

А. М. ВОДОВОЗОВ

ЭЛЕМЕНТЫ СИСТЕМ АВТОМАТИКИ

Допущено

*Учебно-методическим объединением
по образованию в области энергетики и электротехники
в качестве учебного пособия для студентов
высших учебных заведений, обучающихся по специальности
140604 «Электропривод и автоматика промышленных установок
и технологических комплексов» направления подготовки
140600 «Электротехника, электромеханика и электротехнологии»*



Москва
Издательский центр «Академия»
2006

УДК 681.5(075.8)
ББК 32.965.я73
В624

Рецензенты:

директор Института электропривода и силовой электроники
Таллиннского технического университета,
д-р техн. наук, проф. *Ю. Я. Лаугис*;
канд. техн. наук, проф. кафедры АЭП МЭИ *С. К. Козырев*

Водовозов А. М.

В624 Элементы систем автоматики : учеб. пособие для студ. высш. учеб. заведений / А. М. Водовозов. — М. : Издательский центр «Академия», 2006. — 224 с.

ISBN 5-7695-2934-2

Рассмотрены теоретические основы и элементная база современной цифровой автоматики. Исходные постулаты теории информации, алгебры, логики и правила двоичной арифметики связаны воедино с элементами систем автоматики: стандартной логикой, современными программируемыми логическими схемами, микроконтроллерами, цифровыми датчиками, средствами ввода и вывода информации. Приведены многочисленные примеры построения отдельных схем и узлов автоматики на основе современной элементной базы, созданной ведущими мировыми производителями.

Для студентов высших учебных заведений.

УДК 681.5(075.8)
ББК 32.965.я73

*Оригинал-макет данного издания является собственностью
Издательского центра «Академия», и его воспроизведение любым способом
без согласия правообладателя запрещается*

© Водовозов А. М., 2006

© Образовательно-издательский центр «Академия», 2006

ISBN 5-7695-2934-2

© Оформление. Издательский центр «Академия», 2006

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5
Глава 1. Логические основы автоматики	7
1.1. Информация и общие принципы ее преобразования	7
1.2. Способы кодирования информации	10
1.2.1. Системы счисления	10
1.2.2. Кодирование числовой информации	14
1.2.3. Специальные кодировки	17
1.3. Формальные правила двоичной арифметики	22
1.4. Алгебра логики	24
1.4.1. Логические функции	24
1.4.2. Законы и теоремы алгебры логики	27
1.4.3. Описание логических функций	27
1.5. Теория автоматов	31
1.5.1. Абстрактные цифровые автоматы	31
1.5.2. Матричное описание автомата	33
1.5.3. Графическое описание автомата	35
1.5.4. Структурные автоматы	36
Глава 2. Цифровая схемотехника	46
2.1. Классификация цифровых схем	46
2.2. Основные параметры цифровых микросхем	47
2.2.1. Статические параметры	47
2.2.2. Динамические параметры	51
2.3. Технологии производства цифровых микросхем	53
2.4. Специальные входные и выходные каскады	59
2.5. Стандартная логика	62
2.5.1. Логические элементы	63
2.5.2. Буферные элементы	64
2.5.3. Комбинационные логические схемы	65
2.5.4. Триггеры	77
2.5.5. Последовательные логические схемы	82
2.6. Программируемая логика	93
2.6.1. Микросхемы GA	93
2.6.2. Микросхемы PLD	96
2.6.3. Программирование логики	101
2.7. Микросхемы памяти	102
2.7.1. Классификация	103

2.7.2. Микросхемы ROM	104
2.7.3. Микросхемы RAM	126
Глава 3. Цифроаналоговые и аналого-цифровые преобразователи	147
3.1. Цифроаналоговые преобразователи	147
3.2. Аналого-цифровые преобразователи	154
3.3. Цифровые потенциометры	161
Глава 4. Микроконтроллеры	164
4.1. Структура микроконтроллера	164
4.2. Память	166
4.3. Процессорное ядро	167
4.4. Функциональные модули	171
4.4.1. Тактовый генератор	171
4.4.2. Схема сброса	172
4.4.3. Система прерываний	173
4.4.4. Порты ввода-вывода	174
4.4.5. Энергонезависимая память данных	176
4.4.6. Аналого-цифровые преобразователи	176
4.4.7. Аналоговые компараторы	177
4.4.8. Таймеры-счетчики	178
4.4.9. Последовательные интерфейсы	179
4.4.10. Сторожевой таймер	181
4.4.11. Системы энергосбережения	181
4.5. Сравнительная характеристика микроконтроллеров	182
Глава 5. Датчики	186
5.1. Классификация датчиков	186
5.2. Датчики угловой скорости	187
5.3. Датчики положения	193
5.4. Датчики тока	196
5.5. Датчики напряжения	199
5.6. Датчики температуры	200
Глава 6. Средства человекомашинного интерфейса	204
6.1. Клавиатуры	204
6.2. Устройства индикации	206
Заключение	216
Список литературы	217

ПРЕДИСЛОВИЕ

Современный электропривод, являясь сложной электромеханической системой, стремительно развивается. Совершенствуются электрические машины, появляются новые преобразователи электрической энергии, усложняются законы управления, меняется элементная база. В решении задач управления электроприводами ведущее место занимает цифровая техника, а вопросы цифровой автоматики становятся обязательной частью образовательной программы подготовки инженеров по электрическому приводу.

