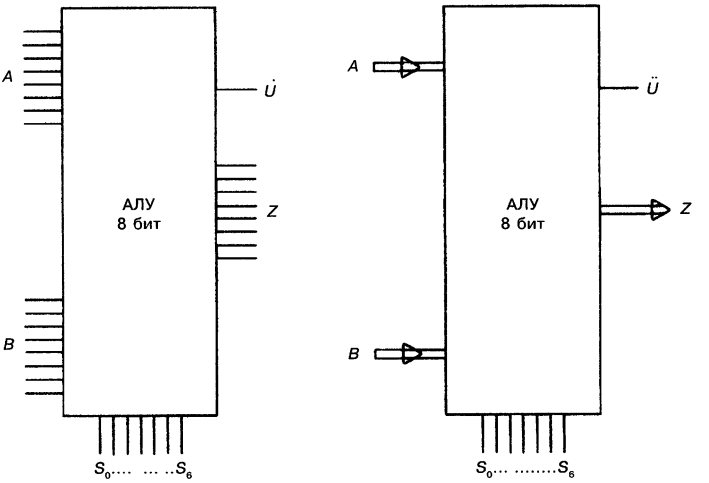


Рис. 15.4. Блок-схема 4-битового АЛУ.

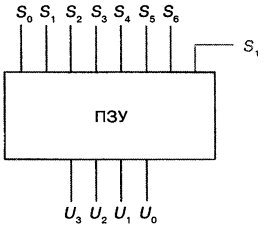


Рис. 15.5. Кодировщик с ПЗУ для перекодирования с 7 на 4 управляющих входа.

	Команды				Функции
	U_3	U_2	U_1	U_0	
1	0	0	0	0	A
2	0	0	0	1	1
3	0	0	1	0	$\bar{A}$
4	0	0	0	1	0
5	0	1	0	0	0
6	0	1	0	1	$A+1$
7	0	1	1	0	$A-1$
8	0	1	1	1	$A+B$
9	1	0	0	0	$A-B$
10	1	0	0	1	$A \wedge B$
11	1	0	1	0	$A \vee B$
12	1	0	1	1	$(A \wedge \bar{B}) \vee (\bar{A} \wedge B)$
13	1	1	0	0	−1
14	1	1	0	1	
15	1	1	1	0	
16	1	1	1	1	

Рис. 15.6. Команды АЛУ.

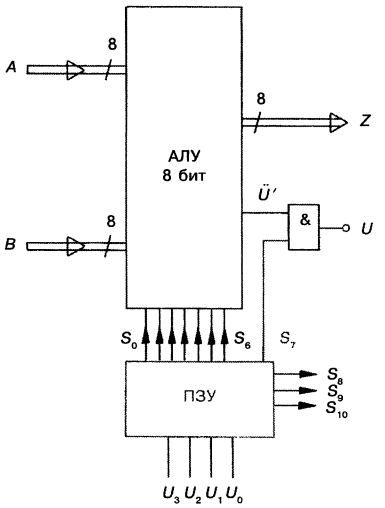


Рис. 15.7. АЛУ с ПЗУ для перекодирования и И-элемент для блокировки переноса.

15.3. Аккумулятор

Аккумулятор состоит из АЛУ с кодированием, регистра и 1-битовой памяти для переноса. Ввод данных происходит только через *B*-входы. *A*-входы связаны с выходами регистра (рис. 15.8). Регистр и память переноса управляются тактирующими синхроимпульсами. Тактирование памяти переноса может быть заблокировано И-элементом. S_8-сигнал поступает из кодирующего ПЗУ.

Если складываются два 8-битовых слова, первое 8-битовое слово подается на *B*-вход. Через АЛУ оно переходит в регистр и сохраняется в нем со следующим тактом. Регистр построен на 8 триггерах, в которых парал-

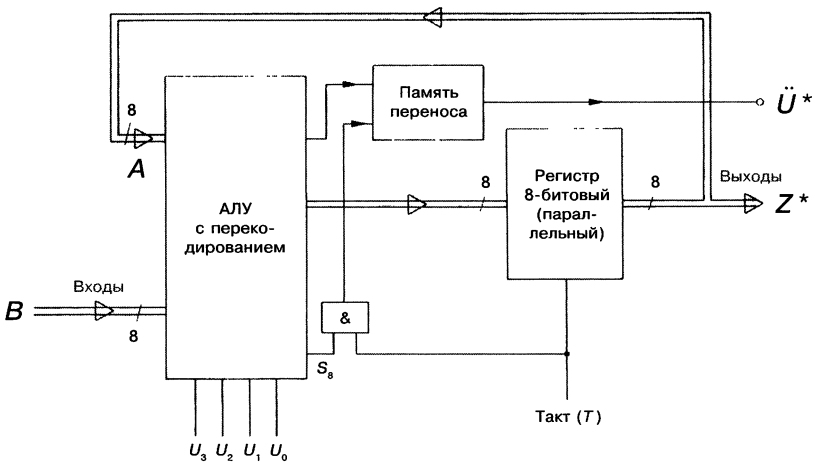


Рис. 15.8. Структура аккумулятора.

Номер команды		Команда				Название команды	Описание команды	Память переноса
десятич.	шестнадц.	U_3	U_2	U_1	U_0			
0	0	0	0	0	0	NOP	Нет операции	*T*
1	1	0	0	0	1	SP1	Записать 1 в аккумулятор	*T*
2	2	0	0	1	0	CMA	Инвертировать содержимое аккумулятора	—
3	3	0	0	1	1	LDA	Записать *B*-сигнал в аккумулятор	—
4	4	0	1	0	0	CLA	Стереть содержимое аккумулятора	—
5	5	0	1	0	1	INK	Увеличить содержимое аккумулятора на 1	*T*
6	6	0	1	1	0	DEC	Уменьшить содержимое аккумулятора на 1	*T*
7	7	0	1	1	1	ADD	Прибавить *B*-сигнал к содержимому аккумулятора	*T*
8	8	1	0	0	0	SUB	Вычесть *B*-сигнал из содержимого аккумулятора	*T*
9	9	1	0	0	1	AND	UND — логическое умножение И содержимого аккумулятора и *B*-сигнала	*T*
10	*A*	1	0	1	0	IOR	ODER — логическое сложение ИЛИ содержимого аккумулятора и *B*-сигнала аккумулятора и *B*-сигнала	*T*
11	*B*	1	0	1	1	XOR	Exklusiv-ODER — ИСКЛЮЧАЮЩЕГО ИЛИ содержимого	*T*
12	*C*	1	1	0	0	SM1	Записать в аккумулятор — 1	*T*
13	*D*	1	1	0	1	—	—	—
14	*E*	1	1	1	0	—	—	—
15	*F*	1	1	1	1	—	—	—

T : память переноса управляется синхроимпульсами
—: память переноса не управляется синхроимпульсами

Рис. 15.9. Команды аккумулятора.

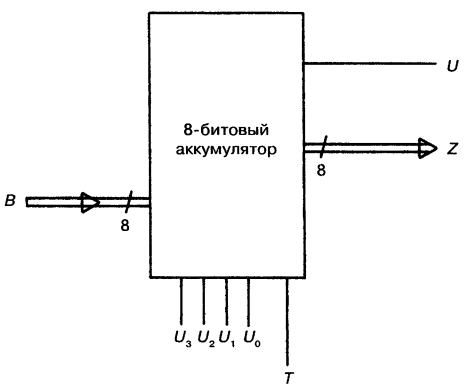


Рис. 15.10. Блок-схема 8-битового аккумулятора.

лельно сохраняется 8-битовое слово. Это слово поступает на 8 выходов регистра и одновременно находится на 8 *A*-входах.

После сохранения первого 8-битового слова на вход поступает второе 8-битовое слово. Два складываемых 8-битовых слова находятся на *A*- и *B*-входах АЛУ. АЛУ по команде складывает оба 8-битовых слова и выдает результат в регистр. Регистр принимает результат со следующим тактом. Возможно, возникший перенос будет также сохранен в регистре переноса со следующим синхроимпульсом. Регистр переноса называется также знаком переноса. Результат сложения находится теперь на Z^*-выходе и на выходе U^*.

Логическое умножение И двух 8-битовых слов происходит аналогичным образом. Только вместо команды сложения АЛУ получает команду на умножение.

Все 13 команд АЛУ распространяются также на аккумулятор. Под *A*-сигналом можно понимать содержание аккумулятора. Командам присваиваются сокращения от английских названий команд. Все команды аккумулятора приведены в так называемом списке команд (рис. 15.9).

15.4. Аккумулятор с памятью

Следующим шагом на пути к универсальной схеме, к так называемому микропроцессору, является **аккумулятор с памятью**. В качестве памяти используется статическое ОЗУ (см. гл. 12, разд. 12.3.1).

> Аккумулятор с памятью может записывать промежуточные результаты в память и считывать их оттуда при необходимости.

Структура аккумулятора с памятью показана на рис. 15.11. Восемь выходов аккумулятора выходят наружу (Z^*) и связаны с входами ОЗУ. Выходные данные аккумулятора могут записываться в ОЗУ. Загрузка в ОЗУ происходит под управлением синхроимпульсов. ПЗУ аккумулятора выдает на управляющий вывод S_9 1-сигнал и одновременно через элемент И выдает управляющий импульс, разрешающий запись.

ОЗУ может иметь различный объем памяти. На рис. 15.11 показана ОЗУ с 16 ячейками памяти по 8 бит. 16 ячеек памяти должны иметь свои адреса (см. гл. 12). С 4-битовым словом можно производить 16 различных адресов (от 0000 до 1111). Так как адреса имеют 4 разряда, нужно 4 адресных входа. Они получают обозначения от A_0 до A_3 (см. рис. 15.11).

Сохраненные в ОЗУ данные могут быть по желанию считаны и записаны в аккумулятор. *B*-входы аккумулятора управляются мультиплексором. Если мультилексор получает 1-сигнал через вход управления S_{10} ПЗУ аккумулятора, то он переключает B^*-входы на *B*-входы аккумулятора. Если на вход S_{10} управления подан 0-сигнал, то выходы ОЗУ связаны с B-входами аккумулятора.

К 13 командам аккумулятора без устройства памяти (рис. 15.9) прибавим две дополнительные команды. Нужна команда, которая переключает мультиплексор на входы B^*. Эта команда должна подаваться подачей 1-сигнала на вход управления S_{10}. Ей присваивается комбинация номер 13 (1101) (рис. 15.12). Название команды INP.

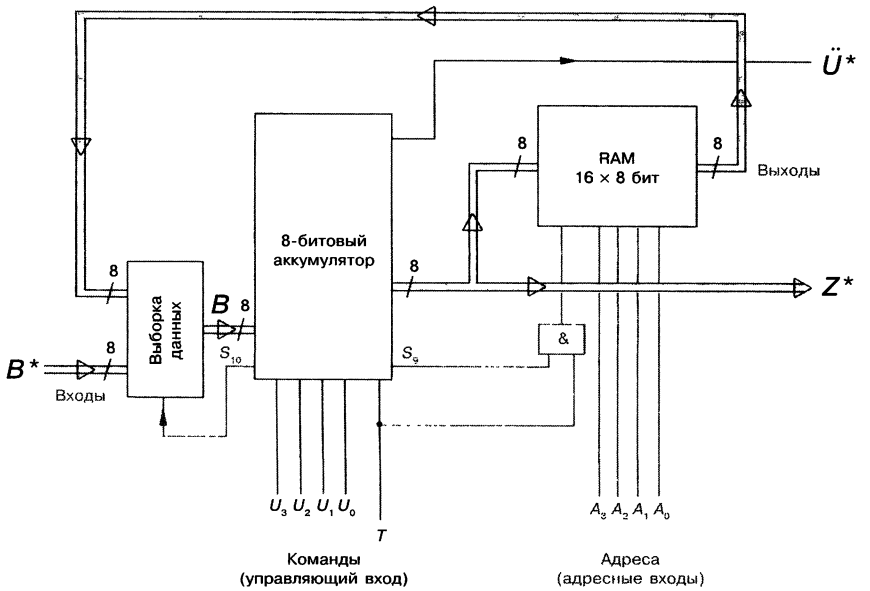


Рис. 15.11. Структура аккумулятора с памятью.

Номер команды		Команда				Название команды	Описание команды	Память переноса
десятич.	шестнадц.	U_3	U_2	U_1	U_0			
13	*D*	1	1	0	1	INP	Входной сигнал *B** загрузить в аккумулятор	—
14	*E*	1	1	1	0	STA	Содержимое аккумулятора загрузить в ячейку памяти по адресу $A_3A_2A_1A_0$	—

Рис. 15.12. Дополнительные команды аккумулятора с памятью.

Кроме того, необходима команда, которая разрешает запись содержимого аккумулятора в ячейку памяти ОЗУ. Эта команда получает номер 14 (1110) и имя STA. Она должна быть обработана в ПЗУ аккумулятора таким образом, чтобы на линии управления S_9 был выведен 1-сигнал, разрешающий очередной такт синхроимпульса к ОЗУ (см. рис. 15.11). Эта команда всегда должна быть связана с адресом.

В аккумуляторе с устройством памяти отдельные команды состоят из 8-битовых слов. Четыре бита отведены для сигнала управления (U_0 до U_3) и четыре — для адреса (от A_0 до A_3).

Биты управления образуют управляющую часть команды, биты адреса — адресную.

Перед началом работы, т. е. перед началом вычисления, должны быть точно установлены отдельные команды и порядок их следования.

Упорядоченная последовательность команд называется **программой**.

Нужно сообщить аккумулятору с памятью, что он должен делать. Сначала вводится команда 1, затем 2, затем 3 и т. д. до ввода всех команд и последующего запуска программы. В конце должен получиться желаемый результат в правильной форме. Если это не произошло, значит программа была ошибочна.

Перед каждым новым процессом вычисления команды должны вводиться заново. Это требует много времени и на практике едва ли выполнимо. Представьте себе управление станком, в котором перед каждой операцией нужно было бы вводить сорок 8-битовых команд с клавиатуры. Никто не будет покупать такую машину.

15.5. Программно-управляемый упрощенный компьютер

Аккумулятор с памятью можно значительно улучшить программным управлением. **Программное управление** состоит из памяти программ, устройства ввода и счетчика команд. После применения этих дополнительных модулей из аккумулятора с памятью получается упрощенный программно-управляемый компьютер (рис. 15.13).

Перед началом процесса загружается программа. Первая команда поступает на входы B^*. Импульсом на порт T_2 выдается сигнал загрузки. Команда будет записана в память программ. Тогда на входы B^* подается

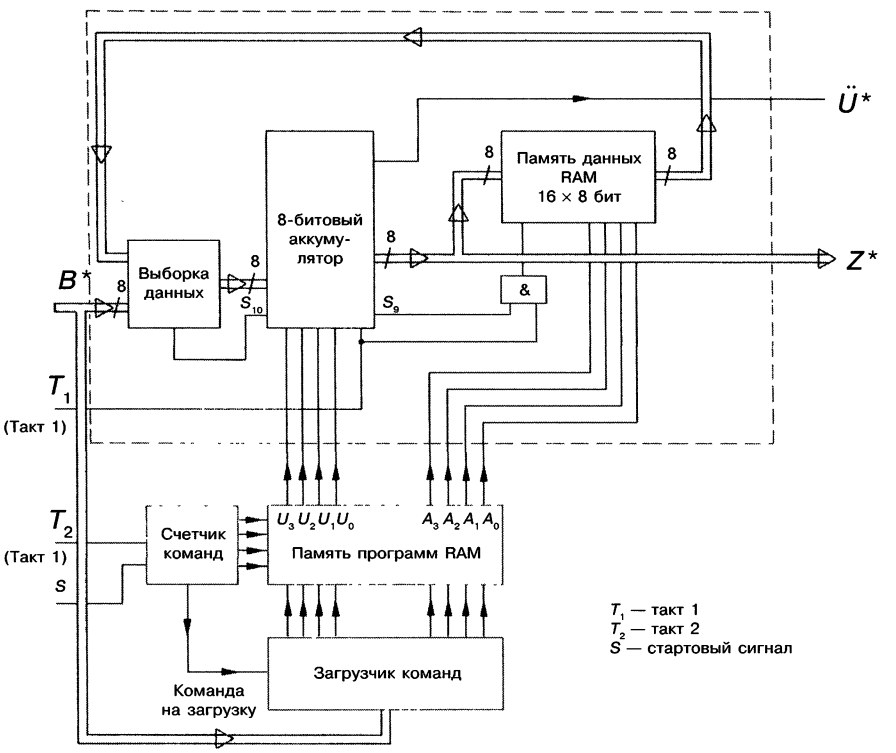


Рис. 15.13. Структура простого программируемого калькулятора.

Номер команды		Команда								Название команды	Описание команды
		Нет перевода				Нет перевода					
десятич.	шестн.	U_3	U_2	U_1	U_0	A_3	A_2	A_1	A_0		
0	0	0	0	0	0	—	—	—	—	NOP	Нет операции
1	1	0	0	0	1	—	—	—	—	SP1	Записать 1 в аккумулятор
2	2	0	0	1	0	—	—	—	—	CMA	Инвертировать содержимое аккумулятора
3	3	0	0	1	1	A	A	A	A	LDA	Содержимое ячейки памяти по адресу *AAAA* загрузить в аккумулятор
4	4	0	1	0	0	—	—	—	—	CLA	Стереть содержимое аккумулятора
5	5	0	1	0	1	—	—	—	—	INK	Увеличить содержимое аккумулятора на 1
6	6	0	1	1	0	—	—	—	—	DEC	Уменьшить содержимое аккумулятора на 1
7	7	0	1	1	1	A	A	A	A	ADD	Содержимое ячейки памяти по адресу *AAAA* прибавить к содержимому аккумулятора
8	8	1	0	0	0	A	A	A	A	SUB	Содержимое ячейки памяти по адресу *AAAA* вычесть из содержимого аккумулятора
9	9	1	0	0	1	A	A	A	A	AND	Логическое умножение содержимого ячейки памяти по адресу *AAAA* и содержимого аккумулятора
10	*A*	1	0	1	0	A	A	A	A	IOR	ODER — логическое сложение содержимого ячейки памяти по адресу *AAAA* и содержимого аккумулятора
11	*B*	1	0	1	1	A	A	A	A	XOR	Exklusiv-ODER — ИСКЛЮЧАЮЩЕЕ ИЛИ содержимого ячейки памяти по адресу *AAAA* и содержимого аккумулятора
12	*C*	1	1	0	0	—	—	—	—	SM1	Записать в аккумулятор — 1
13	*D*	1	1	0	1	—	—	—	—	INP	Входной сигнал *B** загрузить в аккумулятор
14	*E*	1	1	1	0	A	A	A	A	STA	Содержимое аккумулятора загрузить в ячейку памяти по адресу *AAAA*
15	*F*	1	1	1	1	—	—	—	—	HLT	Остановить счетчик команд

—: адресация не нужна
A: разряд адресного бита

Рис. 15.14. Список команд программируемого калькулятора (команды управляют аккумулятором с памятью данных за исключением команды 15).

2-я команда и таким же способом записывается. После 2-й команды следует 3-я и так далее. Команды загружаются в той последовательности, в которой их потом нужно запускать. При этом не должна возникать путаница.

Если все команды программы загружены в память программ, можно начинать их исполнение. Сигнал старта дается импульсом на вход S счетчика команд. Теперь счетчик команд вызывает первую команду из памяти программ. Команда подается на входы от U_0 до U_3 и от A_0 до A_3 и запускается. Затем счетчик команд вызывает вторую команду из памяти программ. После выполнения второй команды будет загружена и выполнена третья команда.

