ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 16022— 2008

Автоматическая идентификация

КОДИРОВАНИЕ ШТРИХОВОЕ

Спецификация символики Data Matrix

ISO/IEC 16022:2006
Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification
(IDT)

Издание официальное

БЗ 9—2008/288

Москва
Стандартинформ
2009

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Ассоциацией автоматической идентификации «ЮНИСКАН/ГС1 РУС» совместно с Обществом с ограниченной ответственностью (ООО) НПЦ «Интелком» на основе аутентичного перевода стандарта, указанного в пункте 4, выполненного ООО НПЦ «Интелком»

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 355 «Автоматическая идентификация»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 18 декабря 2008 г. № 509-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 16022:2006 «Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация символики штрихового кода Data Matrix» (ISO/IEC 16022:2006 «Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification»), за исключением приложения U, содержащего сведения о соответствии терминов на русском и английском языках, приложения V, включающего в себя сведения о наборах знаков по ИСО/МЭК 646, ИСО/МЭК 8859-1 и ИСО/МЭК 8859-5. В приложении M приведены исправления в соответствии со списком технических опечаток 1 (Technical Corrigendum 1) к ISO/IEC 16022.1:2006.

Наименование национального стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5—2004 (подраздел 3.5) и учета его принадлежности к группе стандартов «Автоматическая идентификация».

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных (региональных) стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении W

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомления и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2009

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения, символы и математические/логические обозначения . . . 2
3.1 Термины и определения . . . 2
3.2 Символы . . . 2
3.3 Математические обозначения . . . 3
4 Описание символов . . . 3
4.1 Основные параметры . . . 3
4.2 Дополнительные свойства . . . 4
4.3 Структура символов . . . 4
5 Требования к символам версии ECC 200 . . . 5
5.1 Основные положения процедуры кодирования . . . 5
5.2 Кодирование данных . . . 6
5.3 Рекомендации пользователям . . . 12
5.4 Интерпретация в расширенном канале . . . 13
5.5 Атрибуты символа версии ECC 200 . . . 14
5.6 Структурированное соединение . . . 15
5.7 Обнаружение и исправление ошибок . . . 16
5.8 Формирование символа . . . 18
6 Требования к символам версии ECC 000-140 . . . 19
6.1 Рекомендации по применению . . . 19
6.2 Порядок кодирования . . . 19
6.3 Кодирование данных . . . 19
6.4 Выбор пользователем уровня исправления ошибок . . . 22
6.5 Формирование незащищенного двоичного потока . . . 22
6.6 Построение нерандомизированного двоичного потока . . . 23
6.7 Шаблонная рандомизация . . . 24
6.8 Размещение модулей в матрице . . . 24
7 Размеры символов . . . 24
7.1 Размеры . . . 24
8 Качество печати символов . . . 24
8.1 Параметры качества символа . . . 24
8.2 Измерения в процессе контроля . . . 25
9 Рекомендуемый алгоритм декодирования для символики Data Matrix . . . 25
10 Рекомендации для пользователя . . . 34
11 Передаваемые данные . . . 35
Приложение A (обязательное) Процесс чередования в символике версии ECC 200 . . . 37
Приложение B (обязательное) Шаблонная рандомизация в символике версии ECC 200 . . . 39
Приложение C (обязательное) Наборы кодируемых знаков символики версии ECC 200 . . . 41
Приложение D (обязательное) Направляющие шаблоны символов версии ECC 200 . . . 45
Приложение E (обязательное) Алгоритм обнаружения и исправления ошибок Рида-Соломона для символики версии ECC 200 . . . 47
Приложение F (обязательное) Размещение знаков символа в символе версии ECC 200 . . . 51
Приложение G (обязательное) Параметры символов версии ECC 000-140 . . . 62
Приложение H (обязательное) Сетки размещения модулей данных для символов версии ECC 000-140 67
Приложение I (обязательное) Схемы кодирования знаков символики версии ECC 000-140 . . . 74
Приложение J (обязательное) Алгоритм расчета CRC для ECC 000-140 . . . 82
Приложение K (обязательное) Алгоритмы обнаружения и исправления ошибок для символов версии ECC 000-140 . . . 83
Приложение L (обязательное) Образец рандомизации двоичного потока в символах версии ECC 000-140 (шестнадцатеричные значения) . . . 87
Приложение M (обязательное) Качество печати символов Data Matrix. Аспекты, связанные с особенностями символики . . . 88