В данном учебном пособии рассмотрены основные теоретические аспекты развития аппаратных средств и современная элементная база промышленной автоматики.

Основой при написании учебного пособия послужили конспекты учебного курса, читаемого автором студентам на протяжении более чем 20 лет. За этот период дисциплина «Элементы систем автоматики» превратилась в конкретную область знаний, которая оперирует своими логическими понятиями, терминами и определениями. Базируясь на сравнительно простом математическом аппарате двоичной арифметики, булевой алгебры и теории автоматов, она позволяет решать задачи синтеза сложных высокоинтеллектуальных технических устройств формальными методами, опираясь на множество современных интегральных схем и законченных модулей, созданных ведущими мировыми производителями.

В учебном пособии рассмотрены ключевые понятия и модели, практические схемы, микросхемы и законченные изделия отечественных и зарубежных производителей, решающие самые различные задачи в области автоматизации. Отказ от глубокой математической и физической трактовки процессов объясняется стремлением максимально приблизить студентов к реалиям современной действительности, показать им возможности современной элементной базы и пути ее развития.

Данное учебное пособие не является справочником. Приведенные в нем в качестве примеров элементы скорее можно отнести к стандартным. Они не могут даже частично отразить существующего множества средств автоматики, созданных известными гигантами бизнеса: Siemens, Omron, Mitsubishi, LG, Schneider Electric,

Motorola, Texas Instruments, Analog Device и др. К сожалению, отечественные производители в данном перечне пока отсутствуют. Это обстоятельство существенно усложняет изложение материала. Множество заимствованных англоязычных терминов стало привычным в этой области знаний, где терминология разрабатывается и уточняется постоянно мировым сообществом производителей.

В гл. 1 кратко рассмотрены информационные и математические основы автоматики: элементы теории информации, двоичная арифметика и алгебра логики, системы счисления, способы кодирования информации и основные понятия теории автоматов.

В гл. 2 рассмотрены современные цифровые интегральные схемы, составляющие элементную базу автоматики, и основные технологии их производства.

В гл. 3 рассмотрены цифроаналоговые и аналого-цифровые преобразования сигналов, принципы построения преобразователей и их основные схемотехнические особенности.

В гл. 4 кратко представлены микроконтроллеры как ядро системы автоматики, решающие все основные математические, логические задачи и задачи формирования управляющих воздействий.

Гл. 5 содержит систематизированные сведения о датчиках электрических величин и параметров движения электрического привода.

В гл. 6 дано краткое описание принципов построения и схем человекомашинного интерфейса: клавиатур, обеспечивающих ввод информации в систему, и средств индикации, необходимых для представления цифровой информации человеку.

ВВЕДЕНИЕ

Под *элементами систем автоматики* обычно подразумевается множество технических средств, включающее в себя управляющие вычислительные устройства (контроллеры), датчики, исполнительные устройства и устройства, обеспечивающие взаимодействие человека с системой управления.

Основным элементом современной системы автоматики считается *контроллер* — устройство, ориентированное на реализацию алгоритмов цифрового управления различными объектами и процессами. Сбор информации об изменениях технологических параметров объекта управления (температура, давление, перемещение, скорость движения, ток, напряжение и т. д.) обеспечивают в системе различные датчики. Они преобразуют физические воздействия в электрические сигналы и одновременно производят первичную обработку этих сигналов. Современные датчики способны масштабировать сигналы, передавать их и обеспечивают передачу информации другим элементам системы по выделенным линиям связи.

Различные исполнительные устройства в системе преобразуют команды контроллера в необходимые для функционирования системы физические воздействия. Такими устройствами, в первую очередь, являются электроприводы, приводящие объекты различной физической природы в движение. К исполнительным устройствам также относятся силовые преобразователи электрической энергии, электронные ключи, коммутаторы, осуществляющие подключение и плавное регулирование потребления электрической энергии.

Элементами, обеспечивающими взаимодействие системы с человеком, являются различные устройства индикации, представляющие информацию в доступном человеку виде, и устройства управления, позволяющие человеку вмешиваться в процесс управления. Индикаторы могут быть простейшими или сложными (цифровыми, алфавитными или графическими дисплеями) светодиодными схемами. В качестве элементов управления используются простейшие

кнопки, клавиатуры или манипуляторы самых различных конструкций.

Для связи с системами промышленной автоматизации более высокого уровня в системах локальной автоматики всегда предусматриваются интерфейсы локальных сетей. Они обеспечивают в процессе работы прием и передачу по выделенным каналам связи цифровой информации, содержащей в себе исходные данные и команды управления.

Построение систем автоматики основывается на методах теории управления и теории автоматов. Все преобразование информации в системе осуществляется по законам теории информации и алгебры логики.