Так отрабатывается вся программа команда за командой. После выполнения всех команд счетчик команд должен быть остановлен. Для этого требуется следующая команда, так называемая команда останова (HLT). 4-битовая комбинация номер 15 (1111) еще свободна (см. рис. 15.9 и рис. 15.12). Эта комбинация выделяется команде HLT.

Программно-управляемый упрощенный компьютер располагает теперь 15 командами, которые представлены на рис. 15.14.

Программно-управляемый упрощенный компьютер позволяет проводить сложные процессы счета. Команды запускаются строго в последовательности ввода. Это не всегда желательно. Часто хотелось бы иметь передачу управления команды, т. е. возвращаться, например, после 35-й команды снова к 10-й команде и отработать промежуточные команды еще раз или после 20-й команды перейти к 45-й команде. Таких возможностей программно-управляемый упрощенный компьютер не предоставляет.

15.6. Микропроцессоры

Если усовершенствовать программно-управляемый компьютер (рис. 15.13) так, чтобы он был в состоянии проводить перемещения по программе, получится **микропроцессор**, который будет универсальным модулем для любых задач управления и расчетов. Ядро микропроцессора составляют АЛУ, счетчик команд, регистры и системы управления.

> Микропроцессором называется не полностью программно-управляемая счетная схема, которая предназначена для задач управления.

Основной задачей микропроцессоров является управление. Но для управления требуются также и расчеты.

15.6.1. Виды микропроцессоров

Существует много различных видов микропроцессоров с разными свойствами. В настоящее время производятся примерно 60 типов микропроцессоров. Все они состоят из одной-единственной микросхемы.

> Микропроцессоры производятся исключительно в виде микросхем.

Используются, как правило, DIP-корпуса с количеством выводов до 40. Микропроцессоры характеризуются следующими свойствами:

1. *Длина слова*

Длина слова указывает, сколько бит могут обрабатываться параллельно, т. е. количество бит входных и выходных данных. Существуют 4-битовые, 8-битовые, 16-битовые и 32-битовые микропроцессоры.

2. *Скорость вычислений*

При сравнении скорости вычислений сравнивают друг с другом так называемую длительность цикла обработки. Под длительностью цикла обработки понимают время, которое требуется для параллельного сложения двух двоичных чисел и для ввода—вывода этих чисел из/в память. Обычно цикл длится от 10 мкс до 0,1 мкс.

3. *Технология (семейство схем)*

Микропроцессоры производятся в основном в МОП-исполнении. В этой технологии возможна максимальная степень интеграции. Могут быть построены достаточно сложные схемы. Существует также небольшой ряд биполярных микропроцессоров, которые принадлежат к семейству ТТЛШ. Они работают очень быстро и имеют относительно простую структуру.

Среди МОП-процессоров основу составляют *N*-МОП процессоры. Они потребляют мощность от 0,5 до 1,5 Вт. Реже используют КМОП-тип, их преимуществом является крайне малое энергопотребление — от 1 до 5 мВт. Различают статические микропроцессоры и динамические микропроцессо-

Тип	Производитель	Длина слова в битах	Количество команд	Время цикла в мкс	Логическое семейство (технология)
4040	Intel	4	60	10,0	NMOS
8080 A	Intel/Simens	8	78	2,0	NMOS
8085	Intel/Simens	8	80	1,3	NMOS
IM 6100	Intersil	8	87	2,5	CMOS
M 6800	Motorola	8	72	2,0	NMOS
SCMP	Nat. Semic.	8	46	2,0	PMOS/NMOS
Z 80	Zilog	8	158	1,0	NMOS
8086	Intel	16	135	0,5	CMOS
68000	Motorola	16	56	0,6	CMOS/NMOS
TMS 9900	Texas Instr.	16	69	7,5	NMOS
80386	Intel	32	150	0,3	CMOS
68020	Motorola	32	100	0,4	CMOS/NMOS
32332	Nat. Semic.	32	130	0,3	NMOS
Pentium	Intel	64	205	0,06	NMOS/CMOS

Рис. 15.15. Таблица основных микропроцессоров.

ры. Статические микропроцессоры имеют статическое ОЗУ и не нуждаются в регенерации. Динамические микропроцессоры имеют динамическое ОЗУ. Они нуждаются в регенерации.

4. *Система команд*

Количество команд характеризует производительность микропроцессора. Также важен состав этих команд. Удачно подобранные команды дают в итоге большую производительность. По данным производителей, число команд варьируется между 46 и 158. В таблице на рис. 15.15 представлены некоторые часто используемые микропроцессоры и их свойства.

15.6.2. Микропроцессор SAB 8080A

Микропроцессор 8080A выпускается в настоящее время в больших количествах и превратился в своего рода стандарт для микропроцессоров. Он производится Intel, Siemens и другими. Siemens-обозначение — SAB 8080A.

Микропроцессор SAB 8080A является 8-битовым микропроцессором в *N*-МОП-технике с длительностью обработки цикла 2 мкс. Система команд содержит 78 команд (см. табл. на рис. 15.15). Микропроцессор ТТЛ-совместим и выпускается в TRI-STATE-технологии, то есть информационные входы и выходы, а также адресные выходы могут принимать, кроме уровней *L* и *H*, еще и высокоомное (выключенное) состояние.

ОЗУ микропроцессора SAB 8080A являются памятью динамического типа, требующей обновления. Обновление происходит с сигналом синхронизации. Микропроцессор SAB 8080A выпускается в 40-полюсном DIP-корпусе. Цоколевка показана на рис. 15.16.

Вводы с 3 до 10 являются информационными. Они являются выходами и входами 8-битовых шин данных (см. гл. 11, разд. 11.4). Эти шины являются двунаправленными. Они могут как вводить, так и выводить данные.

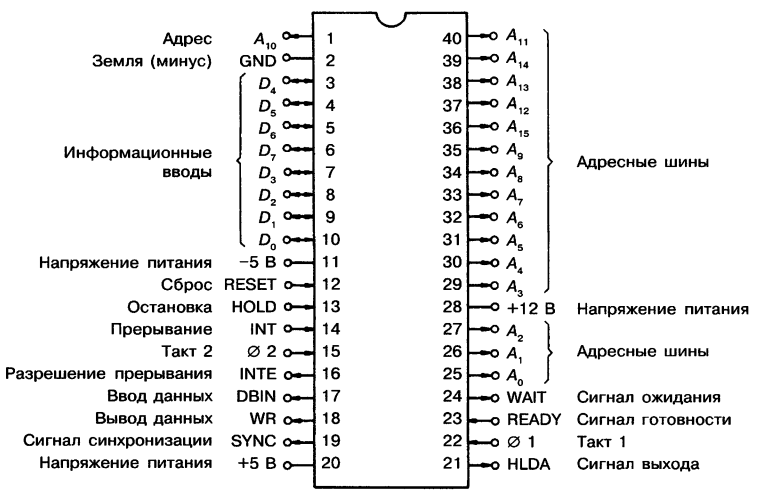


Рис. 15.16. Цоколевка микропроцессора SAB 8080A (Siemens).

Микропроцессор имеет 16 адресных выходов. Могут быть выбраны $2^{16} = 65\,536$ различных адресов, т. е. микропроцессор может адресовать внешнюю память объемом до 64 килобайт. По каждому адресу может находиться 1 байт = 8 бит. Дополнительно еще есть адреса для управления модулями ввода и вывода. Вместо устройства памяти на 64 килобайта могут быть подключены несколько малых блоков памяти. При этом безразлично, идет ли речь об ОЗУ, ПЗУ или ППЗУ. Адресными выходами служат выводы корпуса 1, 25, 26, 27, 29 до 40 (рис. 15.16). Они связаны с адресной шиной.

Другие выводы нужны для питания и управления микропроцессором. Напряжения питания: +12 В (порт 28), +5 В (порт 20), −5 В (порт 11). Землю нужно подключить к порту 2. Микропроцессор нуждается во внешнем тактовом генераторе (генераторе синхронизирующих импульсов), который производит два различных такта — такт 1 и 2 (такт 1 в порт 22, такт 2 в порт 15). Через вход RESET (порт 12) счетчик команд ставится на 0. С 0-сигналом в порту HOLD (порт 13) микропроцессор делает паузу. В это время могут вводиться или выводиться данные. Выводы INT (порт 14) и INTE (порт 16) служат для прерывания работы программы и продолжения работы программы.

Сигнал на входе DBIN (Data BUS In) показывает, что на шине данных находится информация для ввода. Данные могут быть введены в микропроцессор. Порт $\overline{WR}$ выводит 0-сигнал до тех пор, пока данные не будут переписаны из внешней памяти. SYNC-порт выдает синхронный сигнал, когда начинается новый операционный цикл.

Особенно важными являются порты WAIT (ожидание, порт 24) и READY (готовность, порт 23). На WAIT действует 1-сигнал, если микропроцессор находится в режиме ожидания. 1-сигнал в порте READY показывает, что на шине есть данные, готовые к передаче. Микропроцессор кратковременно останавливается для передачи данных.

Через порт HLDA (порт 21) микропроцессор выдает так называемое подтверждение останова как ответ на HOLD-сигнал. Оно показывает, что шины данных и адреса переведены в высокоомное состояние.

Внутренняя структура SAB 8080A достаточно сложная. Поэтому разобьем объяснение на две части. На упрощенном изображении внутреннего строения (рис. 15.17) четко выделено АЛУ. Обращение данных A-входов происходит по внутренней шине. По внутренней шине также циркулируют данные при записи в память и при считывании из памяти.

Команды поступают через информационные входы. Они могут промежуточно сохраняться в памяти. Специальной памяти программ не существует, и команды могут сохраняться в специально подключаемой внешней памяти. Через регистр команд команды поступают в кодировщик и в виде управляющих сигналов подаются на схему управления. Схема управления имеет вводы—выводы, о которых уже было сказано ранее.

С помощью счетчика команд и адресной памяти выводятся адреса для управления внешними модулями, например адресации ПЗУ и ОЗУ. Адреса состоят из 16-битовых слов. Они подаются на адресную шину.

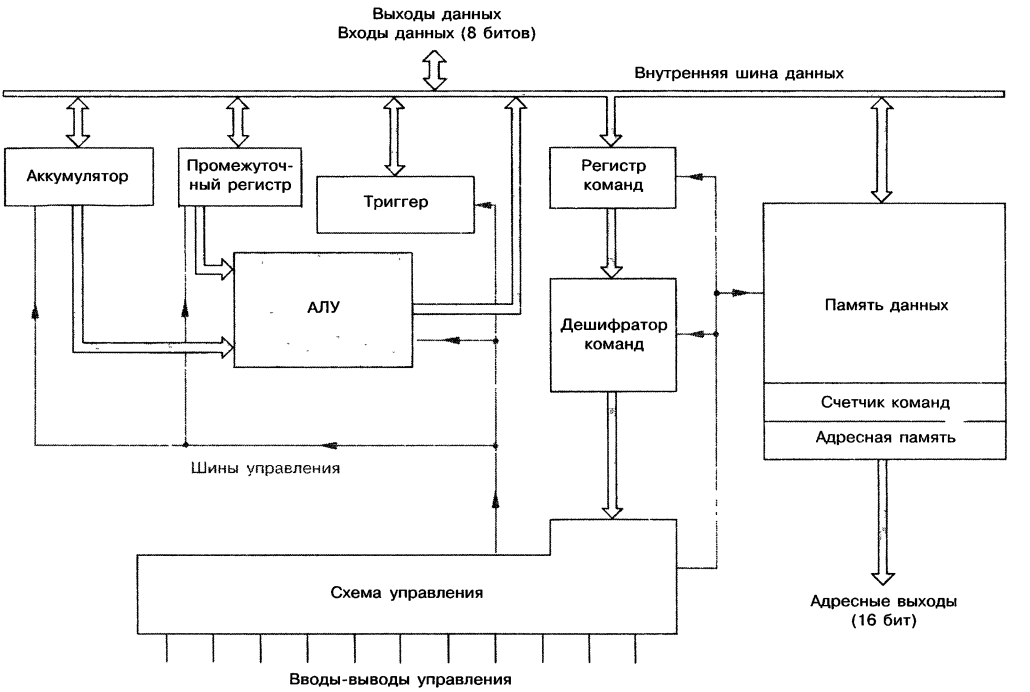


Рис. 15.17. Упрощенная структура микропроцессора SAB 8080A.

Производители приводят в документации блок-схему процессора SAB 8080A (рис. 15 18).

На схеме изображен ряд дополнительных блоков, например буфер для данных и адресов. Буфера — это промежуточная память. Память данных разделена на множество регистров, которые выбираются мультиплексором.

Вся эта схема собрана в одном кремниевом чипе размером 5 × 5 мм. Сегодня эти процессоры стоят сравнительно дешево.

Как сделать систему управления на базе микропроцессора SAB 8080A? С одним микропроцессором систему управления сделать нельзя. Дополнительно требуются: тактовый генератор, модули ввода—вывода, модули памяти ОЗУ, ПЗУ, ППЗУ для программ и данных, блок питания.

15.6.3. Дополнительные модули для микропроцессоров

Микропроцессоры (МП) управляются тактирующими синхроимпульсами. Требуется минимум один тактирующий сигнал, но часто используются и два разных такта. Некоторые МП имеют встроенный генератор, но чаще к ним подключают дополнительные модули, которые производят необходимые тактовые импульсы.

Для МП SAB 8080A в качестве внешнего генератора импульса рекомендован SAB 8224. Он поставляется в 16-пиновом DIP-корпусе и содержит кварцевый осциллятор, управляемый внешним кварцевым генератором. Частота кварца определяет скорость работы МП. Типовой является частота

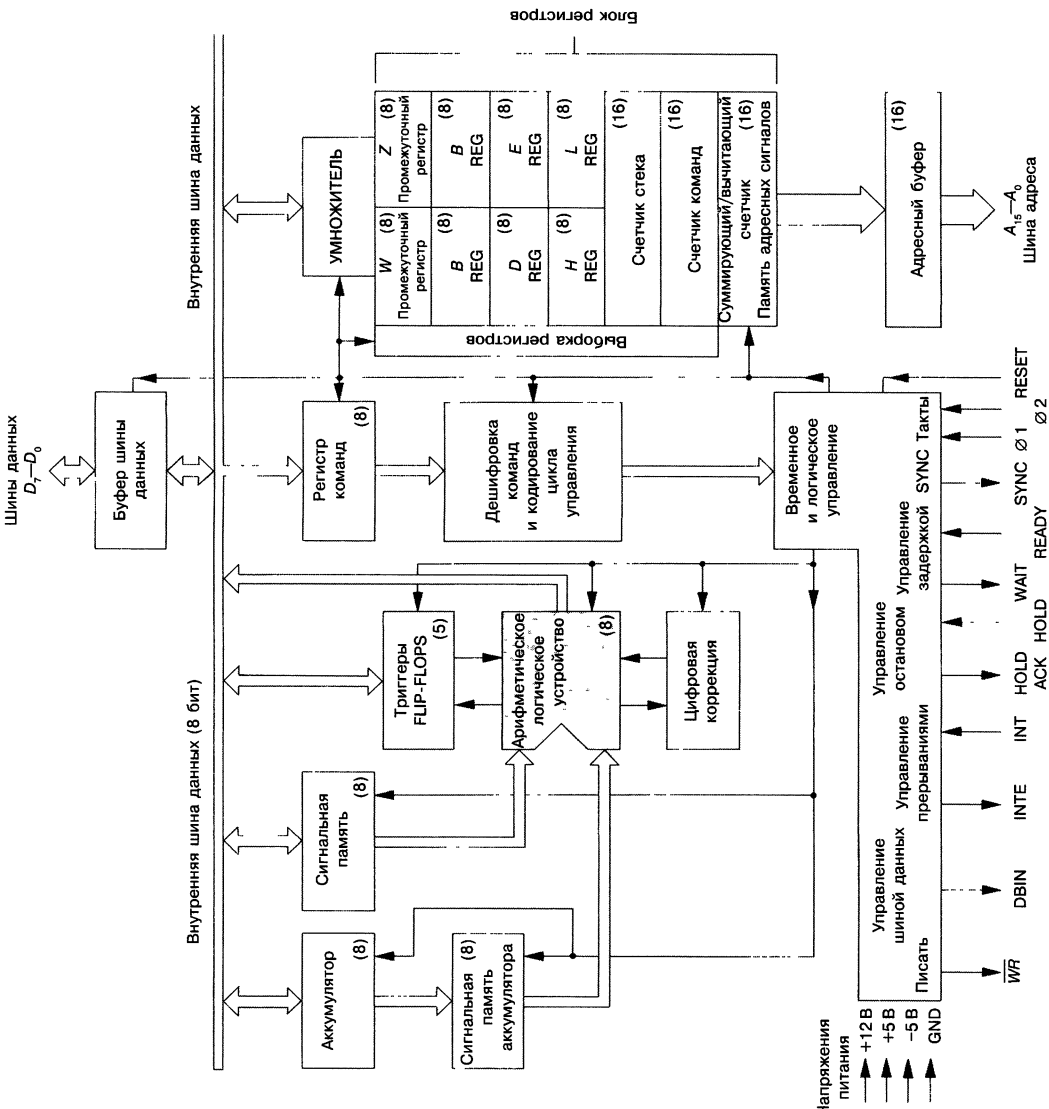


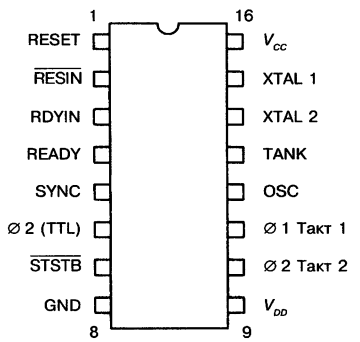
Рис. 15.18. Полная структура микропроцессора SAB 8080A (Siemens).

18 МГц. Колебания осциллятора преобразуются в меандр, и частота меандра делится на 9. Получается рабочая частота 2 МГц.

Цоколевка модуля SAB 8224 и форма тактового сигнала приведена на рис. 15.19. Дополнительно к тактам модуль производит управляющие сигналы, например сигнал о включении напряжения питания.

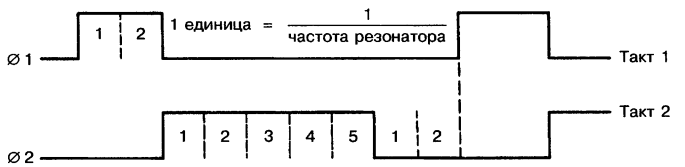
Для ввода—вывода данных также необходимы дополнительные модули. Они называются блоки ввода—вывода (англ. *I/O*-модули, Input/Output-модули). Они принимают данные в промежуточную память (буфер). При вводе они производят сигналы управления, с помощью которых микропроцессор работает до тех пор, пока не закончится ввод данных. Блоки ввода—вывода часто содержат также модуль выборки. Если данные считываются с микропроцессора, блок ввода—вывода выбирает нужные данные. Для микропроцессора SAB 8080A имеются в наличии различные блоки ввода—вывода. Часто используется модуль SAB 8212.

Цоколевка



Обозначение выводов

$\overline{\text{RESIN}}$	Ввод сброса
RESET	Вывод сброса
RDYIN	Ввод готовности
READY	Вывод готовности
SYNC	Ввод сигнала синхронизации
$\overline{\text{STSTB}}$	Передача управления (Low активен)
Ø 1	Генератор импульсов для SAB 8080
Ø 2	
XTAL 1	Подключение внешнего кварцевого генератора
XTAL 2	
TANK	Ввод кварцевого генератора высших гармоник
OSC	Вывод резонатора
Ø 2 (TTL)	Генератор синхроимпульсов (ТТЛ-уровень)
V_{CC}	Напряжение питания +5 В
V_{DD}	Напряжение питания +12 В
GND	Земля (0 В)



Пример: SAB 8080 t_{CY} (период синхроимпульса) = 50 нс
OSC = мГц/55 нс
Ø 1 = 110 нс (2 × 55 нс)
Ø 2 = 275 нс (5 × 55 нс)
Ø 1—Ø 2 = 110 нс (2 × 55 нс)

Рис. 15.19. Цоколевка тактового генератора SAB 8224 и временные диаграммы (Siemens).