Приложение N (обязательное) Идентификатор символики 95
Приложение O (справочное) Пример кодирования символа версии ECC 200 96
Приложение P (справочное) Кодирование данных в символе ECC 200 с использованием минимального числа знаков символа 98
Приложение Q (справочное) Пример кодирования данных в символах версии ECC 000-140 с использованием кода исправления ошибок уровня ECC 050 101
Приложение R (справочное) Рекомендации по методам контроля процесса формирования символов . 107
Приложение S (справочное) Возможность автоматического распознавания 109
Приложение T (справочное) Системный подход 110
Приложение U (справочное) Соответствие терминов на русском и английском языках 111
Приложение V (справочное) Набор знаков ASCII (версия КОИ-7) по ИСО/МЭК 646, графические знаки расширенного набора знаков ASCII (версии КОИ-8) по ИСО/МЭК 8859-1 и набор 8-битовых графических знаков (версия КОИ-8) по ИСО/МЭК 8859-5 112
Приложение W (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам 124
Библиография 125

Введение

Data Matrix — двухмерная матричная символика, состоящая из квадратных модулей, упорядоченных внутри периметра шаблона поиска. В настоящем документе представление символа и его описание приведено, главным образом, для темных модулей на светлом фоне. Тем не менее, символы Data Matrix также могут быть напечатаны в виде светлых модулей на темном фоне.

Производителям оборудования и пользователям технологии штрихового кодирования необходима общедоступная стандартная спецификация символики, на которую они могли бы ссылаться при разработке оборудования и стандартов по применению. С этой целью и был разработан настоящий стандарт.

Следует обратить внимание на возможность того, что некоторые элементы, включенные в настоящий стандарт, могут быть объектом патентного права, и организации ИСО и МЭК не берут на себя ответственность за определение некоторых или всех подобных патентных прав.

Сноски в тексте стандарта, выделенные курсивом, приведены для пояснения текста стандарта.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Автоматическая идентификация

КОДИРОВАНИЕ ШТРИХОВОЕ

Спецификация символики Data Matrix

Automatic identification. Bar coding. Data Matrix symbology specification

Дата введения — 2010 — 01 — 01

1 Область применения

Настоящий стандарт устанавливает требования к символике Data Matrix[1)], а также параметры символики, кодирование знаков данных, форматы символов, требования к размерам и качеству печати, правила исправления ошибок, алгоритм декодирования и прикладные параметры, выбираемые пользователем.

Настоящий стандарт распространяется на все символы символики Data Matrix, напечатанные или нанесенные каким-либо другим способом.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты и другие нормативные документы, которые необходимо учитывать при использовании настоящего стандарта. В случае ссылок на документы, у которых указана дата утверждения, необходимо пользоваться только указанной редакцией. В случае, когда дата утверждения не приведена, следует пользоваться последней редакцией ссылочных документов, включая любые поправки и изменения к ним:

ИСО/МЭК 15424 Информационные технологии. Технологии автоматической идентификации и сбора данных. Идентификаторы носителей данных (включая идентификаторы символик) (Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)

ИСО/МЭК 19762-1 Информационные технологии. Технологии автоматической идентификации и сбора данных. Гармонизированный словарь. Часть 1. Общие термины, связанные с автоматической идентификацией и сбором данных (Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC)

ИСО/МЭК 19762-2 Информационные технологии. Технологии автоматической идентификации и сбора данных. Гармонизированный словарь. Часть 2. Средства для оптического считывания (Information technology — Automatic identification and data capture (AIDC) techniques - Harmonized vocabulary — Part 2: Optically readable media (ORM))

ИСО/МЭК 15415 Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация испытаний качества печати символов штрихового кода. Двумерные символы (Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Two-dimensional symbols)

ИСО/МЭК 15416 Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация испытаний качества печати символов штрихового кода. Линейные символы

[1)] *Название символики произносится как Дата Матрикс, что в переводе на русский язык — «матрица данных».*

Издание официальное

(Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols)

ИСО/МЭК 646:1991 Информационные технологии. Набор 7-битовых кодированных знаков ИСО для обмена информацией (Information technology — ISO 7-bit coded character set for information interchange)

ИСО/МЭК 8859-1 Информационные технологии. Наборы 8-битовых однобайтных кодированных графических знаков. Часть 1. Латинский алфавит № 1 (Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet № 1)