ЛОГИЧЕСКИЕ ОСНОВЫ АВТОМАТИКИ

1.1. Информация и общие принципы ее преобразования

Для современной автоматики понятие «информация» является важнейшим. Оно объединяет систематизированные сведения об объектах системы автоматического управления, сведения об определенных свойствах или параметрах этих объектов и о зависимостях, описывающих взаимодействие элементов системы. Именно процессы обработки, приема, передачи и хранения информации лежат в основе функционирования абсолютно всех автоматических устройств, хотя физические параметры, контролируемые и регулируемые системами автоматики, могут быть самыми различными (например, скорость, перемещение, ускорение, момент, температура, давление др.).

Физические процессы в окружающем нас мире в абсолютном своем большинстве являются процессами непрерывными. Информация об этих процессах обычно передается в систему автоматики с помощью электрических сигналов — тока или напряжения, связанных с физическими параметрами априорно известными соотношениями. Электрический сигнал, как и любой физический процесс, по своей природе тоже является непрерывным (аналоговым). Сигнал описывается соответствующими графиками или аналитическими выражениями, показывающими его изменения во времени. Но при этом всегда имеется в виду, что причиной изменения величины, отображаемой на графике, является не само время, а некоторый физический процесс.

При сборе информации в системе осуществляются преобразования аналоговых сигналов в цифровую форму. При этом преобразовании сигналы подвергают дискретизации, квантованию и кодированию.

Дискретизация. *Дискретизацией* называется выборка мгновенных значений (отсчетов) непрерывного сигнала с равными интервалами времени Δt между ними. Полученный в результате такого преобразования дискретный сигнал описывается решетчатой функцией (последовательностью отсчетов)

$$x(t) = x(i\Delta t) \equiv x(i),$$

где Δt — интервал (шаг) дискретизации; i — независимая переменная, принимающая только целые положительные значения ($i = 0, 1, 2 \dots$) (рис. 1.1).

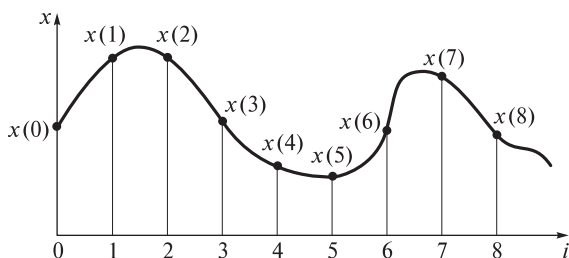


Рис. 1.1. Процесс дискретизации непрерывного сигнала

В результате дискретизации во времени на всем интервале наблюдения непрерывный сигнал $x(t)$ заменяется последовательностью отдельных отсчетов $x(i)$: $x(0)$, $x(1)$, $x(2)$...

Величина $f = 1/\Delta t$, обратная шагу дискретизации, называется *частотой дискретизации*. Правильный выбор частоты дискретизации устанавливает теорема Котельникова, в соответствии с которой для полного восстановления непрерывного сигнала по его дискретным отсчетам необходимо, чтобы частота дискретизации f , как минимум, вдвое превышала наибольшую частоту спектра исходного аналогового сигнала.

Квантование. При квантовании действительное значение сигнала, подвергнувшегося предварительной дискретизации, заменяется ближайшим значением из определенного заранее множества $X = \{X_0, X_1, X_2 \dots X_{k-1}\}$. Функция приобретает ступенчатый вид (рис. 1.2).

В результате квантования сигнал описывается квантованной решетчатой функцией (квантованной последовательностью) $x(i)$, которая может принимать только фиксированные дискретные значения, называемые уровнями квантования: $X_0, X_1, X_2 \dots X_{k-1}$. Количество уровней квантования k определено заранее и в процессе преобразования не изменяется.

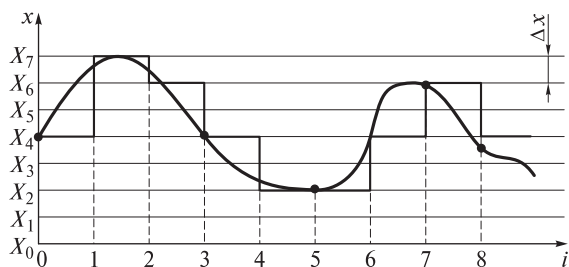


Рис. 1.2. Процесс квантования сигнала

Шагом квантования сигнала Δx называется разность между соседними уровнями квантования. Обычно шаг квантования выбирается постоянным исходя из максимального значения сигнала $x_{\max}$ и числа уровней: $\Delta x \geq x_{\max}/(k - 1)$.

Процесс квантования всегда вносит ошибку в работу системы. Считается, что абсолютная погрешность квантования в процессе преобразования не превышает половины шага квантования.

Кодирование. При кодировании каждому значению квантованного сигнала ставится в соответствие некоторое число (код) из определенного заранее конечного множества чисел. Размер множества определяется числом уровней квантования. Процесс кодирования обычно описывается таблицей кодировки, содержащей два поля. В одном поле таблицы размещается множество значений квантованного сигнала, в другом — соответствующие коды значений. При кодировании используются различные системы счисления и формы представления чисел. Например, при числе уровней квантования $k = 8$ и использовании для кодировки обычной десятичной системы счисления таблица кодировки выглядит следующим образом (табл. 1.1).