Очень важными дополнительными модулями являются блоки памяти. Можно поставить любой тип памяти допустимой емкости: ОЗУ, ПЗУ, ППЗУ. Максимально допустимая емкость памяти составляет для микропроцессора SAB 8080A 64 килобайт, так как микропроцессор имеет всего 65 636 различных адресов.

Внешние блоки памяти так важны, потому что внутренний объем памяти микропроцессора относительно невелик, так что программы и данные обработки должны сохраняться во внешних модулях.

В качестве внешней памяти можно использовать также лентопротяжные устройства (цифровые кассеты) и магнитные диски. Для подключения подобных устройств необходимы модули интерфейсов, которые преобразовывают данные определенным образом и производят управляющие сигналы. Они должны преобразовать, например, параллельно выданные 8-битовые данные микропроцессора в последовательную форму, которую воспринимает накопитель на магнитных дисках.

Применение блоков ввода—вывода, блоков памяти и интерфейсов требует сигналов управления, которые микропроцессор не может поставлять полностью. По этой причине в качестве дальнейшего дополнительного модуля во многих случаях требуется система управления. Такой модуль производит все сигналы, которые нужны для подключения дополнительных модулей. Он часто также содержит так называемый **драйвер шины**. Драйвер шины является усилителем для сигналов, которые подаются в шину.

Система управления SAB 8228 разрабатывалась для микропроцессора SAB 8080A. Она содержит 8-битовый двунаправленный драйвер шины. Управляющая система выдает все необходимые сигналы управления и еще дополнительные сигналы управления для прерываний программы и для применения многобайтовых команд. Многобайтовые команды — это команды, которые имеют длину слова два или больше байт.

15.7. Микрокомпьютер

Если соединить микропроцессор с необходимыми дополнительными модулями, получится **микрокомпьютер**. Некоторые дополнительные модули всегда необходимы — такие как тактовый генератор и память. Другие дополнительные модули выбираются в зависимости от выполняемой задачи.

> Микрокомпьютер является системой, состоящей из микропроцессора и дополнительных модулей.

Микрокомпьютеры чаще всего строятся на одной плате. Такая плата может содержать, например, следующие модули:

Микропроцессор: SAB 8080A
Тактовый генератор: SAB 8224
ППЗУ: SAB 8708
ОЗУ: SAB 8111-2
I/O модули: SAB 8212
Система управления: SAB 8228
Кварцевый генератор для тактового генератора.

Схема такого микрокомпьютера приведена на рис. 15.20. Шина данных от SAB 8080A идет к системе управления SAB 8228. Здесь входящие и исходящие данные усиливаются. Шины данных являются 8-битовыми двунаправленными.

Шиной адреса является 16-битовая однонаправленная шина. Адреса поступают всегда от микропроцессора. К шине данных и к шине адреса подключены устройства ввода—вывода и блоки памяти. Управление происходит от модуля SAB 8228. В REPROM должны быть сохранены отдельные команды, которые нужно запускать по очереди, т. е. программа работы. Необходимые данные вводятся извне в ОЗУ. Теперь МП готов решать задачи управления или счета. Результаты будут сохранены в оперативной памяти и по желанию выведены на внешние устройства.

Программирование микрокомпьютера требует длительной практики. К сожалению, каждый тип микропроцессора имеет свои команды. Поэтому рекомендуется выбрать для начала один МП и сначала работать только с этим типом микропроцессоров. После овладения системой команд одного типа МП переход на другой тип относительно прост.

Производители МП предлагают помощь при изучении программирования. Имеется большое число учебных микрокомпьютеров, с которыми можно вырабатывать навык программирования. Рекомендуется посещение учебного курса.

Микрокомпьютеры могут собираться из микропроцессоров и разнообразных дополнительных модулей.

Можно выбирать разнообразные дополнительные модули и варьировать типы и объемы памяти, чтобы достигнуть оптимальной конфигурации для решения поставленной проблемы. Для этого производители предлагают оценочные модули, которые значительно упрощают проектно-конструкторские работы.

Интересное решение проблемы представляют 1-чиповые микрокомпьютеры. В единственном чипе интегрирован полный микрокомпьютер. Не надо

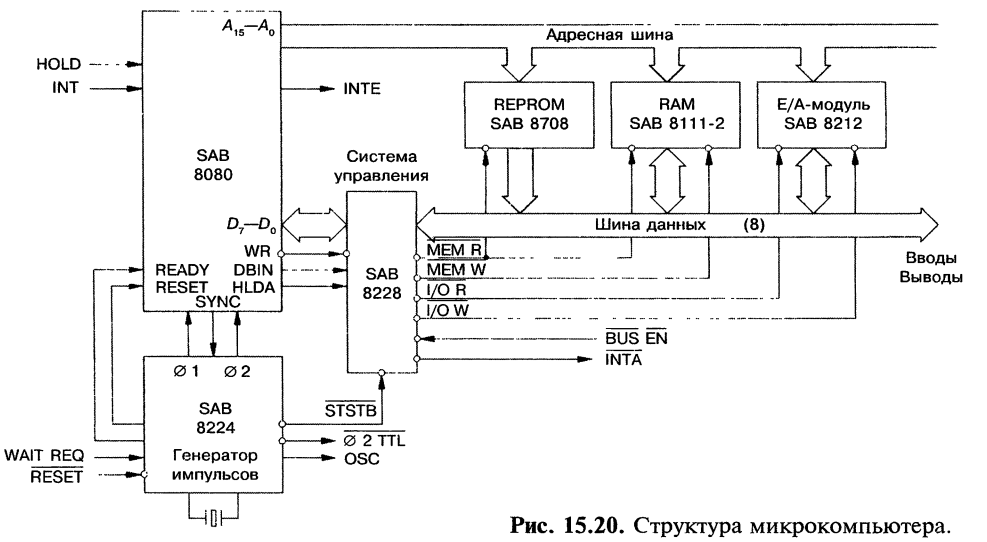


Рис. 15.20. Структура микрокомпьютера.

думать о дополнительных модулях, как на рис. 15.20. Система собрана полностью. 1-чиповый микрокомпьютер должен содержать все необходимое.

Что фактически содержит такой микрокомпьютер? Прежде всего интересно, какую память он имеет и каков ее объем? Если рассмотреть находящийся на рынке 1-чиповый микрокомпьютер, например, TMS 1000 производства Texas Instruments, то выяснится, что объем памяти мал. Применяемая в настоящее время плотность интеграции не допускает большие объемы памяти. Также микропроцессор является всего лишь 4-битовым микропроцессором. Из этого следует:

> 1-чиповые микрокомпьютеры предназначены в настоящее время только для простых задач управления.

Они недостаточно флексибельны, т. е. не допускают расширения для решения сложных задач.

Плотность интеграции микросхем растет с каждым годом. Значит в будущем 1-чиповые микрокомпьютеры будут иметь большие объемы памяти и приобретут большее значение.

Контрольный тест

1. Объясните структуру АЛУ.
2. Какое преимущество имеет перекодирование 6 входов управления АЛУ на 4 входа управления?
3. Что такое аккумулятор? Опишите принцип действия.
4. Изобразите схему аккумулятора с АЛУ, регистрами и памятью переноса.
5. Сколько различных команд можно передать 4-битовым модулем?
6. Объясните принцип действия аккумулятора с памятью с помощью схемы на рис. 15.11.
7. Чем отличается микропроцессор от микрокомпьютера?
8. Что означают обозначения: 4-битовый микропроцессор, 8-битовый микропроцессор, 16-битовый микропроцессор?
9. По каким технологиям производятся микропроцессоры?
10. Что понимают под шиной данных, шиной адреса?
11. Микропроцессор нуждается в дополнительных модулях. Назовите четыре возможных дополнительных модуля.
12. Что такое 1-чиповый микрокомпьютер, что такое 1-платный микрокомпьютер?

ГЛАВА 16

ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ СХЕМЫ

С развитием интегральных микросхем возникло желание программировать их для выполнения специальных задач. Проектирование микросхемы относительно дорого и окупает себя только при производстве большого количества микросхем. В настоящее время разработаны микросхемы, которые программируются производителем или самими пользователями.

16.1. Логические схемы, программируемые изготовителем

Изготовление микросхем происходит посредством ряда технологических операций. После подготовительной стадии разрабатываются транзисторные каскады. Затем они собираются в логические структуры. Затем производятся резисторы и связи между логическими структурами. Изготовление напоминает проявление фотографии с помощью так называемых масок. Маски — это стеклянные пластины, немного больше, чем кремниевые пластины, на которые нанесены фотографически уменьшенные логические структуры. Они переносятся на полупроводниковые пластины с помощью светочувствительного лака, освещения и напыления.

Схемы с большими логическими структурами называются **программируемыми вентильными матрицами**. Вентильные матрицы — это схемы-полуфабрикаты. Имеющиеся на них вентили (логические элементы) могут быть соединены друг с другом по желанию пользователя. Линии связи производятся масками. Это называют масочным программированием.

Вентили могут быть собраны на полупроводниковой пластине в некоторые наборы схемных элементов, называемых **стандартными ячейками**. Эти ячейки затем собираются в более сложные схемы. Схема программируется производителем и называется схемой, ориентированной на конечного заказчика.

Если потребителю нужна микросхема с определенными свойствами, то производитель выбирает полупроводниковую пластину с необходимым количеством транзисторов, собранных в стандартные ячейки. Разработчик проектирует маску с необходимыми логическими связями. С помощью этой маски изготавливается необходимая интегральная микросхема. В зависимости от функций микросхема встраивается в корпус, который имеет от 16 до 144 выводов (рис. 16.1).

> Матричные кристаллы, базовые ячейки и схемы, ориентированные на конечного заказчика — это разновидности ASIC микросхем (Application Specific Integrated Circuit).

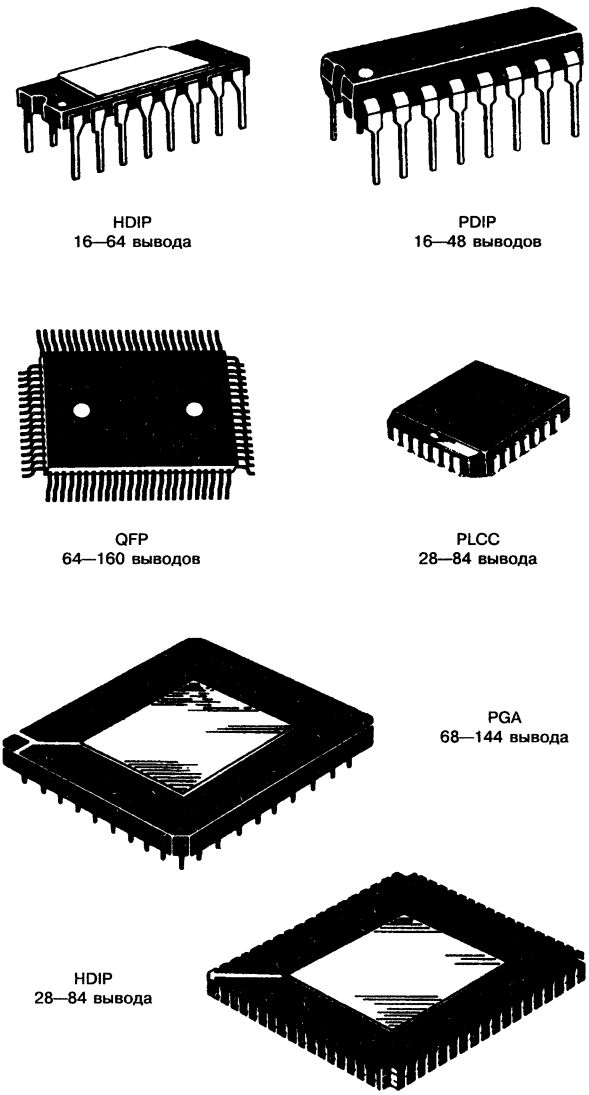


Рис. 16.1. Типичные ASIC-корпуса, от 16 до 144 выводов (pin).

ASIC имеют большие преимущества. Большая логическая схема может быть размещена на маленьком пространстве. Особенно высокую плотность размещения можно достигнуть с *N*-МОП-технологией. Крохотные габариты транзистора и короткие соединения приводят к высокому быстродействию. Энергопотребление относительно мало.

Применение ASIC экономически выгодно при условии изготовления партии более 5000 штук. При меньшем количестве единичная схема получается очень дорогой.

Обычно разработанная для клиента ASIC принадлежит именно этому клиенту. Только он может ее использовать и имеет, таким образом, монополию. Однако часто готовые ASIC по договоренности с заказчиком используются другими фирмами. Это выгодно для всех участников сделки.

16.2. Логические схемы, программируемые потребителем

Применяя логические схемы, программируемые пользователем, можно сильно сократить время на разработку. Выпуск малых партий становится экономически целесообразным.

> Логические схемы, программируемые потребителем, называются PLD (Programmable Logic Devices) — программируемые логические устройства.

Наряду с этим существует множество фирменных обозначений, которые будут рассмотрены дальше. Некоторые производителей относят PLD к ASIC. Однако в основном под понятием ASIC понимают схемы, программируемые пользователем.

16.2.1. Основы

Как уже известно, логические связи любой схемы могут быть выражены в нормальной форме ИЛИ, т. е. логической суммы ИЛИ полных конъюнкций (см. гл. 5). Для схемы с двумя входными переменными получаются четыре полных конъюнкции (рис. 16.2).

Схема на рис. 16.2 содержит программируемые связи, обозначенные косой чертой. Эти связи являются непроводящими. Проводящие связи обозначены крестом. В схемотехнике в общем используются условные обозначения как на рис. 16.3.

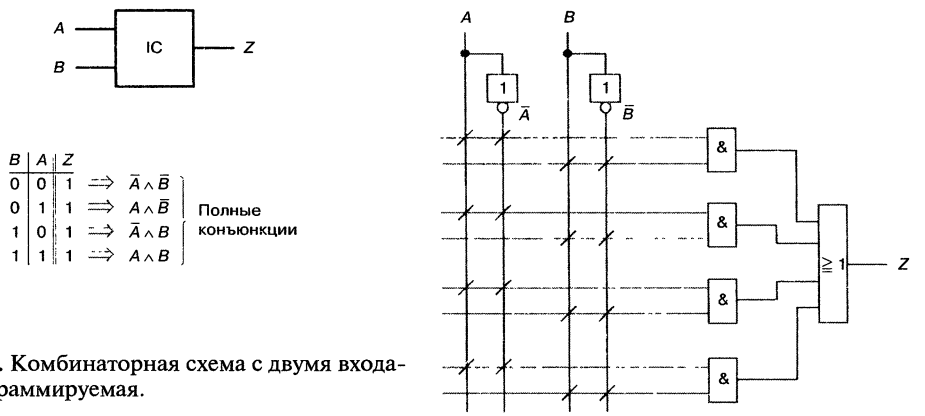


Рис. 16.2. Комбинаторная схема с двумя входами, программируемая.

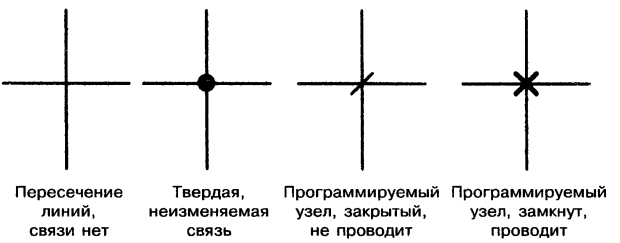


Рис. 16.3. Графические условные обозначения состояний узлов.

На схеме (рис. 16.2) каждый элемент И может производить полную конъюнкцию. Четыре полные конъюнкции возможны при двух входных переменных. Таким образом, существуют все возможные нормальные формы ИЛИ. В общем, из схемы на рис. 16.2 нужны только три И-элемента, так как нормальная форма ИЛИ, образованная из четырех полных конъюнкций, дает в итоге, как известно, 1.

> Если по схеме можно составить все возможные нормальные формы ИЛИ, то с этой схемой могут производиться все возможные логические операции.

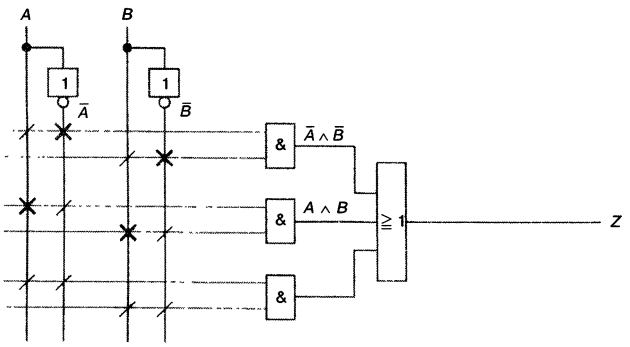
Составим по рис. 16.4 схему эквивалентности. Как программировать? Верхний И-элемент должен производить $\bar{A} \wedge \bar{B}$, средний — $A \wedge B$. Программирование состоит в том, чтобы правильно перемкнуть линии управления.

Там, где должна быть проводящая связь, ставится крест. Третий И-элемент не нужен. Его можно отключить от питания. Существуют PLD, в котором после изготовления все пересечения проводящие. В таких PLD при программировании электрически пережигаются перемычки. После прожигания программирование считается завершенным. Перепрограммирование, естественно, становится невозможным. При крупной ошибке программиста модуль можно выбрасывать.

В местах пересечения могут также находиться полевые транзисторы FAMOS-FET (см. разд. 12.6). Если затвор такого транзистора заряжен, то узел проводящий. Если разряжен, то узел разорван. Разрядка затвора может быть произведена ультрафиолетовым светом. Через окошко в корпусе облучается вся микросхема. Все затворы разряжаются. Этот способ программирования аналогичен СППЗУ (стираемое программируемое постоянное запоминающее устройство).

> Стираемые ультрафиолетом PLD называются EPLD (Erasable PLD; англ. стираемые PLD).

EPLD можно перепрограммировать сколь угодно часто. Программирование производится посредством заряда определенных затворов. Недавно появились PLD, которые можно программировать электрическими импульсами. Эти PLD называются EEPLD (Electrical Erasable PLD, электрически стираемые PLD). Часто используется также обозначение E^2PLD. EEPLD устроены аналогично EEPROM (разд. 12.6.2).



B	A	Z	
0	0	1	$\Rightarrow \bar{A} \wedge \bar{B}$
0	1	0	
1	0	0	
1	1	1	$\Rightarrow A \wedge B$

Рис. 16.4. PLD-схема ИСКЛЮЧАЮЩЕГО ИЛИ.

Программирование EEPLD производится электрическими импульсами. Возможно частичное стирание ячеек. Необязательно стирать весь модуль.

Производимые в настоящее EEPLD могут перепрограммироваться, по данным производителя, минимум сто раз. Заряд затвора FAMOS-FET очень стабилен, т. е. затвор очень хорошо изолирован. Запрограммированный модуль должен хранить информацию, по данным производителя, десять лет и более.

При графическом изображении PLD-схем приняты условные обозначения согласно рис. 16.3.

Типичная PLD-схема представлена на рис. 16.5, *а*. Схема состоит из И-связи, также называемой И-матрицей (AND Array) и ИЛИ-связи (ИЛИ-матрица; OR Array), обе программируемы. Все программируемые связи выполнены проводящими.