ИСО/МЭК 8859-5:1999 Информационные технологии. Наборы 8-битовых однобайтных кодированных графических знаков. Часть 5. Латинский/кирилловский алфавит (Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet)

AIM Inc. ITS/04-001 Международный технический стандарт. Интерпретации в расширенном канале. Часть 1. Схемы идентификации и протокол (AIM Inc. ITS/04-001 International Technical Standard: Extended Channel Interpretations — Part 1: Identification Schemes and Protocol)

3 Термины, определения, символы и математические/ логические обозначения

3.1 Термины и определения

В данном документе используются термины, определенные в ИСО/МЭК 19762-1, ИСО/МЭК 19762-2, а также следующие:

3.1.1 **кодовое слово** (codeword): Значение знака символа, формируемое на промежуточном уровне кодирования в процессе преобразования исходных данных в их графическое представление в символе.

3.1.2 **модуль** (module): Отдельная ячейка матричной символики, используемая для кодирования одного бита информации и имеющая номинально квадратную форму в символах Data Matrix.

3.1.3 **сверточное кодирование** (convolutional coding): Алгоритм контроля и исправления ошибок, преобразующий множество битов на входе во множество битов на выходе, которое может быть восстановлено после повреждения, путем кодирования с разделением множества входящих битов на блоки с последующим проведением операции свертки каждого входящего блока с регистром сдвига со множеством состояний для получения защищенных на выходе блоков.

Примечание — Такие алгоритмы кодирования могут быть реализованы с помощью аппаратных средств путем использования входных и выходных коммутаторов, регистров сдвига и вентилей исключающих ИЛИ[1)].

3.1.4 **шаблонная рандомизация** (pattern randomising): Процедура, с помощью которой исходный набор битов превращают в другой набор битов путем инвертирования отдельных битов с целью уменьшения вероятности повторения в символе одинаковых наборов.

3.2 Символы

В данном документе, если иное не предусмотрено в особых случаях, применяют следующие математические символы:

d — число кодовых слов исправления ошибок;

e — число стираний;

k — (для версии ECC 000-140) число битов в полном сегменте на входе в конечный автомат для генерирования сверточного кода;
(для версии ECC 200) общее число кодовых слов исправления ошибок;

m — порядок памяти сверточного кода,

n — (для версии ECC 000-140) число битов в полном сегменте, сгенерированных конечным автоматом, порождающим сверточный код;
(для версии ECC 200) общее число кодовых слов данных;

N — числовое основание в схеме кодирования;

p — число кодовых слов, зарезервированных для обнаружения ошибок;

S — знак символа;

t — число ошибок;

u — сегмент битов на входе в конечный автомат, принимающий *k* битов за единицу времени;

v — сегмент битов на выходе из конечного автомата, генерирующего *n* битов за единицу времени;

[1)] *Международное обозначение операции исключающее ИЛИ: exclusive-or — XOR.*

X — горизонтальный и вертикальный размеры модуля;

ε — кодовое слово исправления ошибок.

3.3 Математические обозначения

В настоящем стандарте используются следующие обозначения и математические операции:

div — оператор деления на целое число;

mod — остаток при делении на целое число;

XOR — исключающее ИЛИ (exclusive-or) — логическая функция или операция, результатом которой является единица только в случае неэквивалентности двух входов;

LSB — младший значащий разряд (Least Significant Bit);

MSB — старший значащий разряд (Most Significant Bit).

4 Описание символов

4.1 Основные параметры

Data Matrix представляет собой двумерную матричную символику.

Существуют две версии символики Data Matrix:

- версия, обозначаемая ECC 200, в которой используют алгоритм исправления ошибок Рида-Соломона. Версия ECC 200 рекомендуется для разработки любого нового применения;

- версия, обозначаемая ECC 000-140, с несколькими доступными уровнями сверточного исправления ошибок, такими как ECC 000, ECC 050, ECC 080, ECC 100 и ECC 140. Версию ECC 000-140 следует использовать только для замкнутых прикладных систем, в которых одна и та же сторона контролирует создание и считывание символов и обеспечивает функционирование всей системы.

Символика Data Matrix имеет следующие параметры:

a) кодируемый набор знаков:

1) знаки набора ASCII (версии КОИ-7) по ИСО/МЭК 646[1)] (согласно национальной версии США[2)]) (далее — знаки ASCII (КОИ-7)) с десятичными значениями от 0 до 127.