При выбранной кодировке цифровой сигнал, соответствующий сигналу $x(t)$ (см. рис. 1.1), после дискретизации, квантования и кодирования описывается последовательностью десятичных чисел $x(i) = 4, 7, 7, 6, 4, 2, 2, 4, 6, 4$.

После кодирования передача или обработка цифрового сигнала сводится к операциям над безразмерными числами (кодами). На практике дискретизация, квантование и кодирование электрических аналоговых сигналов осуществляются при помощи специализированных схем аналого-цифровых преобразователей.

Мера информации. Для оценки количества информации в цифровом сигнале используется так называемая аддитивная мера (мера Хартли), в которой количество информации связывается логарифмической функцией с числом уровней квантования: $I = \log_2 k$. В общем случае для расчета количества информации в цифровом сигнале вводятся два параметра:

- q — общее количество символов в кодировке сигнала;
- n — количество символов, достаточное для представления любого значения закодированного сигнала (разрядность кода).

При заданном количестве символов в кодировке q и разрядности кода n число возможных уровней квантования можно рассчитать по формуле $k = q^n$. Тогда

Таблица 1.1

Пример таблицы кодировки

Уровень сигнала	Десятичный код
X_0	0
X_1	1
X_2	2
...	...
X_7	7

количество информации в одном отсчете сигнала $I = \log_2 k = n \log_2 q$. Полученный результат округляется до ближайшего большего целого числа.

Наиболее часто в цифровой технике используется двоичная кодировка, в которой любой сигнал можно описать, используя всего два символа: 0 и 1. Тогда $q = 2$, а количество информации в одном отсчете равно разрядности кода: $I = n \log_2 2 = n$. Поэтому для измерения количества информации принято использовать специальную единицу, эквивалентную одному символу двоичной кодировки, — бит (bit — binary digit).

В привычной для нас десятичной кодировке сигнал кодируется десятичными цифрами. Количество символов (цифр), используемых для представления числа q , равно 10. В результате $I = n \log_2 10 \approx 3,32n$. Полученное значение обязательно округляется до ближайшего большего целого числа. Например, двухразрядный десятичный код несет в себе 7 бит информации, трехразрядный — 10 бит, четырехразрядный — 14 бит.

При одновременной обработке нескольких отсчетов сигнала или при использовании нескольких источников информации об одном сигнале количество информации суммируется:

$$I = \sum I_j,$$

где I_j — количество информации, полученной от j -го источника.

1.2. Способы кодирования информации

1.2.1. Системы счисления

В процессе переработки цифровой информации элементы автоматики оперируют многоразрядными кодами, рассматриваемыми как числа. Каждый символ кода однозначно сопоставляется с его количественным эквивалентом, называемым цифрой.

Совокупность приемов и правил, установленных для записи чисел с помощью цифр, устанавливается системой счисления. Запись числа в некоторой системе счисления и считается кодом числа.

Для кодирования цифровой информации используются только позиционные системы счисления, в которых значение каждой цифры зависит не только от ее количественного эквивалента, но и от места (позиции) в числе, т.е. в позиционной системе счисления одна и та же цифра может иметь различные значения. Например, в десятичном коде 2255 первая слева цифра 2 обозначает две тысячи, а вторая — две сотни; первая цифра 5 обозначает пять десятков, вторая — пять единиц.

Любая позиционная система счисления характеризуется основанием. Основание q позиционной системы счисления — это количество знаков или символов, используемых в данной системе для изображения числа. В десятичной системе счисления используется десять цифр от 0 до 9, основание $q = 10$; в двоичной системе счисления — две цифры: 0 и 1, основание $q = 2$. Теоретически возможно бесчисленное множество позиционных систем, так как за основание можно принять любое целое число. На практике используются только системы счисления с основаниями 2, 8, 10 и 16.

При записи некоторого числа в позиционной системе счисления соответствующие цифры числа размещаются по отдельным нужным позициям, которые принято называть разрядами числа. Количество разрядов n в записи числа называется его *разрядностью*.

В общем случае в позиционной системе счисления с основанием q число A представляется в виде полинома:

$$A = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1q^1 + a_0q^0, \quad (1.1)$$

где n — количество цифр в числе; a_i — коэффициенты, в качестве которых могут быть использованы любые цифры данной системы счисления.

Числа в позиционной системе счисления представляются в виде последовательности соответствующих цифр: $A = a_{n-1} \dots a_2 a_1 a_0$. При такой записи вес каждого разряда при движении справа налево увеличивается в q раз по сравнению с соседним. Нулевым считается младший разряд.

Десятичная система счисления. Десятичная система счисления наиболее привычна для человека и используется в системах автоматики в основном при вводе и выводе информации. Запись числа в десятичной системе счисления является его десятичным кодом, обозначаемым как d -код (decimal code).

Основание десятичной системы $q = 10$, и в записи вида (1.1) используется десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9. Поэтому полином (1.1) представляется в виде

$$A = a_{n-1}10^{n-1} + \dots + a_210^2 + a_110^1 + a_010^0.$$

Десятичный код числа, в отличие от остальных кодов, обычно не обозначают никакими символами. Количество информации, бит, содержащейся в n -разрядном десятичном числе: $I = n \log_2 10 \approx 3,32n$.