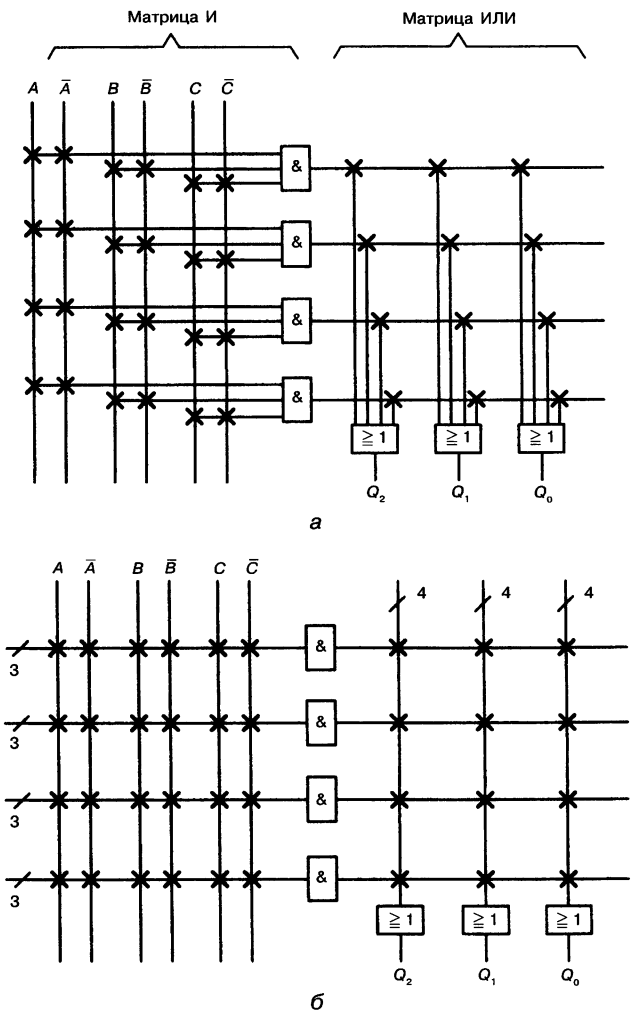


Рис. 16.5. PLD-схема:

а — одножильные провода; *б* — многожильные провода.

Программированием связей могут производиться логические операции. Связи разделяют в соответствии с заявленными нормальными формами ИЛИ или другими уравнениями алгебры логики и получают нужную логическую схему. Ненужные элементы выключаются.

Для PLD с большим количеством элементов получается много проводников, которые усложняют чтение схем. Чтобы улучшить наглядность, проводники изображаются как шины, как представлено на рис. 16.5, *б*.

> В PLD с большим количеством проводников они изображаются как шины.

Ясно, например, что И-элемент с тремя входами имеет также три входных провода. Их можно обозначить шиной из трех проводников, с косой чертой 3, как изображено на рис. 16.5, *б*.

16.2.2. PAL-схемы

Сокращение PAL обозначает Programmable Array Logic (программируемая матричная логика). Обозначение PAL используется повсеместно, однако является товарным знаком фирмы Monolitic Memories.

> PAL-схемы имеют программируемую пользователем И-матрицу и фиксированную ИЛИ-матрицу.

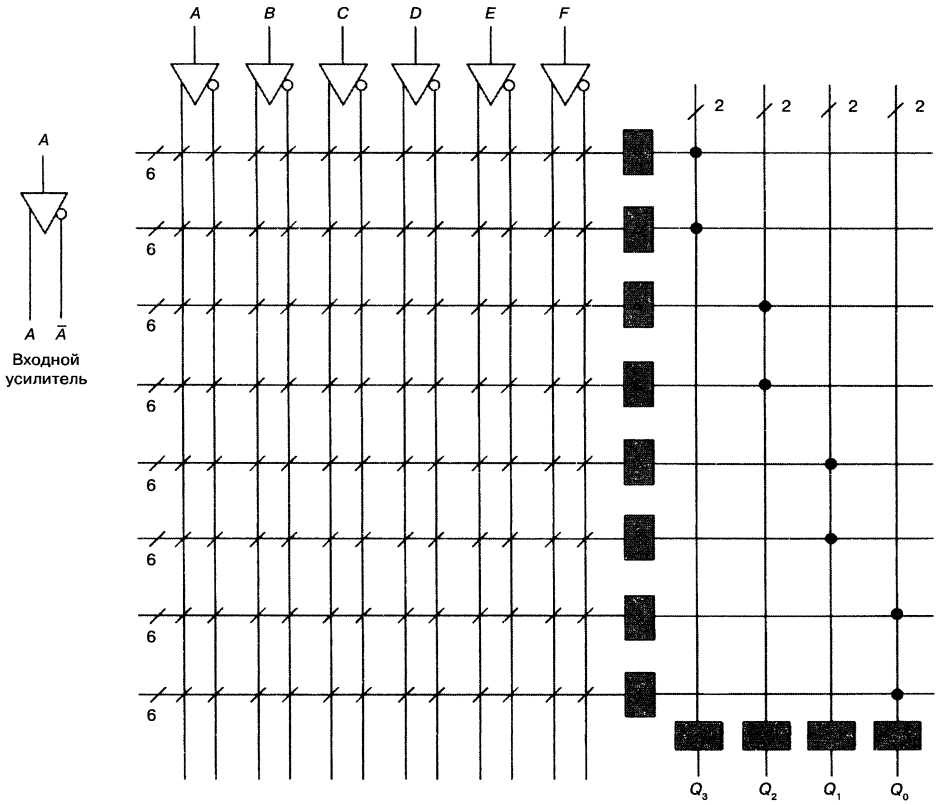


Рис. 16.6. PLD с программируемой И-матрицей (PAL-схема).

Принципиальная PAL-схема представлена на рис. 16.6. Программированием обозначенных косой чертой узлов определенные входные сигналы подключаются — в этой схеме их максимально шесть — к И-элементам. Каждый И-элемент имеет шесть входов (шесть горизонтальных проводников). Выходы двух И-элементов подаются на ИЛИ-элемент. С этой схемой можно производить четыре различных связи. Они действуют на выходах от Q_0 до Q_3. Ненужные связи остаются неиспользованными.

PAL-схема на рис. 16.6 должна программироваться так, чтобы возникла связь согласно следующему уравнению:

$$Z = \left(\bar{A} \wedge B \wedge \bar{C} \wedge D \wedge \bar{E} \wedge F\right) \vee \left(A \wedge B \wedge C \wedge \bar{D} \wedge E \wedge \bar{F}\right).$$

Решение показано на рис. 16.7. Величина Z поступает на выход Q_3. Обычно PAL-схемы имеют множество входов и выходов, так что можно организовать очень сложные логические связи. Чаще всего они работают при положительной логике (H-активно). Часто на выходах находятся триггеры, в которых сохраняется результат. Такие схемы имеют так называемый выходной регистр. Также выходы могут переключаться нужным образом (комбинаторный выход, TRI-STATE-выход, выходной регистр, выход с инверсией сигналов). Выходы такой разновидности называют программируемыми выходами.

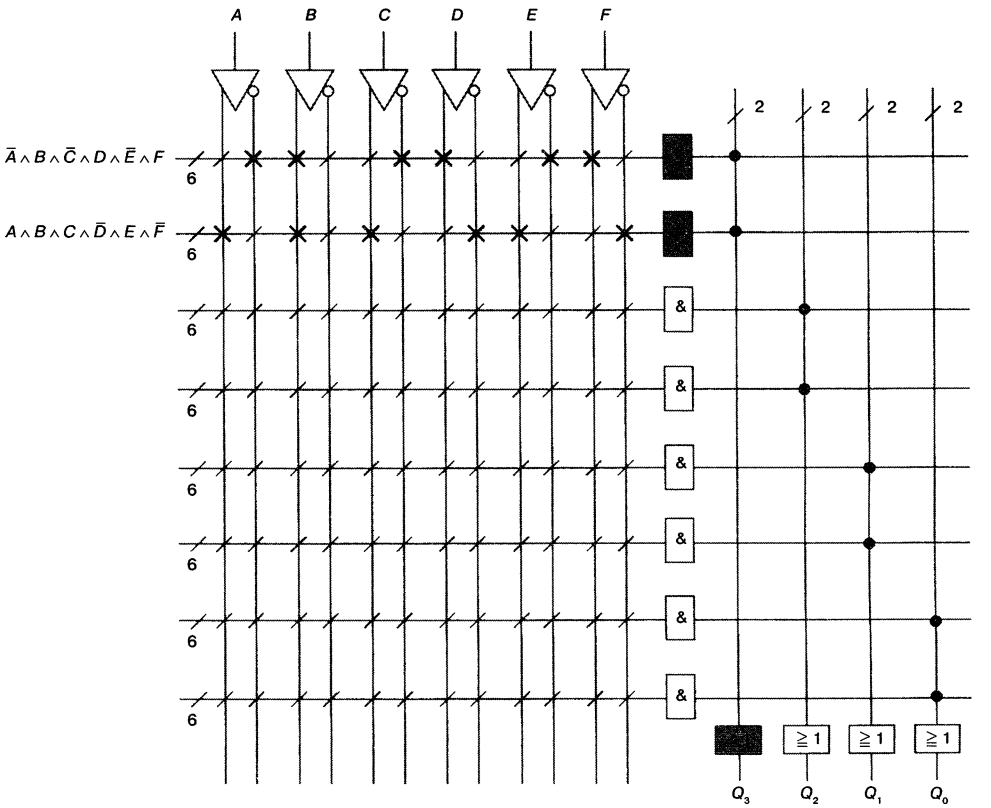
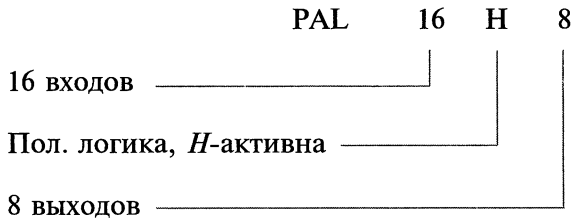


Рис. 16.7. Программируемая PAL-схема.

Обозначение PAL-модуля рассмотрим на примере:



Вместо буквы *H* могут использоваться другие буквы, которые имеют следующие значения:

L — *L*-актив, отрицательная логика

R — выходной регистр

C — дополнительный выход (переключается *L*-актив, *H*-актив)

P — программируемый выход

X — выход ИСКЛЮЧАЮЩЕГО ИЛИ (через регистр, сигналы могут инвертироваться или нет).

На рис. 16.8 показана схема модуля PAL 12 H 6. Модуль имеет 12 входов. Входные сигналы усиливаются входными усилителями. Наряду с не инвертированным выходом усилителя имеется инвертируемый выход. Входные сигналы идут на 24 вертикальных проводника и оттуда распределяются на И-элементы. Каждый И-элемент имеет 24 входа. Следовательно, к И-элементам подходят 24 жильные шины. Эта PAL-схема относится к малым схемам. Она производится в 20-выводном DIP-корпусе. Программируемые связи на рис. 16.8 не показаны. При 16 И-элементах с 24 проводниками на каждый получается 384 программируемых узла. Схема была бы очень большой и запутанной.

PAL-схемы потребляют относительно большой ток (от 95 до 200 мА на модуль) и работают относительно медленно. Время задержки составляет в среднем 50 нс. Первые разработанные PAL-схемы были построены в биполярной технологии, т. е. на базе NPN и PNP транзисторов. В новых PAL-схемах применяются исключительно *N*-МОП- и *K*-МОП-технологии. Благодаря этому снижается потребление тока и уменьшается время задержки. Напряжение питания 5 В.

16.2.3. GAL-схемы

GAL — это сокращение от Generic Array Logic (логика на базовых матричных кристаллах). Это обозначение является товарным знаком фирмы Lattice Semiconductor. GAL-схемы были спроектированы с целью улучшить некоторые известные недостатки PAL-схем. Программирование прожигом очень тонких перемычек больше не применяется. GAL-схемы являются полностью EPLD (стираемые ультрафиолетом) или EEPLD (стираемые электрически). Они могут перепрограммироваться более ста раз.

GAL-схемы, как и PAL-схемы, используют программируемую И-матрицу. Они имеют дополнительные модули ввода—вывода. Матрица ИЛИ фиксирована.

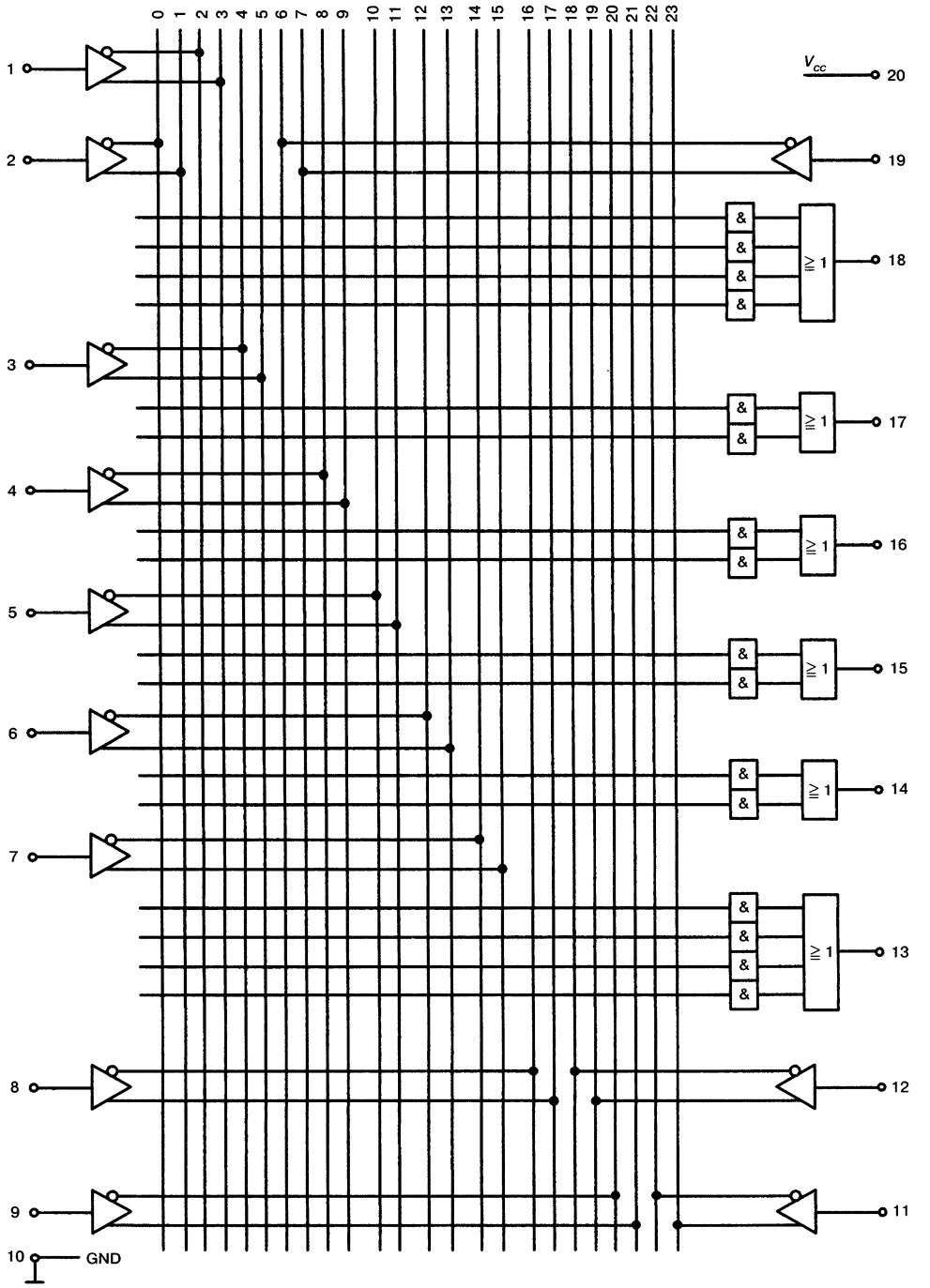


Рис. 16.8. PAL-схема 12H с 12 входами и 6 выходами.