Примечание 1 — Указанная версия ASCII (КОИ-7) состоит из набора знаков G0 по ИСО/МЭК 646 и C0 по ИСО/МЭК 6429, в котором знаки с десятичными значениями от 28 до 31 соответствуют знакам FS, GS, RS и US соответственно;

2) знаки расширенного набора ASCII (версия КОИ-8) по ИСО/МЭК 8859-1[3)] (далее — знаки расширенного набора ASCII (КОИ-8)) с десятичными значениями от 128 до 255;

b) представление данных: темный модуль соответствует двоичной единице, светлый — двоичному нулю.

Примечание 2 — Настоящий стандарт определяет символы Data Matrix как темные модули, расположенные на светлом фоне. Однако в 4.2 предусмотрено, что символы могут также быть образованы с заменой цвета на противоположный, и для таких символов положения настоящего стандарта в отношении темных модулей должны применяться к светлым модулям и наоборот;

c) размеры символа в модулях (без учета свободной зоны).

- для версии ECC 200 — от 10 × 10 до 144 × 144, только четные значения;

- для версии ECC 000-140 — от 9 × 9 до 49 на 49, только нечетные значения;

d) число знаков данных в символе (для символа максимального размера версии ECC 200):

1) алфавитно-цифровые данные — до 2335 знаков,

2) в 8-битовых байтах — 1555 знаков,

3) числовые данные — 3116 числовых разрядов;

e) задаваемое исправление ошибок:

- для версии ECC 200 — исправление ошибок Рида - Соломона;

[1)] *Набор знаков ASCII (версия КОИ-7) по ИСО/МЭК 646 приведен в приложении V.*

[2)] *Набор знаков по ANSI INCITS 4-1986 (R2007) Information Systems — Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII) (Информационные системы — Кодированные наборы знаков — 7-битовый американский национальный стандартный код для обмена информацией (7-битовый ASCII).*

[3)] *Набор знаков расширенного набора ASCII (КОИ-8) приведен в приложении V.*

- для версии ECC 000-140 — четыре уровня исправления ошибок на основе сверточного кода плюс (по выбору) только обнаружение ошибки;

f) тип кода: матричный;

g) независимость от ориентации: присутствует.

4.2 Дополнительные свойства

Символика Data Matrix обладает следующими дополнительными, неотъемлемыми или устанавливаемыми по выбору свойствами:

a) обратимость изображения (неотъемлемое свойство). Символы предназначены для считывания как напечатанные темным на светлом фоне, так и светлым на темном фоне (рисунок 1). Положения настоящего стандарта установлены для темного изображения на светлом фоне, следовательно указания о темных или светлых модулях должны рассматриваться как указания о светлых или темных модулях соответственно для символов с обращением изображения;

b) интерпретации в расширенном канале (только для версии ECC 200, свойство по выбору). Данный механизм позволяет использовать знаки из иных наборов (например, знаки арабского, кирилловского, греческого, еврейского алфавитов) и иных различных интерпретаций данных или представлять их в соответствии с особыми отраслевыми требованиями;

c) прямоугольная форма символов (только для версии ECC 200, свойство по выбору). Установлены шесть форматов символа прямоугольной формы;

d) структурированное соединение (только для версии ECC 200, свойство по выбору). Позволяет представить один документ в виде нескольких (до 16) символов Data Matrix. Исходные данные могут быть восстановлены вне зависимости от порядка сканирования символов.

4.3 Структура символов

Каждый символ Data Matrix состоит из областей данных, составленных из номинально квадратных модулей, структурированных в регулярную матрицу. В больших символах версии ECC 200 области данных отделены направляющими шаблонами. Область данных окружена шаблоном поиска, вокруг которого со всех четырех сторон должна быть свободная зона. На рисунке 1 приведен пример символа уровня ECC 140 и два примера символа версии ECC 200.



a) темное изображение на светлом фоне для версии ECC 200



b) светлое изображение на темном фоне для версии ECC 200



c) темное изображение на светлом фоне для уровня ECC 140

Рисунок 1 — Пример кодирования сообщения «A1B2C3D4E5F6G7H8I9J0K1L2» в символе версии ECC 200 (рисунки a) и b)) и уровня ECC 140 (рисунок c))

4.3.1 Шаблон поиска

Шаблоном поиска является периметр области данных шириной в один модуль. Две смежные стороны — левая и нижняя, являются сплошными темными линиями и формируют L-образную границу. Они используются, прежде всего, для определения реального размера, ориентации и искажений символа. Две противоположные стороны состоят из чередующихся темных и светлых модулей. Они используются, прежде всего, для определения структуры символа, состоящей из ячеек, но также могут применяться для определения физического размера и искажений символа. Наличие свободной зоны обозначено на рисунке 1 угловыми метками.