Двоичная система счисления. Двоичная система счисления считается основной в автоматике и вычислительной технике. Запись числа в двоичной системе счисления считается двоичным кодом числа, обозначаемым как b -код (binary code).

Основанием двоичной системы счисления является число 2, и в записи вида (1.1) используется только две цифры: 0 и 1. Поэтому полином (1.1) выглядит следующим образом:

$$A = a_{n-1}2^{n-1} + \dots + a_22^2 + a_12^1 + a_02^0. \quad (1.2)$$

Если в одном документе одновременно используются разные системы счисления, то двоичные числа обычно обозначают символом b , записываемым слева от числа. Например, при вычислениях по формуле (1.2)

$$b11010 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 26.$$

Количество информации, содержащейся в n -разрядном двоичном числе, равно количеству разрядов этого числа, бит: $I = n \log_2 2 = n$. Поэтому отдельные разряды двоичных чисел тоже принято называть битами.

Для перевода целого десятичного числа в двоичный код используется деление на основание системы счисления, т. е. на число 2. На каждом этапе деления последовательно, начиная с младшего разряда двоичного кода, фиксируется остаток. Процесс деления заканчивается, если результат деления меньше 2.

Например, перевод числа 53 в двоичную систему счисления выполняется следующим образом:

$$\begin{array}{ccccccccc} 53/2 = 26/2 = 13/2 = 6/2 = 3/2 = 1 & & & & & & & & \\ \text{(младший бит)} & 1 & 0 & 1 & 0 & 1 & 1 & & \text{(старший бит)} \end{array}$$

В результате: $53 = b110101$.

Восьмеричная система счисления. Запись числа в восьмеричной системе счисления считается его восьмеричным кодом. Код принято называть q -кодом, поскольку буква «о», фигурирующая в английской аббревиатуре восьмеричного кода (octal code), ассоциируется с цифрой 0 и неудобна для записи.

В восьмеричной системе счисления используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Основание системы счисления равно 8, и запись вида (1.1) соответствует полиному

$$A = a_{n-1}8^{n-1} + \dots + a_28^2 + a_18^1 + a_08^0.$$

Восьмеричные коды принято отмечать слева символом q . Например: $q4712 = 4 \times 8^3 + 7 \times 8^2 + 1 \times 8 + 2 \times 8^0 = 2506$.

Количество информации, бит, содержащейся в n -разрядном восьмеричном числе: $I = n \log_2 8 = 3n$.

Восьмеричная система счисления используется как удобная короткая форма представления двоичных чисел, поскольку каждой восьмеричной цифре можно поставить в соответствие трехразрядный двоичный эквивалент (табл. 1.2).

Для перехода от двоичного кодирования к восьмеричному двоичный код числа начиная с младшего разряда разбивают на группы по три символа. Если в последней (левой) группе остается менее трех символов, то ее снаружи дополняют нулями. Затем каж-

Соответствие восьмеричных и двоичных кодировок

<i>q</i> -код	0	1	2	3	4	5	6
<i>b</i> -код	000	001	010	011	100	101	110

дую группу заменяют соответствующим восьмеричным символом в соответствии с табл. 1.1. Например: $b001\ 101\ 111\ 110 = q1576$.

Перевод десятичных чисел в восьмеричные осуществляется многократным делением десятичного числа на основание системы счисления 8. На каждом этапе деления фиксируется остаток. Процесс заканчивается, когда результат деления становится меньше 8. Например, перевод числа 129 выглядит следующим образом:

$$\begin{array}{rcll} 129/8 = & 16/8 = & 2 & \\ \text{(младший разряд)} & 1 & 0 & 2 \text{ (старший разряд)} \end{array}$$

В результате: $128 = q201$.

Шестнадцатеричная система счисления. Запись числа в шестнадцатеричной системе счисления считается его шестнадцатеричным кодом и обозначается как *h*-код (hexadecimal code).

В шестнадцатеричной системе счисления для кодирования используется 16 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *A*, *B*, *C*, *D*, *E*, *F*. Запись шестнадцатеричного числа представляется в виде полинома с основанием 16:

$$A = a_{n-1}16^{n-1} + \dots + a_216^2 + a_116^1 + a_016^0.$$

Шестнадцатеричные числа обычно обозначаются слева символом *h*. Например: $hB2E = B \times 16^2 + 2 \times 16^1 + E \times 16^0 = 2862$.

Количество информации, бит, содержащейся в *n*-разрядном шестнадцатеричном числе: $I = n \log_2 16 = 4n$.

Шестнадцатеричная система счисления, как и восьмеричная, используется как удобная форма представления двоичной информации. При этом каждой группе из четырех двоичных символов

Соответствие шестнадцатеричных и двоичных кодировок

<i>h</i> -код	<i>b</i> -код	<i>h</i> -код	<i>b</i> -код	<i>h</i> -код	<i>b</i> -код	<i>h</i> -код
0	0000	4	0100	8	1000	<i>C</i>
1	0001	5	0101	9	1001	<i>D</i>
2	0010	6	0110	<i>A</i>	1010	<i>E</i>
3	0011	7	0111	<i>B</i>	1011	<i>F</i>

ставится в соответствие один символ шестнадцатеричный (табл. 1.3).