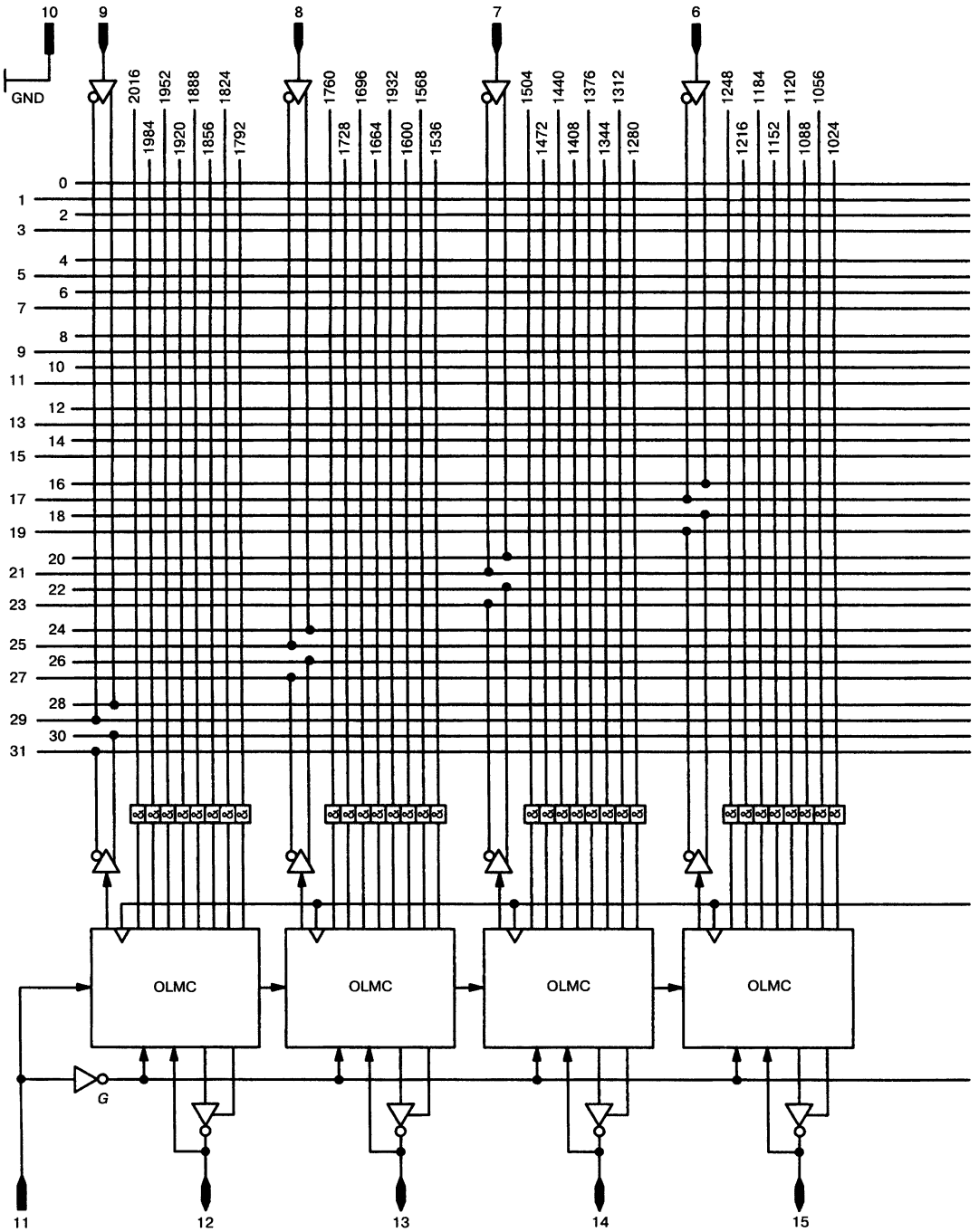
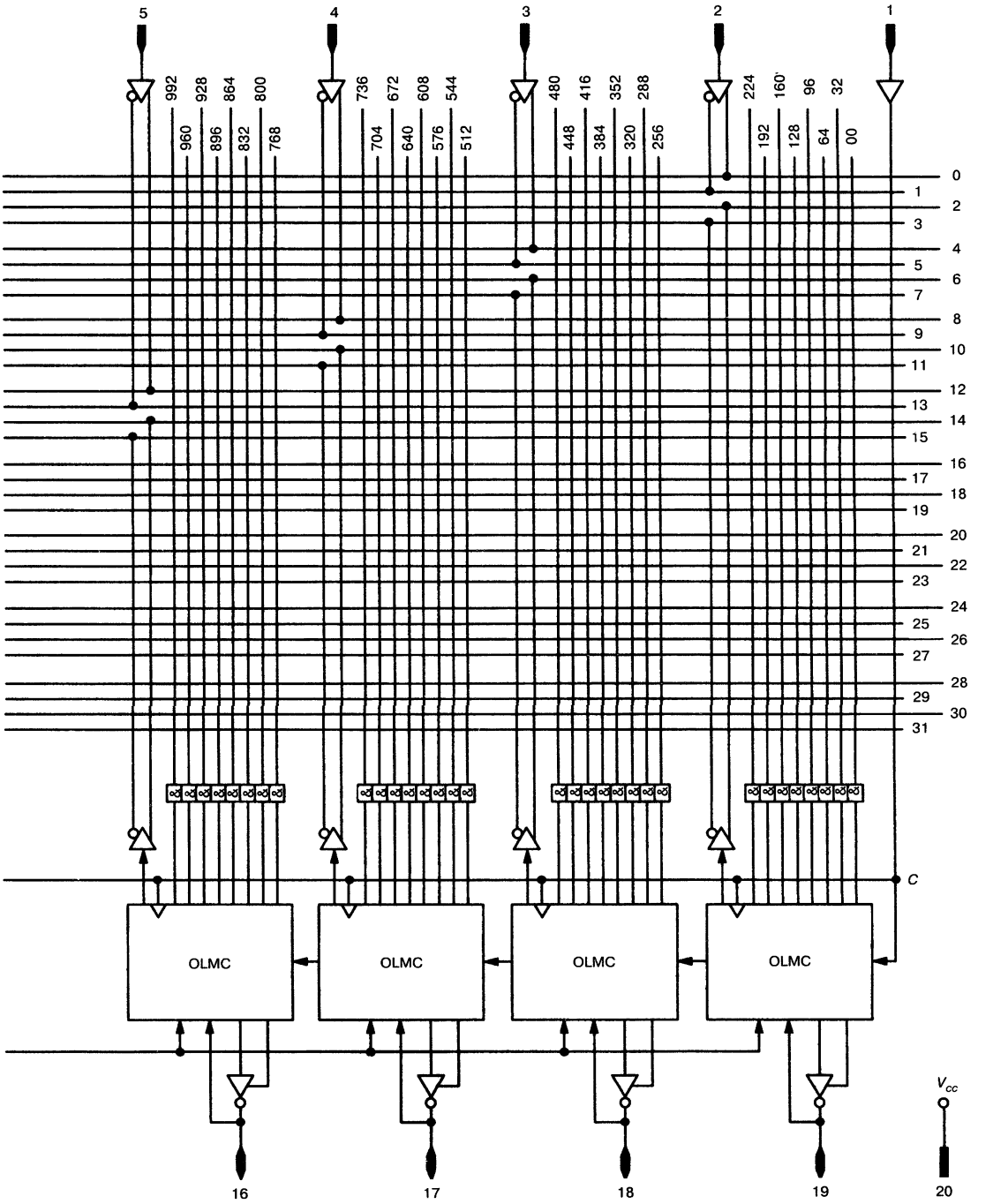
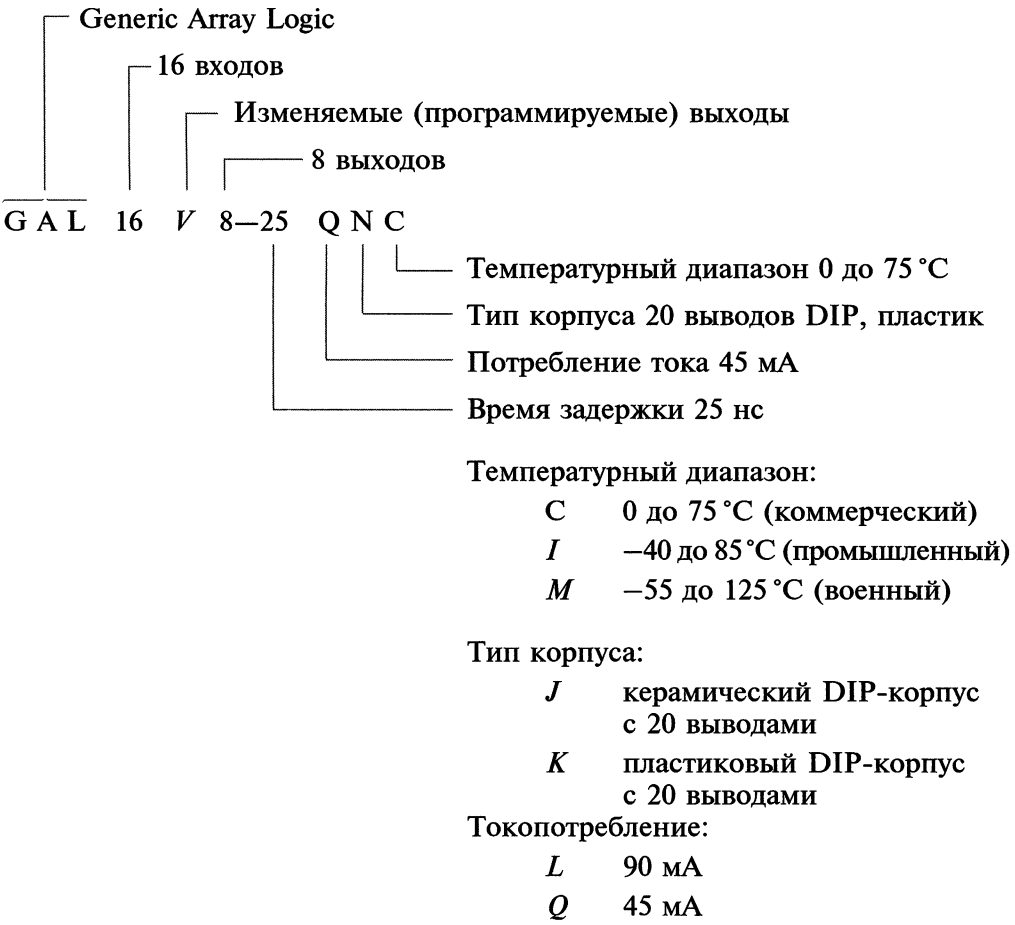


Рис. 16.9. GAL-схема GAL 16 V 8.


5
4
3
2
1
992
928
864
800
960
896
832
768
736
672
608
544
704
640
576
512
480
416
352
288
448
384
320
256
224
160
96
32
192
128
64
00
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
C
OLMC
OLMC
OLMC
OLMC
V_{CC}
16
17
18
19
20

На рис. 16.9 показана структура типичной GAL-схемы. Она имеет обозначение 16 *V*8, следовательно имеет 16 входов и 8 выходов. Буква *V* значит программируемые выходные блоки. Восемь входов доступны через порты от 2 до 9. Другие восемь входов подключаются через выходы. Обозначение модуля аналогично PAL-схемам. Дополнительно указываются время задержки, токопотребление, тип корпуса и диапазон температуры.

Пример



Температурный диапазон:

C	0 до 75 °C (коммерческий)
I	−40 до 85 °C (промышленный)
M	−55 до 125 °C (военный)

Тип корпуса:

J	керамический DIP-корпус с 20 выводами
K	пластиковый DIP-корпус с 20 выводами

Токопотребление:

L	90 мА
Q	45 мА

Схема на рис. 16.9 имеет входной усилитель, который имеет на выходе инвертированный и неинвертированный сигналы. Эти сигналы выводятся на 32 вертикальных проводника и распределяются оттуда на горизонтальные проводники.

> Горизонтальные проводники, которые ведут к И-элементам, состоят из 32 однопроводных линий. Каждый И-элемент имеет 32 входа.

Номера горизонтальных проводников указаны слева от входных усилителей. Схема содержит всего 2048 горизонтальных проводников. Каждый И-элемент может создавать одну из возможных полных конъюнкций.

На восьми выходах находятся так называемые OLMC-схемы (OLMC — Output Logic Macro Cell). Каждая OLMC-схема содержит 8-кратный элемент ИЛИ, который производит нормальную форму ИЛИ, *D*-триггер для сохранения результата, управление возвратом сигнала на вертикальные проводники и управление переключением выходов. Вывод результата может осуществляться как прямым, так и инвертированным сигналом. Выходной порт может подключаться как TRISTATE-вывод.

16.2.4. FPLA-схемы

FPLA — это сокращение от Field Programming Logic Array (программируемые пользователем логические матрицы). PAL- и GAL-схемы имеют только программируемую И-матрицу (AND Array). В FPLA-схемах вводится дополнительно программируемая ИЛИ-матрица. На рис. 16.10 представлена схема с программируемой ИЛИ-матрицей и программируемой И-матрицей. И-матрица может производить все желаемые полные конъюнкции. С помощью ИЛИ-матрицы они подаются на ИЛИ-элементы с выходами от Q_0 до Q_3. Схема может обеспечить связи для четырех нормальных форм ИЛИ.

На рис. 16.11 показана FPLA-схема. Представлен однопроводной вариант с непроводящими узлами. Схема должна программироваться в полный сумматор.

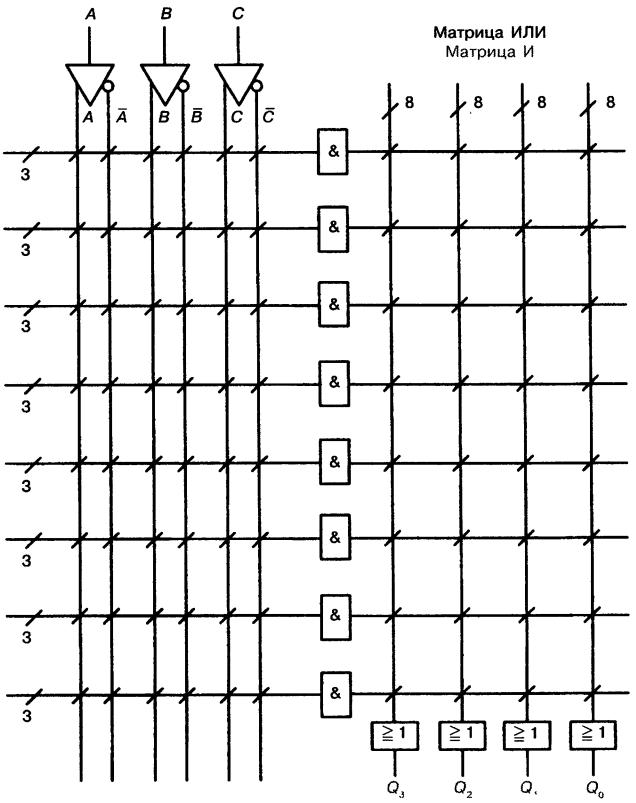


Рис. 16.10. FPLA-схема, программируемые матрица И и матрица ИЛИ. После изготовления узлы непроводящие.

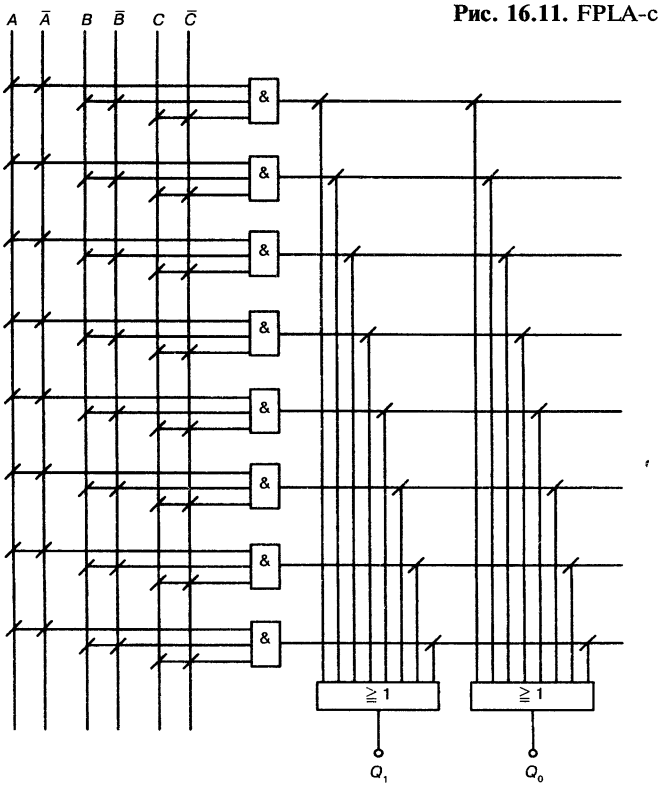


Рис. 16.11. FPLA-схема с 3 входами и 2 выходами.

Пример

Полный сумматор

Вар.	A	B	C	U	Z
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

Нормальные формы ИЛИ

	Ⓩ	Ⓤ
⇒	$\bar{A} \wedge \bar{B} \wedge C$	
⇒	$\bar{A} \wedge B \wedge \bar{C}$	
⇒	----------	$\bar{A} \wedge B \wedge C$
⇒	$A \wedge \bar{B} \wedge \bar{C}$	
⇒	----------	$A \wedge \bar{B} \wedge C$
⇒	----------	$A \wedge B \wedge \bar{C}$
⇒	$A \wedge B \wedge C$	$A \wedge B \wedge C$

Прежде всего необходимо запрограммировать полные конъюнкции. Программируемые узлы И-матрицы переводятся на проводящий режим. Обозначение проводящего режима — крестик. На выходы И-элементов должны быть выведены полные конъюнкции. Они поступают затем на ИЛИ-элементы:

$$Z = \left(\bar{A} \wedge \bar{B} \wedge C\right) \vee \left(\bar{A} \wedge B \wedge \bar{C}\right) \vee \left(A \wedge \bar{B} \wedge \bar{C}\right) \vee (A \wedge B \wedge C);$$

$$\ddot{U} = \left(\bar{A} \wedge B \wedge C\right) \vee \left(A \wedge \bar{B} \wedge C\right) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge B \wedge C).$$

Схема запрограммированного модуля представлена на рис. 16.12. Модуль использован только частично. Ненужные связи отключаются при программировании от электроснабжения.

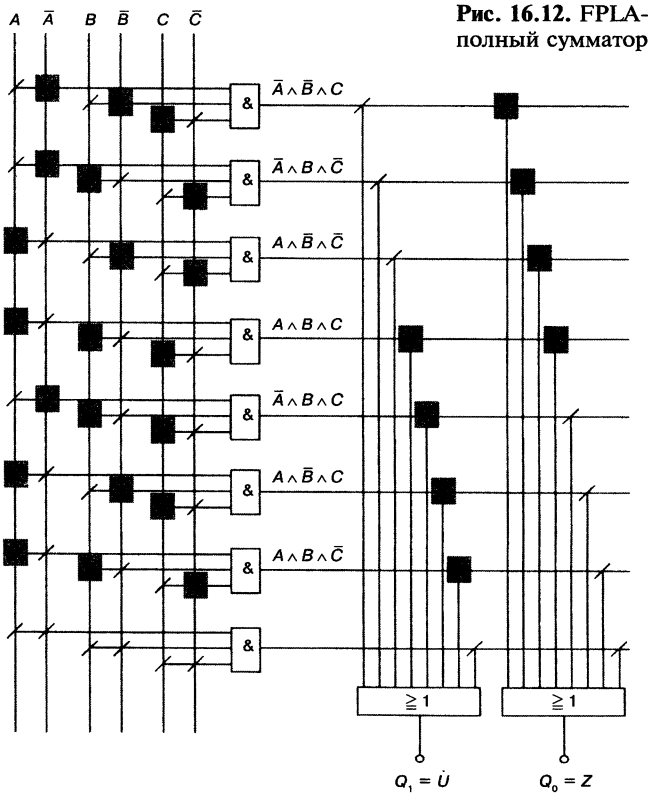


Рис. 16.12. FPLA-схема, запрограммированная как полный сумматор.

16.2.5. PROM-схемы

PROM-схемы уже известны как программируемые ПЗУ (Programmable Read Only Memories) из разд. 12.5. С PROM-схемами также могут осуществляться сложные логические операции. Представленный на рис. 12.32 дешифратор является в принципе И-матрицей. Массив памяти имеет функцию ИЛИ-матрицы. И-матрица является аппаратной, ИЛИ-матрица — программируемой.

PROM-схемы можно использовать как программируемую логику (PLD).

В зависимости от полученного адреса И-матрица переключает полную конъюнкцию на ИЛИ-матрицу (рис. 16.13). ИЛИ-матрица производит нормальную форму ИЛИ.

16.2.6. MACRO-схемы

Развитие IC-технологий позволяет разместить сложные схемы на относительно малом пространстве. Несколько фирм разработали так называемые macrocell-структуры (макроячейки), которые похожи по структуре на большие GAL-схемы и содержат дополнительно большое количество триггеров.

Множество макроячеек соединяются на чипе в одну макросхему.

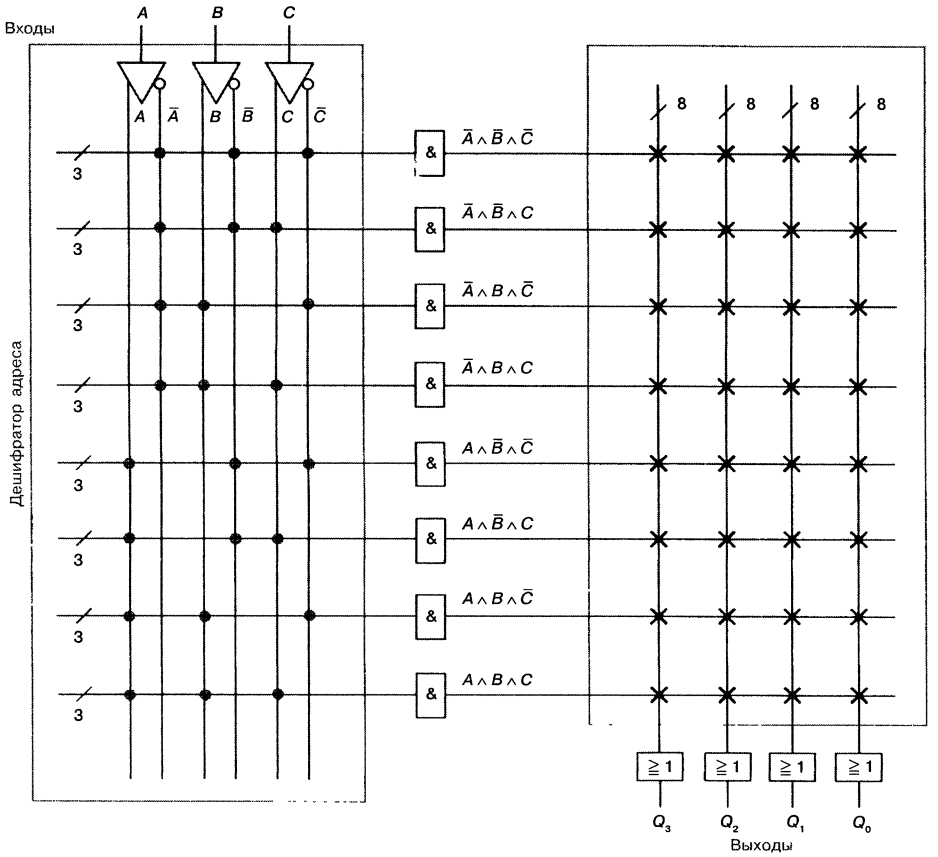


Рис. 16.13. PROM в качестве PLD.

Сегодня существуют макросхемы до 64 макроячеек. Часто используют схемы с 24 макроячейками. Они поставляются в 40-полюсных корпусах. Для очень больших макроячеек корпуса имеют до 289 выводов (рис. 16.14).

> У больших макросхем возникают трудности с охлаждением.

Несмотря на отключение от питания при программировании ненужных элементов, эти схемы отличаются относительно высоким энергопотреблением. При обычном напряжении питания 5 вольт оно составляет до 10 Вт. С развитием новых *N*-МОП- и *K*-МОП-технологий удалось снизить напряжение питания до 3,3 В. Новые макросхемы работают с напряжением питания 3,3 В.

Очень маленькие полевые транзисторы имеют маленькие паразитные емкости и вследствие этого — высокое быстродействие.

> Быстрые макросхемы имеют время задержки 5 нс и менее.

Существующие в настоящее время макросхемы позволяют решить любую логическую задачу программированием со стороны конечного пользо-

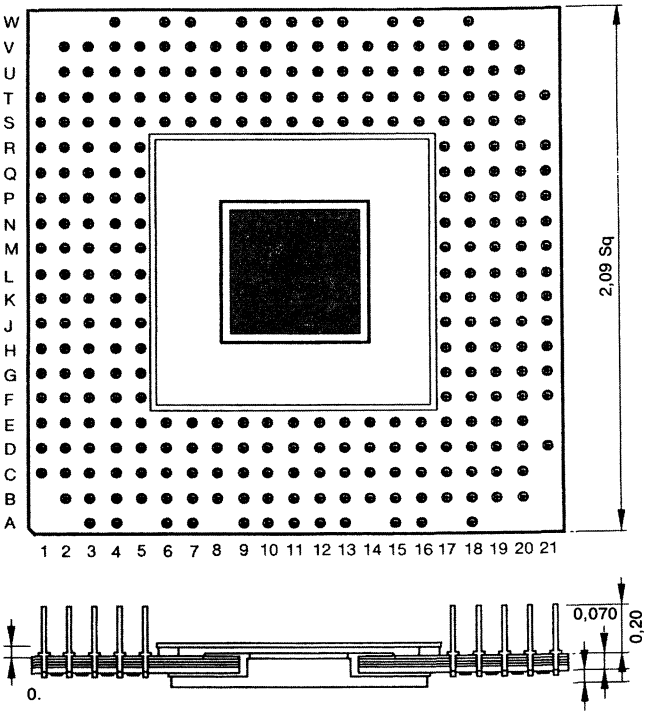


Рис. 16.14. 289-пиновый корпус (Motorola).

вателя. Задачи могут быть как комбинаторными, так и последовательными. Макросхемы из семейства MPA 1000 состоят из 10 000 логических элементов и 2900 триггеров (согласно документации Motorola).

16.3. Разновидности PLD

В рамках конкуренции каждый производитель PLD использует для своей продукции специальные названия. Некоторые из этих имен защищены как товарный знак, некоторые нет.

Рассмотрим сначала схемы, программируемые производителем. Для них используется обозначение ASIC. Фирменными обозначениями являются GATE-ARRAYS, Standard Cells, PLA (Programmable Logic Array, программируемая логическая матрица) и HAL (Hardware Array Logic).

Схемы, программируемые пользователем, называются PLD. Они в основном не относятся к ASIC: если они стираются *UV*-светом, то называются EPLD. Модули, которые могут стираться электрическими импульсами, называются EEPLD. PLD можно разделить на четыре группы, как показано в табл. 16.1.

Из-за быстрого развития в этой области таблица не может претендовать на полноту.

Сложность PLD оценивается в эквивалентных вентилях. Это число указывает количество логических элементов — в пересчете на элементы с двумя входами, которые содержит модуль. Триггеры считаются как два или четыре элемента.

Таблица 16.1. Типы PLD.

Схемы с программируемыми матрицами И (матрицами И-НЕ) в AGA и LCA	Схемы с программируемыми матрицами ИЛИ	Схемы с программируемыми матрицами И и ИЛИ	Макросхемы
PAL GAL EPAL AGA LCA	PROM EPROM EEPROM PLE	FPLA EPL FPLAS FPLS FPGA PL	Macrocell μ PLD Macrochip Macrocell-Array

PLD имеют от 100 до 90 000 эквивалентных вентилей.

Важными параметрами является время задержки t_{PD} и максимальная рабочая частота f_{max}.

Время задержки находится в диапазоне от 50 нс до 1 нс, максимальная рабочая частота — между 20 МГЦ и 400 МГЦ.

Для охлаждения важна рассеиваемая мощность, следовательно, величина напряжения питания и потребление тока на максимальной допустимой рабочей частоте. Большинство PLD производятся в *N*-МОП- или *K*-МОП-технологии. Для модулей с малым временем задержки используются ЭСЛ-технология.

16.4. Программирование PLD

Ручное программирование прожиганием тонких проводников рационально только для очень маленьких PLD. Так было в начале их развития. В сегодняшних сложных PLD программирование производится при помощи компьютеров. Для этого нужно специальное программное обеспечение.

Каждый производитель PLD предлагает собственное программное обеспечение.

Стандарты программирования еще не сформировались. Поэтому покупкой программного обеспечения и оборудования пользователь привязывает себя к определенному производителю. К программному обеспечению прилагаются аппаратные средства. Прежде всего необходима сменная плата, которую нужно устанавливать в компьютер. Кроме того, нужен переходник между *PC*-платой и различными корпусами PLD. Переходник подключается к плате.

Программируемый модуль подключается к соответствующему разъему переходника. Теперь можно программировать. Существующее в настоящее время программное обеспечение очень разнообразно и достаточно удобно в применении.

Программист не обязан подробно знать структуру модуля.

Логическая функция, которая должна осуществляться модулем, может вводиться как уравнение алгебры логики, например как нормальная форма ИЛИ, как таблица истинности или как рисунок. Это имеет значение для комбинационных схем. Для последовательных схем (например счетчика) для облегчения ввода применяется так называемая State Machine. Программа вводится поэтапно.

Ввод программы: в виде уравнения, таблицы истинности или схемы.

При вводе уравнений алгебры логики возникает ряд трудностей. Нормированные знаки для операций И и ИЛИ и знак инверсии НЕ не нанесены на компьютерной клавиатуре.

В зависимости от программного обеспечения используются следующие знаки:

И	$\wedge$	$\Rightarrow$	&	,	·
ИЛИ	$\vee$	$\Rightarrow$	#	,	+
НЕ	–	$\Rightarrow$	!	,	/

Компьютер ищет с помощью программы самое благоприятное решение. При этом проверяется, может ли данный модуль выполнить желаемую логическую функцию. Если нет, то выдается сообщение с просьбой взять более мощный модуль.

После завершения программирования тестовая программа проверяет, выполняет ли модуль желаемую функцию. Если результат проверки положителен, то программирование заканчивается. Посредством составленной программы прожигаются (название напоминает о первых ПЛМ, когда тонкие проводники просто пережигались) следующие модули с тем же содержанием. Можно производить серию аналогичных модулей. Программа прожига может быть сохранена для последующего использования.

Если программирование было сделано с ошибкой, модуль необходимо очистить (стереть). Модули, в которых пережигаются перемычки, придется выбросить. Стертые модули программируются заново.

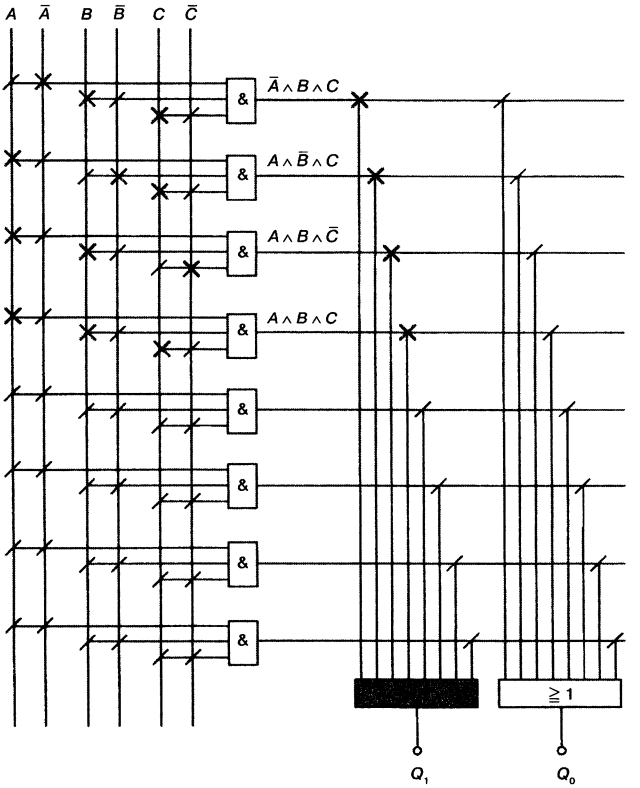
В последнее время появилась возможность перепрограммировать уже встроенный в схему модуль, т. е. целиком или частично удалять содержимое и записывать новое содержимое. Такой модуль должен быть электрически стираемым. Далее должна иметься возможность подключиться к нескольким выводам посредством разъема. Делаются попытки программирования с помощью внешнего электрического поля сквозь корпус. Этот метод находится еще на этапе разработки. PLD ждет интересное будущее.

Контрольный тест

1. Что значит сокращение PLD?
2. Какое принципиальное различие состоит между GAL-модулями и FPLA-модулями?
3. Что значит эквивалентный вентиль?
4. Объясните понятие «масочное программирование».

Вар.	A	B	C	Z
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Вар.	A	B	C	Z	
1	0	0	0	0	
2	0	0	1	0	
3	0	1	0	0	
4	0	1	1	1	$\Rightarrow \bar{A} \wedge B \wedge C$
5	1	0	0	0	
6	1	0	1	1	$\Rightarrow A \wedge \bar{B} \wedge C$
7	1	1	0	1	$\Rightarrow A \wedge B \wedge \bar{C}$
8	1	1	1	1	$\Rightarrow A \wedge B \wedge C$



5. Что такое EPLD, EEPLD?
6. Что значит в PAL-модуле обозначение 12 H 6?
7. Как программируются PLD?
8. Что понимают под программируемым выходом?
9. Объясните структуру макросхемы.
10. Синтезируйте цифровую схему управления, которая производит логические операции согласно данной таблице истинности. Искомая схема является так называемой «схемой 2 из 3». На выходе Z только тогда 1, когда два или три входа имеют состояние 1.

Напишите полные конъюнкции нужной нормальной формы ИЛИ.

FPLA-схема согласно рис. 16.11 должна быть запрограммирована таким образом, чтобы производить желаемые логические операции. Изобразите схему и обозначьте нужные узлы крестами.

ГЛАВА 17

РЕШЕНИЯ ЗАДАНИЙ КОНТРОЛЬНЫХ ТЕСТОВ

В этой заключительной главе приводятся решения расчетных заданий. Ответы на вопросы на понимание можно узнать из текста соответствующей главы. Они приводятся только в том случае, если не изложены в тексте книги.

Глава 1

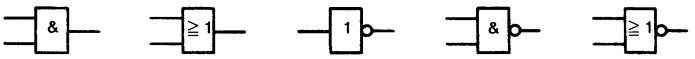
1. Цифровая величина состоит из количественных элементов. Она чаще всего представляет собой бинарную величину 0 или 1. Аналоговая величина может принимать любое значение в пределах допустимого диапазона.

2. Преимущество аналогового представления — наглядность. Недостатки — низкая точность, ошибки при переносе и сохранении аналоговых величин.

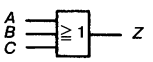
3—7. См. текст книги.

Глава 2

1. Условные графические обозначения:

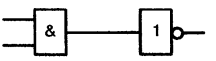


2. Таблица истинности и условное обозначение ИЛИ-элемента с тремя входами

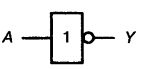


Вар.	C	B	A	Z
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

3. Структура И-НЕ-элемента на базе основных элементов

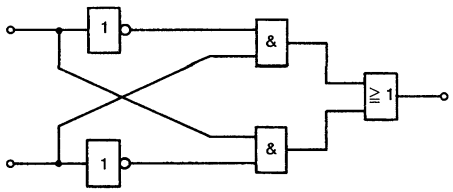


4. Таблица истинности и условное обозначение НЕ-элемента



Вар.	A	Y
1	0	1
2	1	0

5. Структура элемента неравнозначности на базе основных элементов



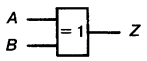
$$Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$$

6—7. См. текст книги.

8. На выходе элемента неравнозначности действует логическая 1 только в том случае, если только на одном входе действует 1.

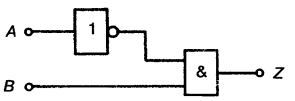
Таблица истинности и условное обозначение XOR-элемента

Вар.	B	A	Z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

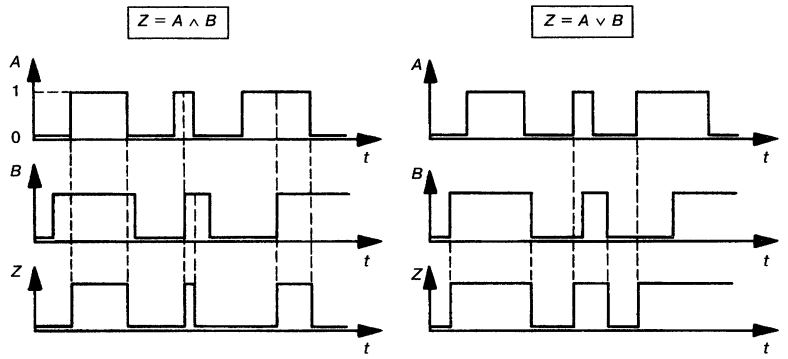


9. Логическую операцию выполняет элемент ИЛИ-НЕ.

10. Операция запрещения является особенной разновидностью умножения И. Состояние входа инвертируется перед логическим умножением И. Элемент запрещения строится на базе основных элементов (запрещение *A*):

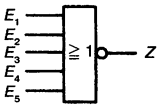


11. Диаграммы умножения И и сложения ИЛИ сигналов *A* и *B*



12. Схема выполняет операцию ИЛИ.

13. Таблица истинности элемента ИЛИ-НЕ и пятью входами:



Вар.	E_5	E_4	E_3	E_2	E_1	Z
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	1	0	0
4	0	0	0	1	1	0
5	0	0	1	0	0	0
6	0	0	1	0	1	0
7	0	0	1	1	0	0
8	0	0	1	1	1	0
9	0	1	0	0	0	0
10	0	1	0	0	1	0
11	0	1	0	1	0	0
12	0	1	0	1	1	0
13	0	1	1	0	0	0
14	0	1	1	0	1	0
15	0	1	1	1	0	0
16	0	1	1	1	1	0
17	1	0	0	0	0	0
18	1	0	0	0	1	0
19	1	0	0	1	0	0
20	1	0	0	1	1	0
21	1	0	1	0	0	0
22	1	0	1	0	1	0
23	1	0	1	1	0	0
24	1	0	1	1	1	0
25	1	1	0	0	0	0
26	1	1	0	0	1	0
27	1	1	0	1	0	0
28	1	1	0	1	1	0
29	1	1	1	0	0	0
30	1	1	1	0	1	0
31	1	1	1	1	0	0
32	1	1	1	1	1	0

14. Элемент производит операцию эквивалентности. На выходе Z действует 1-уровень, если состояния входов равны.

Глава 3

1. Таблица истинности для схемы рис. 3.13:

Вар.	B	A	$\bar{A}$	$\bar{A} \vee B$	$Z = \overline{\bar{A} \vee B}$
1	0	0	1	1	0
2	0	1	0	0	1
3	1	0	1	1	0
4	1	1	0	1	0

2. Таблица истинности для схемы на рис. 3.14:

Вар.	C	B	A	$\bar{A}$	$\bar{A} \wedge B$	$A \vee C$	$Z = (\bar{A} \wedge B) \wedge (A \vee C)$
1	0	0	0	1	0	0	0
2	0	0	1	0	0	1	0
3	0	1	0	1	1	0	0
4	0	1	1	0	0	1	0
5	1	0	0	1	0	1	0
6	1	0	1	0	0	1	0
7	1	1	0	1	1	1	1
8	1	1	1	0	0	1	0

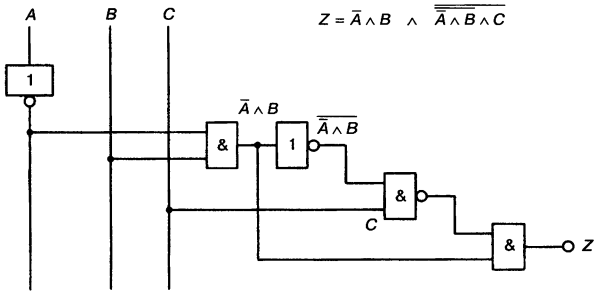
3. Таблица требуемых логических операций схемы, содержащей ошибку, на рис. 3.14:

Вар.	C	B	A	$\bar{A}$	$\bar{A} \wedge B$	$A \vee C$	Z
1	0	0	0	1	1	0	0
2	0	0	1	0	1	1	1
3	0	1	0	1	1	0	0
4	0	1	1	0	1	1	1
5	1	0	0	1	1	1	1
6	1	0	1	0	1	1	1
7	1	1	0	1	1	1	1
8	1	1	1	0	1	1	1

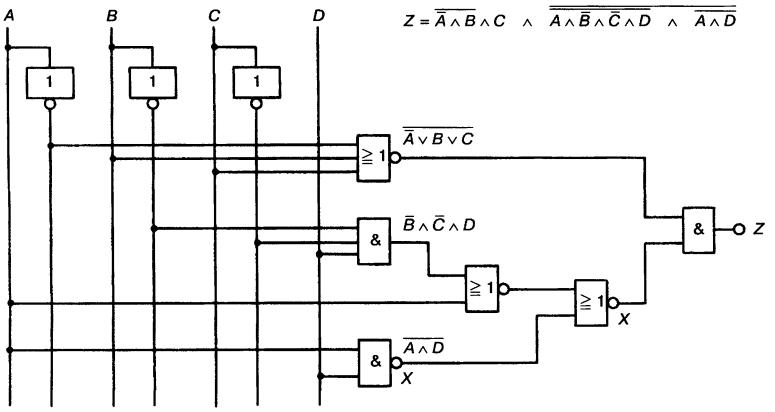
4. $Z = \left[\left(\bar{A} \wedge \bar{B} \wedge \bar{C}\right) \vee \left(A \wedge B \wedge C\right)\right] \wedge \overline{A \vee \bar{B} \vee \bar{C}}$

Вар.	C	B	A	$\bar{C}$	$\bar{B}$	$\bar{A}$	$\bar{A} \wedge \bar{B} \wedge \bar{C}$	$A \wedge B \wedge C$	X	$A \wedge \bar{B} \wedge \bar{C}$	Y	Z
1	0	0	0	1	1	1	1	0	1	1	0	0
2	0	0	1	1	1	0	0	0	0	1	0	0
3	0	1	0	1	0	1	0	0	0	1	0	0
4	0	1	1	1	0	0	0	0	0	1	0	0
5	1	0	0	1	1	1	0	0	0	1	0	0
6	1	0	1	1	1	0	0	0	0	1	0	0
7	1	1	0	1	0	1	0	0	0	0	0	0
8	1	1	1	1	0	0	0	1	1	1	0	0

5. $Z = \bar{A} \wedge B \wedge \overline{\overline{\bar{A} \wedge B} \wedge C}$



6. $Z = \overline{\bar{A} \vee B \vee C} \wedge \overline{\overline{A \vee \bar{B} \wedge \bar{C} \wedge D} \vee \overline{A \wedge D}}$



Вар.	D	C	B	A	$\overline{A}$	$\overline{B}$	$\overline{C}$	$\overline{A} \vee B \vee C$	$\overline{\overline{A} \vee B \vee C}$	$B \wedge \overline{C} \wedge D$	$A \vee \overline{B} \wedge \overline{C} \wedge D$	$\overline{A \vee \overline{B} \wedge \overline{C} \wedge D}$	$\overline{A \wedge D}$	X	Z
1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0
2	0	0	0	1	0	1	1	0	1	0	1	0	1	0	0
3	0	0	1	0	1	0	1	1	0	0	0	1	1	0	0
4	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0
5	0	1	0	0	1	1	0	1	0	0	0	1	1	0	0
6	0	1	0	1	0	1	0	1	0	0	1	0	1	0	0
7	0	1	1	0	1	0	0	1	0	0	0	1	1	0	0
8	0	1	1	1	0	0	0	1	0	0	1	0	1	0	0
9	1	0	0	0	1	1	1	1	0	1	1	0	1	0	0
10	1	0	0	1	0	1	1	0	1	1	1	0	0	1	1
11	1	0	1	0	1	0	1	1	0	0	0	1	1	0	0
12	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0
13	1	1	0	0	1	1	0	1	0	0	0	1	1	0	0
14	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	0	0	1	0	0	0	1	1	0	0
16	1	1	1	1	0	0	0	1	0	0	1	0	1	1	0

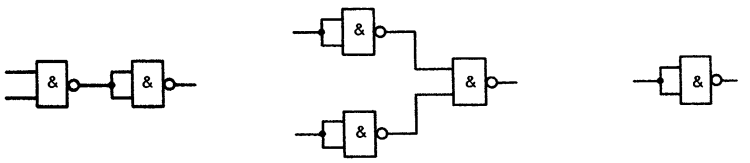
7. Элемент 4 (ИЛИ-НЕ) работает с ошибкой.