Переход от двоичной кодировки к шестнадцатеричной подразумевает разбиение шестнадцатеричного числа начиная с младшего разряда, на группы из четырех символов. Например: $b1111\ 1000\ 1011\ 1110 = h7C5F$.

При переводе десятичного кода в шестнадцатеричный десятичное число многократно делится на основание системы счисления 16 и фиксируются остатки. Например, перевод числа 2005 можно представить в следующем виде:

	$2005/16 =$	$125/16 =$	7
(младший разряд)	5	D	7(старший разряд)

В результате: $2005 = h7D5$.

1.2.2. Кодирование числовой информации

Разрядность числовых данных. Данные, представляющие числовую информацию, как правило, имеют фиксированную длину (разрядность). В автоматике, как и в вычислительной технике, допускается использовать коды с разрядностью 4, 8, 16, 32 или 64 бит. Для обозначения соответствующих форматов используются общепринятые термины:

- 4 бит — полубайт или тетрада;
- 8 бит — байт;
- 16 бит — полуслово;
- 32 бит — слово;
- 64 бит — двойное слово;

Форматы числовых данных. При двоичном кодировании чисел в определенном месте разрядной сетки всегда предполагается наличие запятой и фиксируется положение знакового разряда числа.

Для кодирования запятой в числовой информации никогда не предусматриваются специальные символы. Просто подразумевается, что запятая находится или перед старшим разрядом кода, или после младшего разряда кода. В первом случае речь может идти только о дробных числах, которые по модулю меньше 1, во втором — только о целых числах.

Для кодирования знака обычно предусматривается специальный знаковый разряд. В этом разряде знак «+» кодируется символом 0, а знак «-» кодируется символом 1.

На рис. 1.3 показана 8-разрядная сетка, где запятая зафиксирована после младшего разряда. Веса отдельных разрядов в этом случае меняются от 2^0 до 2^6 .

В 8-разрядной сетке могут быть представлены целые числа от $b01111111 = +2^7 - 1 = 127$ до $b11111111 = -(2^7 - 1) = -127$.

7	6	5	4	3	2	1	0
Знак	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Рис. 1.3. Представление целых чисел с фиксированной запятой

На рис. 1.4 изображена 8-разрядная сетка, где запятая фиксируется перед старшим разрядом. Веса разрядов в этом случае меняются от 2^{-7} до 2^{-1} .

В такой разрядной сетке можно представить числа в диапазоне от $+(1 - 2^{-7}) = 0,9921875$ до $-(1 - 2^{-7}) = -0,9921875$.

Числа со знаком. Для представления двоичных чисел со знаком можно использовать три вида кодов: прямой, обратный и дополнительный. В любом случае знак числа принято записывать в его старшем разряде.

Прямой код n -разрядного двоичного числа со знаком имеет вид

$$A = a_{n-1}a_{n-2}a_{n-1} \dots a_1a_0,$$

где старший (знаковый) разряд числа $a_{n-1} = \begin{cases} 0 & \text{при } A \geq 0, \\ 1 & \text{при } A < 0. \end{cases}$

При использовании прямого кода сложение чисел, имеющих одинаковые знаки, выполняется по обычным правилам двоичной арифметики: числа складываются, сумме присваивается код знака слагаемых. Однако операция алгебраического сложения чисел с разными знаками существенно затрудняется. В этом случае необходимо определить большее по модулю число, произвести вычитание чисел и присвоить разности знак большего по модулю числа. Поэтому для представления отрицательных чисел обычно применяют другие коды: обратный и дополнительный.

Чтобы сменить знак числа с положительного на отрицательный и представить результат в обратном коде, необходимо все разряды числа, включая знаковый, изменить на противоположные.

Например, число $+12$ в 8-битном представлении записывается в прямом коде как $b00001100$, а число -12 в обратном коде — как $b11110011$. В обоих случаях старший бит кода является знаковым.

7	6	5	4	3	2	1	0
Знак	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

Рис. 1.4. Представление дробных чисел с фиксированной запятой

Разрядная сетка длиной n бит обеспечивает запись в обратном коде отрицательных целых чисел от $b11...0 = -1$ до $b10...0 = -(2^{n-1} - 1)$.

При 8-разрядной сетке это диапазон от -1 до -127 . Для дробных чисел возможно представление от $b11...10 = -2^{-(n-1)}$ до $b10...00 = -(1 - 2^{n-1})$. При 8-разрядной сетке — это числа от $-1/128 = 0,0078125$ до $-127/128 = 0,9921875$.

Обратный код имеет две различные записи нуля: как положительное и как отрицательное число. Например, в 8-разрядном представлении $+0 = b00000000$, а $-0 = b11111111$.

Изменение знака отрицательного числа, представленного в обратном коде, соответствует инвертированию его кода. В результате получается прямой код соответствующего положительного числа.

Дополнительный код больше обратного на единицу младшего разряда, поэтому для представления числа в дополнительном коде его можно сначала перевести в код обратный, а затем прибавить 1.

Таблица 1.4

Примеры записи целых 8-разрядных чисел

Десятичное число	Прямой код	Обратный код	Дополнит код
+127	<i>b01111111</i>		
+126	<i>b01111110</i>		
+7	<i>b00000111</i>		
+3	<i>b00000011</i>		
+2	<i>b00000010</i>		
+1	<i>b00000001</i>		
+0	<i>b00000000</i>		
-128			<i>b1000</i>
-127	<i>b11111111</i>	<i>b1000 0000</i>	<i>b1000</i>
-126	<i>b11111110</i>	<i>b1000 0001</i>	<i>b1000</i>
-7	<i>b10000111</i>	<i>b1111 1000</i>	<i>b1111</i>
-3	<i>b10000011</i>	<i>b1111 1100</i>	<i>b1111</i>
-2	<i>b10000010</i>	<i>b1111 1101</i>	<i>b1111</i>
-1	<i>b10000001</i>	<i>b1111 1110</i>	<i>b1111</i>
-0	<i>b10000000</i>	<i>b1111 1111</i>	<i>b0000</i>

В области целых чисел дополнительный код позволяет представить числа от $b100...00 = -2^{n-1}$ до $b11...11 = -1$. 8-разрядный (байтовый) формат позволяет записать числа от 0 до -128 .

В области дробных чисел можно представить числа от $b10...00 = -1$ до $b11...11 = 2^{-(n-1)}$. Байтовый формат позволяет представить дробные числа от -1 до $-1/128 = -0,0078125$.

При представлении положительных чисел прямым кодом, а отрицательных — дополнительным кодом положительный и отрицательный 0 изображаются совершенно одинаково: $-0 = +0 = b00...00$.

В табл. 1.4 приведены примеры записи целых 8-разрядных положительных и отрицательных чисел в диапазоне от $+127$ до -128 в прямом, обратном и дополнительном кодах.

1.2.3. Специальные кодировки

Двоично-десятичный код. Двоично-десятичное кодирование, или *bcd*-код (binary code decimal), используется для записи десятичных чисел с помощью двоичных символов. Он используется в устройствах ввода и вывода данных, где информация представляется человеку.

В *bcd*-коде десятичного числа каждая десятичная цифра изображается четырьмя двоичными разрядами (тетрадой), а коды цифр совпадают с кодами цифр обычного *b*-кода. Из 16 кодовых комбинаций *b*-кода в *bcd*-коде можно использовать только 10, а 6 остальных комбинаций запрещены. В частности, запрещены все 4-битные комбинации (1010, 1011, 1100, 1101, 1110, 1111), соответствующие в двоичном коде числам более 9. Поскольку для представления двоично-десятичных чисел в *bcd*-коде в каждой тетраде используется только 10 комбинаций двоичных символов из 16 возможных, то двоично-десятичные числа получаются длиннее двоичных.

Таблица 1.5

Примеры записи десятичных чисел в *bcd*-коде

<i>d</i> -код	<i>bcd</i> -код	<i>d</i> -код	<i>bcd</i> -к
10	0001 0000	20	0010 (
11	0001 0001	21	0010 (
12	0001 0010	22	0010 (
...		...	
19	0001 1001	89	1000 1

Таблица кодировки кода Грея

<i>d</i> -код	<i>g</i> -код	<i>d</i> -код	<i>g</i> -код	<i>d</i> -код	<i>g</i> -код	<i>d</i> -код
0	0000	4	0110	8	1100	12
1	0001	5	0111	9	1101	13
2	0011	6	0101	10	1111	14
3	0010	7	0100	11	1110	15

В табл. 1.5 приведены примеры записи некоторых десятичных чисел в *bcd*-коде.

Код Грея. Код Грея, или *g*-код (Gray code), используется в различных датчиках перемещения. Соседние числа в таблице кодировки кода Грея отличаются друг от друга только одним битом. Это позволяет избавиться от ошибок при переходе от одной кодовой комбинации к другой.

Для представления любого числа в коде Грея необходимо столько же бит, сколько и для обычного *b*-кода. В табл. 1.6 в коде Грея представлены десятичные числа от 0 до 15.

Код «1 из *n*». Код «1 из *n*» используется в различных дешифраторах и элементах индикации. Для представления каждого числа в коде «1 из *n*» выделяется *n* бит, но единичное значение может принимать только один из них. Код является избыточным, из 2^n возможных кодовых комбинаций в нем используется только *n*. Например, при 8-битном кодировании (табл. 1.7) из 256 кодовых комбинаций 248 не используются. Это позволяет легко выявлять ошибки в представлении данных и обеспечивает высокую помехозащищенность кода.