Глава 4

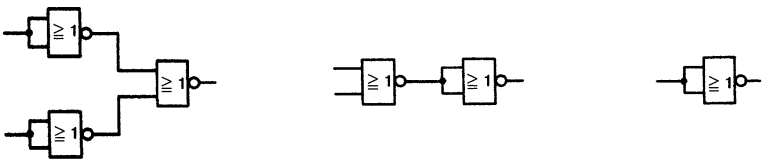
1—7. См. текст книги.

8.

а)



б)



9. а) $Z = 0$;
б) $Y = 1$;
в) $X = \overline{A} \wedge B$;
г) $Q = 1$;
д) $S = \overline{A \wedge B}$.

10. **И-НЕ**

а) $Z = \overline{\overline{\overline{A \wedge S \wedge R} \wedge Q \wedge \overline{C} \wedge \overline{B}}}$;

б) $Y = \overline{\overline{\overline{A} \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}}}$;

в) $X = \overline{\overline{\overline{A} \wedge \overline{B} \wedge \overline{C} \wedge \overline{M \wedge N \wedge P} \wedge \overline{\overline{R} \wedge \overline{S}}}}$;

г) $Q = \overline{\overline{\overline{A} \wedge B \wedge \overline{C} \wedge \overline{D} \wedge \overline{S} \wedge \overline{R}}}$;

д) $Q = \overline{\overline{A \wedge \overline{B} \wedge \overline{C} \wedge D} \wedge \overline{P \wedge Q \wedge S}}$.

ИЛИ-НЕ

а) $Z = \overline{\overline{\overline{A} \vee \overline{S} \vee \overline{R} \vee \overline{Q} \vee C \vee B}}$;

б) $Y = \overline{A \vee B \vee C \vee D}$;

в) $X = \overline{\overline{A \vee B \vee C} \vee \overline{\overline{M} \vee \overline{N} \vee \overline{P}} \vee \overline{R \vee S}}$;

г) $Q = \overline{\overline{A \vee \overline{B} \vee C \vee D \vee S \vee R}}$;

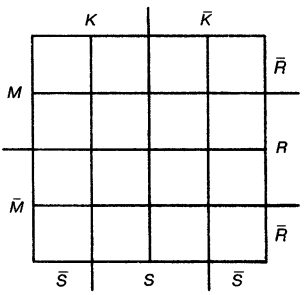
д) $Q = \overline{\overline{\overline{A} \vee B \vee C \vee \overline{D}} \vee \overline{\overline{P} \vee \overline{Q} \vee \overline{S}}}$.

Глава 5

1—3. См. текст книги.

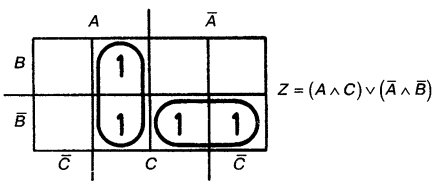
4. $Z = \left(A \wedge \overline{B} \wedge \overline{C}\right) \vee \left(A \wedge B \wedge \overline{C}\right) \vee \left(\overline{A} \wedge \overline{B} \wedge C\right) \vee \left(\overline{A} \wedge B \wedge C\right)$.

5. Карта Карно для переменных K, M, S и R.

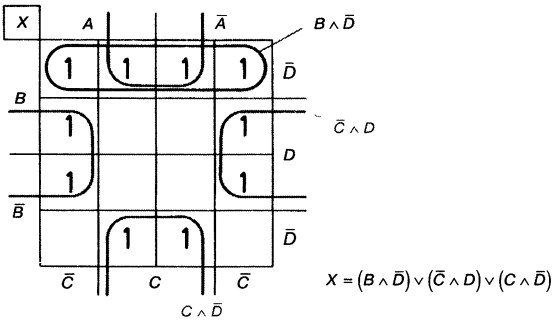


6. См. текст книги.

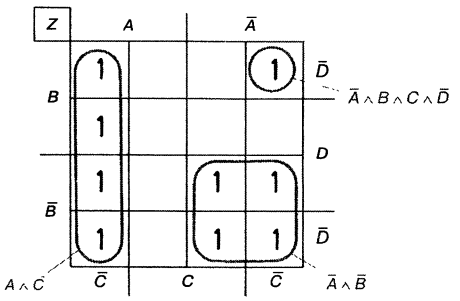
7.



8.



9.



10. См. текст книги.

Глава 6

1—5. См. текст книги.

6. Рабочая таблица схемы на рис. 6.108:

Вар.	B	A	Z
1	L	L	L
2	L	H	L
3	H	L	L
4	H	H	H

7. Положительная логика: умножение И.
Отрицательная логика: сложение ИЛИ.

8—14. См. текст книги.

15. Схема производит И-умножение.

16. В «насыщенном семействе» транзисторы управляются в насыщенном состоянии. Благодаря этому получаются удобные логические уровни и хорошая помехоустойчивость. Энергопотребление незначительно, скорость удовлетворительная. Если транзисторы работают в ненасыщенном состоянии, то энергопотребление повышается. Логические уровни не так благоприятны. Скорости переключения возрастают и, следовательно, растет быстродействие схемы. Такие схемы называются ненасыщенными (Пример: ЭСЛ-схемы).

17—19. См. текст книги.

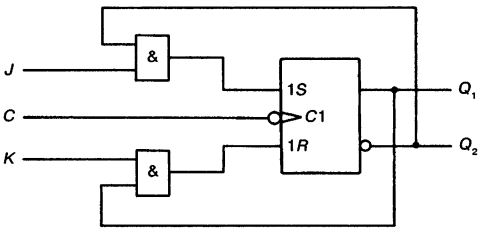
20. Схема на рис. 6.94, см. текст книги.

Глава 7

1. На условном обозначении изображен триггер с особенным режимом работы. Если оба входа имеют состояние 1, то верхний выход (например A_1) имеет состояние 1. Нижний выход (например A_2) имеет состояние 0. Вход сигнала установки (например E_1) доминирует.

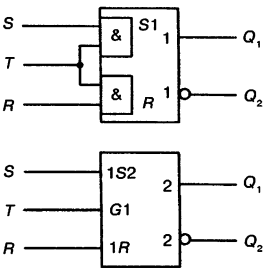
2. См. текст книги.

3.



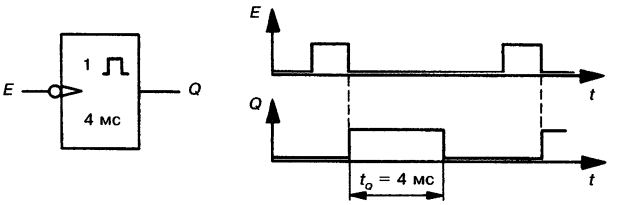
4. Таблица истинности и условное обозначение *RS*-триггера с доминирующим *S*-входом, управляемого уровнем сигнала. Принцип действия см. текст книги.

	t_n		t_{n+1}
Вар.	R	S	Q_1
1	0	0	Q_{1n}
2	0	1	1
3	1	0	0
4	1	1	1



5. См. текст книги.

6. Временная диаграмма и условное обозначение моностабильной ячейки с t_Q = 4 мс.



7. Уравнение является характеристическим уравнением *JK*-триггера.

8. Подробная таблица истинности и характеристическое уравнение триггера.

	t_n			t_{n+1}	
Вар.	E_2	E_1	Q_1	Q_1	
1	0	0	0	0	
2	0	0	1	1	Хранение
3	0	1	0	0	
4	0	1	1	1	Хранение
5	1	0	0	0	
6	1	0	1	0	Сброс
7	1	1	0	1	
8	1	1	1	1	Установка

$$Q_{1(n+1)} = \left[\left(E_1 \wedge E_2\right) \vee \left(\bar{E}_2 \wedge Q_1\right)\right]_n$$

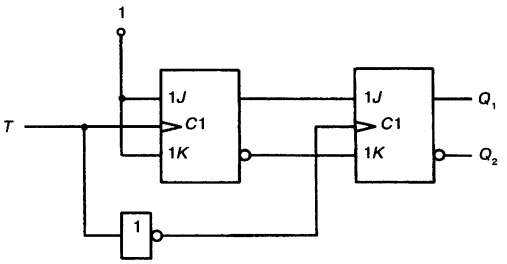
Вход E_2 является входом подготовки. Триггер заблокирован, если на E_2 действует 0-сигнал (режим хранения). Если на E_2 действует 1-сигнал, то триггер работает как *D*-триггер — с E_1 в качестве *D*-входа. Такой триггер также называется *DV*-триггер. (Обозначения входов: $E_2 = V$, $E_1 = D$.)

9. См. текст книги.

10. Триггер является *JK*-MASTER-SLAVE-триггером с тремя *J*-входами и тремя *K*-входами, нетактируемым входом сигнала установки *S* и нетактируемым входом сброса *R*. Верхний из трех *J*-входов инвертируется. Он срабатывает, следовательно, при 0-сигнале. *J*-входы связаны логическим умножением в общий *J*-вход, нижний их трех *K*-входов инвертируется. *K*-входы связаны логическим умножением в общий *K*-вход.

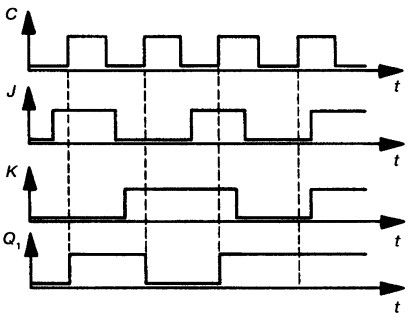
11. См. текст книги.

12. Структура *T*-MASTER-SLAVE-триггера

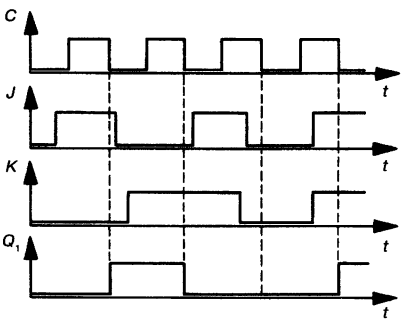


13. Временные диаграммы:

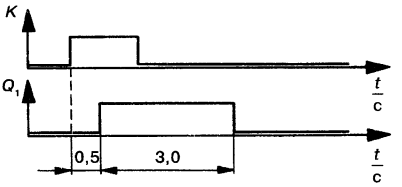
а) триггер переключается с передним фронтом синхроимпульса



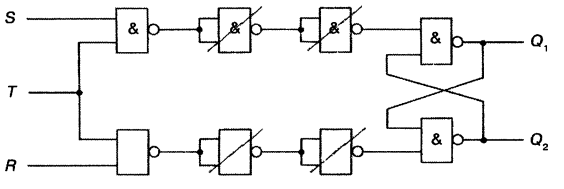
б) триггер переключается с задним фронтом синхроимпульса

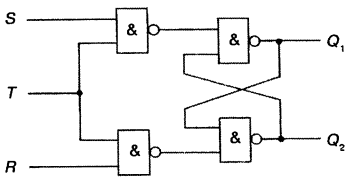


14. На условном обозначении изображена моностабильная ячейка с задержкой 0,5 секунды. Время нахождения в нестабильном состоянии составляет три секунды.



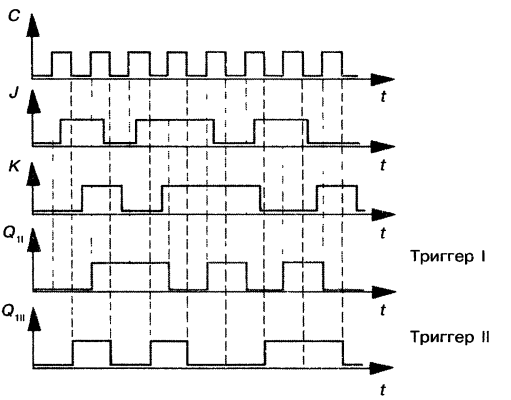
15. Управляемый уровнем синхроимпульса *RS*-триггер, построенный на базе элементов И-НЕ.



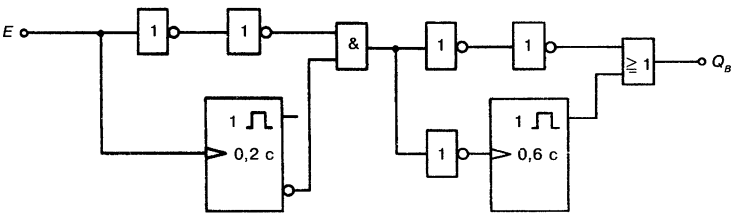


16. См. текст книги.

17. Временные диаграммы.



18. Время нахождения в нестабильном состоянии моностабильных ячеек составляет 0,2 секунды и 0,6 секунды.



Глава 8

1. См. текст книги.

2.

Десятичное число	Двоичное число												
	2^{12} 4096	2^{11} 2048	2^{10} 1024	2^{9} 512	2^{8} 256	2^{7} 128	2^{6} 64	2^{5} 32	2^{4} 16	2^{3} 8	2^{2} 4	2^{1} 2	2^{0} 1
50								1	1	0	0	1	0
215						1	1	0	1	0	1	1	1
172						1	0	1	0	1	1	0	0
688				1	0	1	0	1	1	0	0	0	0
909				1	1	1	0	0	0	1	1	0	1
1820			1	1	1	0	0	0	1	1	1	0	0
3276		1	1	0	0	1	1	0	0	1	1	0	0
2423		1	0	0	1	0	1	1	1	0	1	1	1
6052	1	0	1	1	1	1	0	1	0	0	1	0	0
6381	1	1	0	0	0	1	1	1	0	1	1	0	1
5511	1	0	1	0	1	1	0	0	0	0	1	1	1
7732	1	1	1	1	0	0	0	1	1	0	1	0	0

3.

Десятичное число	Двоичное число															
	2^{15} 32768	2^{14} 16384	2^{13} 8192	2^{12} 4096	2^{11} 2048	2^{10} 1024	2^{9} 512	2^{8} 256	2^{7} 128	2^{6} 64	2^{5} 32	2^{4} 16	2^{3} 8	2^{2} 4	2^{1} 2	2^{0} 1
58											1	0	1	0	1	0
512							1	0	0	0	0	0	0	0	0	0
1298						1	0	1	0	0	0	1	0	0	1	0
1983						1	1	1	1	0	1	1	1	1	1	1
20000		1	0	0	1	1	1	0	0	0	1	0	0	0	0	0
17750		1	0	0	0	1	0	1	0	1	0	1	0	1	1	0
2730					1	0	1	0	1	0	1	0	1	0	1	0
9990			1	0	0	1	1	1	0	0	0	0	0	1	1	0
11000			1	0	1	0	1	0	1	1	1	1	1	0	0	0
32000		1	1	1	1	1	0	1	0	0	0	0	0	0	0	0

4.
а) 54,625;
б) 37,8125;
в) 10,90625;
г) 0,65625;
д) 0,453125.

5.
а) 10001;
б) 1000110;
в) 111111;
г) 1001110;
д) 10100011;
е) 10000111;
ж) 10000,10;
з) 10100,00.

6.
а) 1001;
б) 110100;
в) 1100;
г) 100010;
д) 10000;
е) 11001;
ж) 100101;
з) $|11001| = -111$.

7.

Десятичное число	Двоично-десятичное число				
а) 10941	1	0000	1001	0100	0001
б) 3890		11	1000	1001	0000
в) 7863		111	1000	0110	0011
г) 98001	1001	1000	0000	0000	0001
д) 7989		111	1001	1000	1001

8.
а) |0111;
б) 1|0100;
в) 1|0110;
г) |1001;
д) 1|0111;
е) 1|0000;
ж) 1|0010;
з) 1|0101.

9.

а) 0001;
б) 0001;
в) 0001;
г) 0010;
д) 0100;
е) –0010;
ж) 0101;
з) –0101.

10.

Десятичное число		Двоичное число				
а)	2737			1010	1011	0001
б)	34802		1000	0111	1111	0010
в)	58885		1110	0110	0000	0100
г)	48340		1011	1100	1101	0100
д)	76593	1	0010	1011	0011	0001
е)	47642		1011	1010	0001	1010
ж)	201817	11	0001	0100	0101	1001
з)	6683		1	1010	0001	1011

11.

Шестнадцатеричное число		Двоичное число			
а)	16			110	0100
б)	103		1	0000	0011
в)	3FC		11	1111	1100
г)	7BF		111	1011	1111
д)	2710	10	0111	0000	0000
е)	7E			111	1110
ж)	4664	100	0110	0110	0100
з)	3E7		11	1110	0111

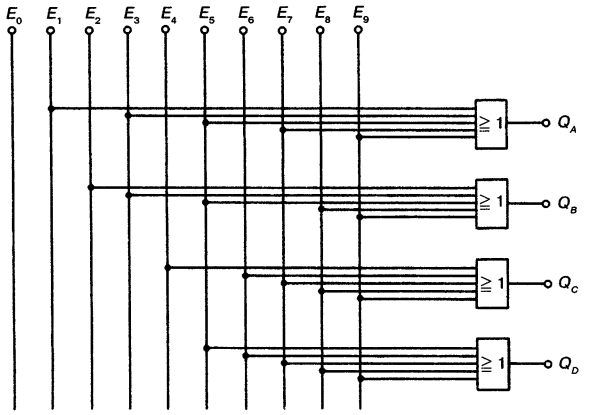
12.

Десятичное число	Двоичное число					Шестнадцатеричное число	Восьмеричное число	Двоично-десятичное число				
2560			1010	0000	0000	A00	5000		10	0101	0110	0000
1270			100	1111	0110	4F6	2366		1	0010	0111	0000
44854		1010	1111	0011	0110	AF36	127466	100	0100	1000	0101	0100
1018			11	1111	1010	3FA	1772		1	0000	0001	1000
39718		1001	1011	0010	0110	9B26	115446	11	1001	0111	0001	1000
107196	1	1010	0010	1011	1100	1A2BC	321274	1	0000	0111	0001	1001	0110

13—20. См. текст книги.

Глава 9

1. Рис. 9.8, см. текст книги.
2. См. текст книги.
3. Схема преобразователя кода из десятичного кода в код Айкена.

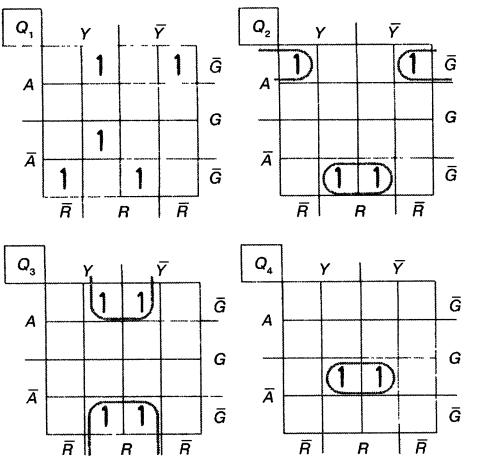


Десятичная цифра	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Код Айкена

4. Схема преобразователя кода Грея в BCD-код:

Десятичная цифра	Вход кода Грея				Выход BCD-кода			
	G	R	A	Y	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
					(2^3)	(2^2)	(2^1)	(2^0)



$$Q_1 = \left(Y \wedge \bar{A} \wedge \bar{R} \wedge \bar{G}\right) \vee \left(\bar{Y} \wedge A \wedge \bar{R} \vee \bar{G}\right) \vee \left(Y \wedge A \wedge R \wedge \bar{G}\right) \vee$$
$$\vee \left(\bar{Y} \wedge \bar{A} \wedge R \wedge \bar{G}\right) \vee \left(Y \wedge \bar{A} \wedge R \wedge G\right);$$

$$Q_2 = \left(A \wedge \bar{G} \wedge R\right) \vee (A \wedge R \wedge G);$$

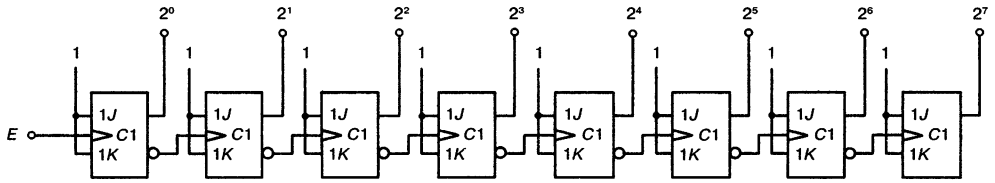
$$Q_3 = (G \wedge R);$$

$$Q_4 = \bar{A} \wedge G \wedge R.$$

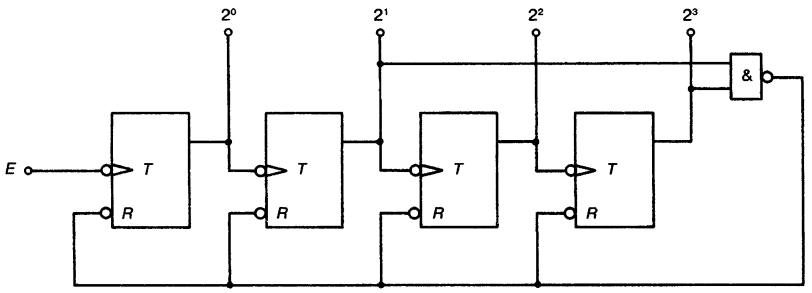
5—6. См. текст книги.

Глава 10

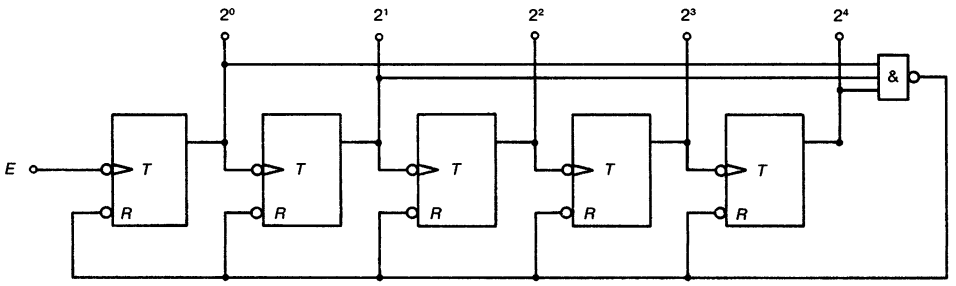
1. См. текст книги.
2. Асинхронный 8-битовый суммирующий счетчик



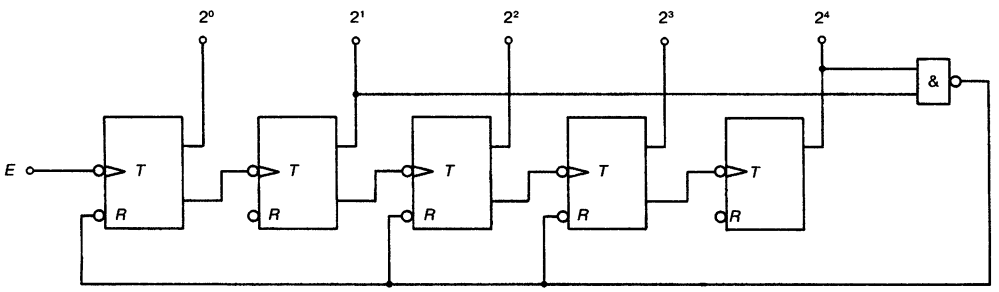
3. После того, как счетчик досчитает до 1010 = $10_{(10)}$, он должен быть сброшен на нуль. Сброс производится с помощью И-НЕ-элемента. При 1010 на входе элемента И-НЕ действует 0-сигнал. Этот сигнал инициализирует сброс через *R*-входы.



4. См. текст книги.
5. Счетчик по модулю 19:



6. Преобразование счетчика по модулю 19 в вычитающий счетчик:

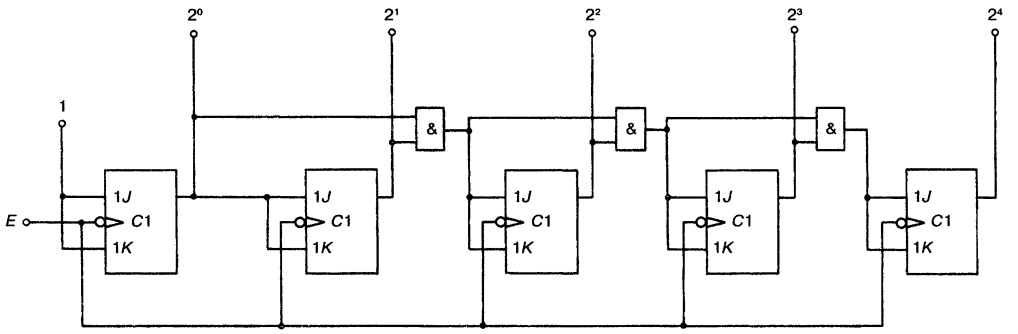


7. Счетчик на рис. 10.68 является реверсивным. Он работает при $X = 0$ на сложение, а при $X = 1$ — на вычитание.

8. См. текст книги.

9. Решение см. рис. 10.9.

10.

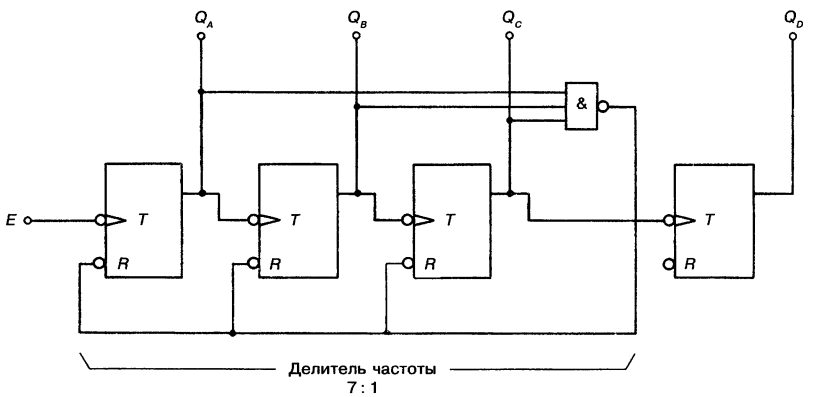


11. См. текст книги.

12. $\bar{Q}$-выходы 4-битового синхронного двоичного суммирующего счетчика нужно использовать в качестве выходов результата. Если $\bar{Q}$-выходы недоступны, надо инвертировать прямые Q-выходы.

13. С выхода с весом 2^2 (3-й триггер) можно снять уменьшенную в 8 раз частоту входа.

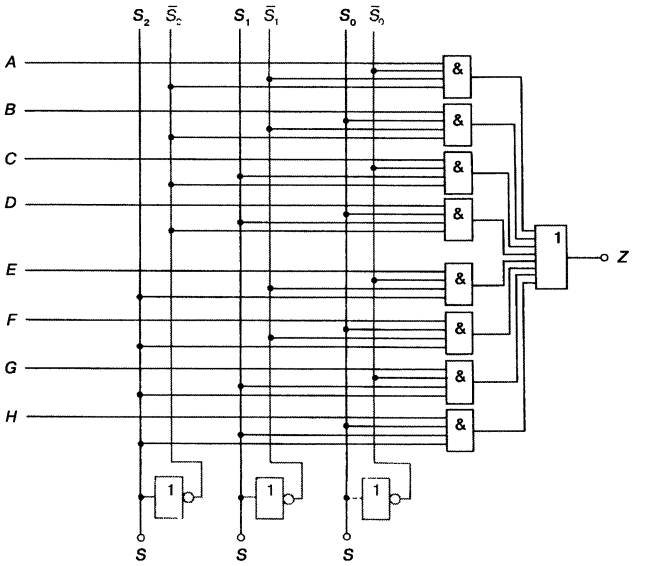
14. Делитель частоты с коэффициентом 14 : 1.



Глава 11

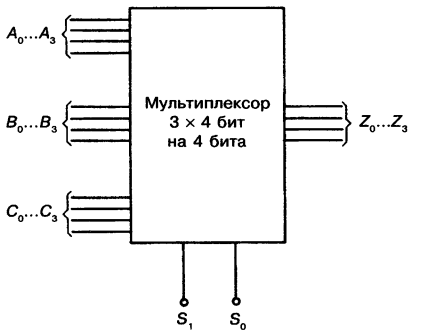
1—2. См. текст книги.

3. Мультиплексор «8 к 1».

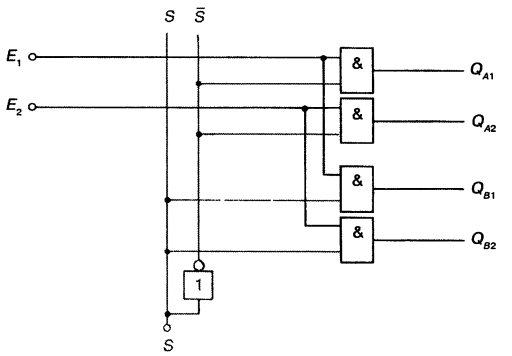


4. Мультиплексор 3 × 4 бит на 4 бита.

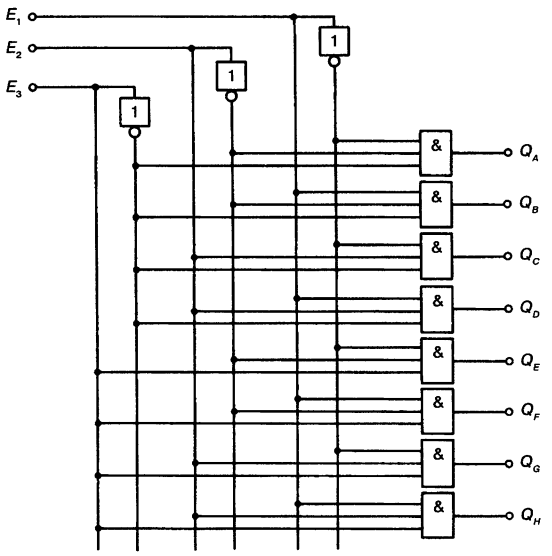
Сигналы $A_0...A_3$, $B_0...B_3$ и $C_0...C_3$ выборочно через И-элемент подключаются к выходу $Z_0...Z_3$. Нужно иметь два управляющих входа, так как три необходимы три различных команды управления.



5. Схема демультиплексора 2 на 2 × 2:



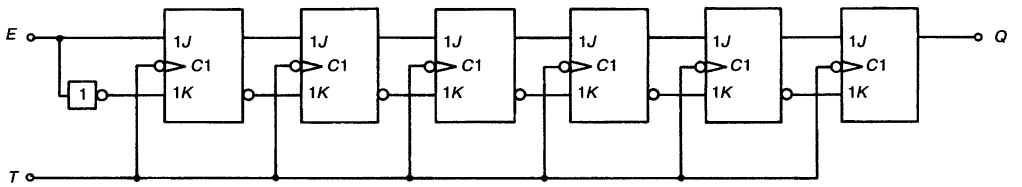
6. См. текст книги.
7. Схема 3-битового адресного формирователя:



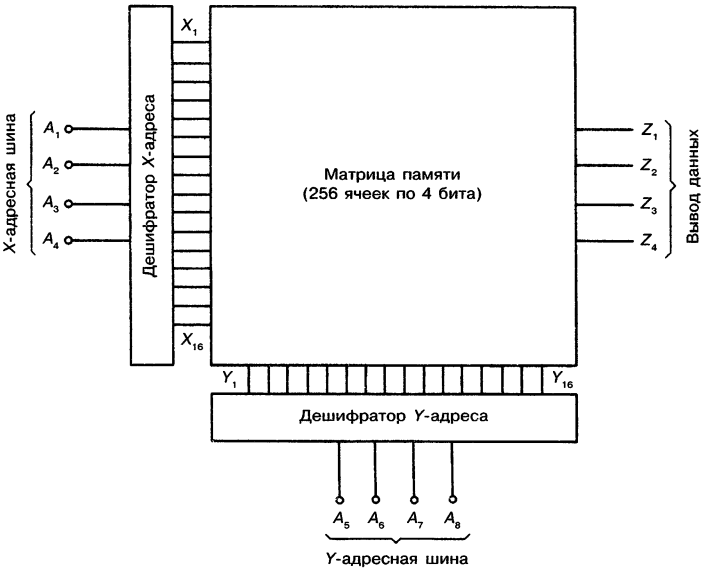
8—10. См. текст книги.

Глава 12

1. Схема 6-битового сдвигающего регистра для последовательного ввода—вывода данных:

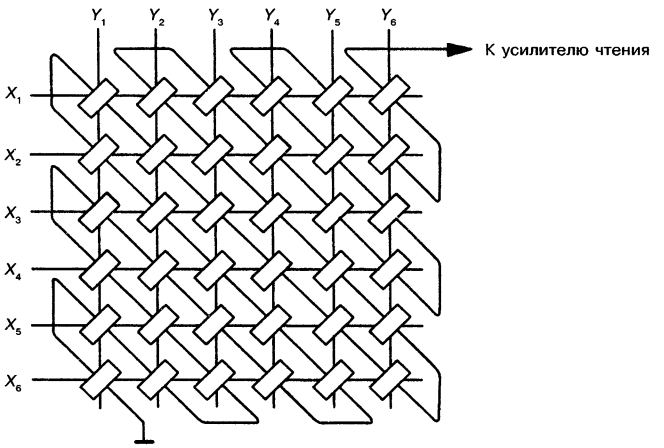


2—5. См. текст книги.
6. Требуемая схема представлена на рис. 12.18.
7. См. текст книги.
8. С 4 *X*-адресными линиями могут адресоваться по 16 адресам 16 *X*-координатных линий. Требуется *X*-дешифратор адреса. С 4 *Y*-адресными линиями могут адресоваться по 16 адресам 16 *Y*-координатных линий. Каждая из 256 ячеек памяти имеет объем памяти 4 бита.



9—13. См. текст учебника.

14. Структура матрицы на магнитных кольцах 6 × 6 бит:



15—18. См. текст книги.

Глава 13

1. Аналого-цифровой преобразователь представлен на рис. 13.3. Принцип действия см. текст книги.

2—7. См. текст учебника.

Глава 14

1. Таблица истинности показана на рис. 14.1. Схема с основными элементами показана на рис. 14.2.

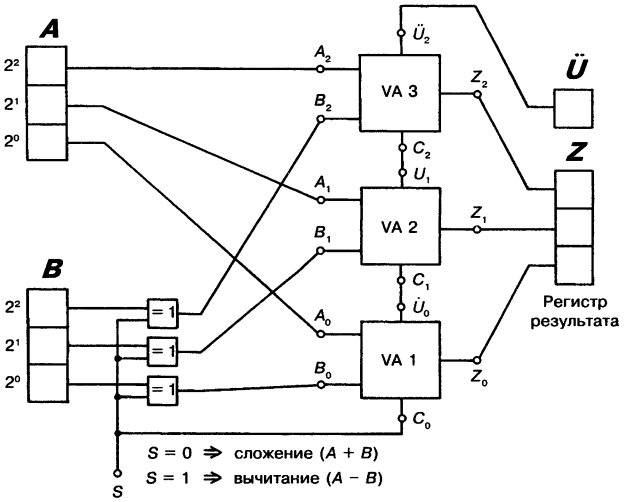
2. См. текст книги.

3. Схема представлена на рис. 14.7.

4—5. См. текст книги.

6. Схема полувычитателя представлена на рис. 14.14. Комментарии к принципу действия см. текст книги.

7. Схема 3-битового сумматора-вычитателя.



8. 1-битовый умножитель производит логическое умножение И. Таблица истинности 1-битового умножителя представлена на рис. 14.25.

9—10. См. текст книги.

Глава 15

1—3. См. текст книги.

4. Схема аккумулятора с арифметико-логическим устройством, регистрами и памятью переноса показана на рис. 15.8.

5—12. См. текст книги.

Глава 16

1—9. См. текст книги.

10. Таблица и рис. 16.16 со стр. 460.

Заявки на книги присылайте по адресу:
125319 Москва, а/я 594
Издательство «Техносфера»
e-mail: knigi@technosphera.ru
sales@technosphera.ru
факс: (495) 956 33 46

В заявке обязательно указывайте
свой почтовый адрес!

Подробная информация о книгах на сайте
http://www.technosphera.ru

Бойт Клаус

Цифровая электроника

Редактор – Т.Н. Зазаева
Компьютерная верстка – Н.А. Попова
Дизайн – И.А. Куколева
Корректор – Н.А. Шипиль
Выпускающий редактор – О.Н. Кулешова
Ответственный за выпуск – С.В. Зинюк

Формат 70x100/16. Печать офсетная.
Гарнитура Ньютон
Печ.л. 29,5. Тираж 3000 экз. (1-й завод 1500 экз.) Зак. № 2114.
Бумага офсет №1, плотность 65г/м^2 .

Издательство «Техносфера»
Москва, Лубянский проезд, 27/1

Диапозитивы изготовлены ООО «Европолиграфик»

Отпечатано в ООО ПФ «Полиграфист»
160001, г. Вологда, ул. Челюскинцев, дом 3