Семисегментный код. Семисегментный код разработан специально для схем управления цифровыми индикаторами. Такой индикатор содержит семь светодиодов, расположенных в корпусе в определенном порядке. При этом семь сегментов индикатора, обо-

Таблица 1.7

Таблица кодировки кода «1 из *n*» при *n* = 8

<i>d</i> -код	Код «1 из <i>n</i> »	<i>d</i> -код	Код «1
0	00000001	4	00010
1	00000010	5	00100
2	00000100	6	01000
3	00001000	7	10000

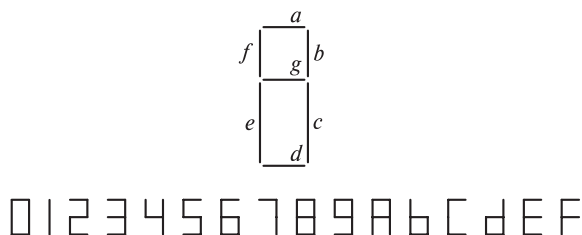


Рис. 1.5. Семисегментный индикатор

значаемых буквами a, b, c, d, e, f и g , способны отобразить все шестнадцатеричные цифры от 0 до F (рис. 1.5).

Семисегментный код (табл. 1.8) 7-разрядный. Каждый бит кода соответствует одному сегменту индикатора (см. рис. 1.3). Код является избыточным, из $2^7 = 128$ возможных кодовых комбинаций в нем используется только 16. Никакие вычисления в таком коде никогда не производятся.

Кодирование текста и команд. При обмене данными между элементами систем автоматики возникает необходимость в передаче не только числовой информации, но и различных знаков и буквенных символов. Эти символы также кодируются с помощью двоичных кодов, причем различных кодировок существует довольно много.

Наиболее распространенным в автоматике считается введенный в 1963 г. в США американский стандартный код обмена информацией ASCII (American Standard Code for Information Interchange).

В коде ASCII для кодирования каждого символа отводится 7 бит. В результате он позволяет представить $2^7 = 128$ различных символов (табл. 1.9). Шестнадцатеричный код каждого символа в табл. 1.9 складывается из старшей цифры, соответствующей столбцу, и младшей цифры, соответствующей строке, в которых размещен символ. Например, символ G , расположенный на пересечении четвертого столбца и седьмой строки, имеет код $h47$.

Таблица 1.8

Таблица кодировки семисегментного кода

h -код	s -код	h -код	s -код	h -код	s -код	h -код	
0	1111110	4	0110011	8	1111111	C	1
1	0110000	5	1011011	9	1111101	D	0
2	1101101	6	1011111	A	1110111	E	1
3	1111001	7	1111000	B	0011111	F	1

Таблица кодировки кода ASCII

Млад- шая цифра	Старшая цифра						
	0	1	2	3	4	5	6
0	<i>NUL</i>	<i>DLE</i>	<i>SP</i>	0	@	<i>P</i>	`
1	<i>SOH</i>	<i>DC1</i>	!	1	<i>A</i>	<i>Q</i>	<i>a</i>
2	<i>STX</i>	<i>DC2</i>	“	2	<i>B</i>	<i>R</i>	<i>b</i>
3	<i>ETX</i>	<i>DC3</i>	#	3	<i>C</i>	<i>S</i>	<i>c</i>
4	<i>EOT</i>	<i>DC4</i>	\$	4	<i>D</i>	<i>T</i>	<i>d</i>
5	<i>ENQ</i>	<i>NAK</i>	%	5	<i>E</i>	<i>U</i>	<i>e</i>
6	<i>ACK</i>	<i>SYN</i>	&	6	<i>F</i>	<i>W</i>	<i>g</i>
7	<i>BEL</i>	<i>ETB</i>	‘	7	<i>G</i>	<i>W</i>	<i>g</i>
8	<i>BS</i>	<i>CAN</i>	(	8	<i>H</i>	<i>X</i>	<i>h</i>
9	<i>HT</i>	<i>EM</i>	)	9	<i>I</i>	<i>Y</i>	<i>i</i>
<i>A</i>	<i>LF</i>	<i>SUB</i>	“	:	<i>J</i>	<i>Z</i>	<i>j</i>
<i>B</i>	<i>VT</i>	<i>ESC</i>	+	;	<i>K</i>	[	<i>k</i>
<i>C</i>	<i>FF</i>	<i>FS</i>	,	<	<i>L</i>	\	<i>l</i>
<i>D</i>	<i>CR</i>	<i>GS</i>	-	=	<i>M</i>	]	<i>m</i>
<i>E</i>	<i>SO</i>	<i>RS</i>	.	>	<i>N</i>	^	<i>n</i>
<i>F</i>	<i>SI</i>	<i>US</i>	/	?	<i>O</i>	_	<i>o</i>

Код ASCII содержит две группы символов: символы пишущей машинки и управляющие символы.

Символами пишущей машинки (коды *h20...h7E*) считаются прописные и строчные латинские буквы, цифры и специальные знаки, к которым относятся: минус (-), подчеркивание (_), апостроф (‘), тупое (обратное) ударение (’), кавычки (“”), амперсant (&), номер (#), стрелка вверх (^), вертикальная черта (|), надчеркивание (тильда) (~), коммерческий «эт» (@) и пробел (Space) (SP).

Управляющие символы, размещенные в основном в двух первых столбцах таблицы (коды *h00...h1F* и *h7F*), используются в различных системах связи для передачи команд.

В каждом конкретном устройстве могут использоваться не все управляющие символы, а задействованные символы могут интерпретироваться по усмотрению разработчика.