4.3.2 Размеры и емкость символов

Символы версии ECC 200 состоят из четного числа строк и четного числа столбцов. Символы версии ECC 200 могут быть квадратной формы с размерами (в модулях) от 10 × 10 до 144 × 144 без учета свободных зон, либо прямоугольной формы размерами (в модулях) от 8 × 18 до 16 × 48 без учета свободных зон. Все символы версии ECC 200 можно распознать по светлому модулю в правом верхнем углу. Полный перечень атрибутов символа версии ECC 200 приведен в 5.5 (таблица 7).

Символы версии ECC 000-140 состоят из нечетного числа строк и нечетного числа столбцов. Символы версии ECC 000-140 имеют квадратную форму размерами от 9 × 9 до 49 × 49 модулей без учета свободных зон. Данные символы можно распознать по темному модулю в правом верхнем углу. Полный перечень атрибутов символов версии ECC 000-140 приведен в приложении G.

5 Требования к символам версии ECC 200

5.1 Основные положения процедуры кодирования

Настоящий раздел содержит общие сведения о процедуре кодирования. В следующих разделах приведено более детальное рассмотрение указанной процедуры. Пример кодирования для символа версии ECC 200 приведен в приложении О. Преобразование данных пользователя в символ версии ECC 200 происходит в следующей последовательности:

Этап 1. Кодирование данных

Анализируют поток данных для определения разнообразия типов различных знаков, подлежащих кодированию. Символы версии ECC 200 содержат различные схемы кодирования, которые позволяют найденные множества знаков преобразовать в кодовые слова более эффективно по сравнению со схемой кодирования, принятой по умолчанию. Вводят дополнительные кодовые слова для переключения между схемами кодирования и для выполнения других функций. Добавляют необходимое количество знаков-заполнителей для образования требуемого числа кодовых слов. Если пользователь не установил размер матрицы, то выбирают наименьший размер, в котором могут быть размещены данные. Полный перечень размеров матриц приведен в 5.5 (таблица 7).

Таблица 1 — Схемы кодирования для символов версии ECC 200

Наименование схемы кодирования	Знаки	Число битов на один знак данных
ASCII (КОИ-7/ КОИ-8)	Сдвоенные разряды чисел	4
	Знаки ASCII (КОИ-7) с десятичными значениями от 0 до 127	8
	Знаки расширенного набора ASCII (КОИ-8) с десятичными значениями от 128 до 255	16
C40	Цифры и прописные латинские буквы	5,33
	Специальные знаки и строчные латинские буквы	10,66[a]
Text	Цифры и строчные латинские буквы	5,33
	Специальные знаки и прописные латинские буквы	10,66[b]
ANSI X12	Набор знаков данных для электронного обмена данными по ANSI X12 EDI	5,33
EDIFACT	Знаки ASCII (КОИ-7) с десятичными значениями от 32 до 94	6
По основанию 256	Любые байты с десятичными значениями от 0 до 255	8

[a] Кодируют как два значения в схеме кодирования C40 с использованием знака регистра (Shift).
[b] Кодируют как два значения в схеме кодирования Text с использованием знака регистра (Shift).

Этап 2. Формирование кодовых слов проверки и исправления ошибок

Для символов, содержащих более 255 кодовых слов, поток кодовых слов подразделяют на чередующиеся блоки, чтобы дать возможность обработки алгоритмами исправления ошибок (приложение А). Для каждого блока формируют кодовые слова исправления ошибок. Результатом этого процесса является удлинение потока кодовых слов на число кодовых слов исправления ошибок. Кодовые слова исправления ошибок помещают после кодовых слов данных.

Этап 3. Размещение модулей в матрице

Модули кодовых слов размещают в матрице. В матрицу вставляют модули направляющих шаблонов (при их наличии). Вокруг матрицы добавляют модули шаблона поиска.

5.2 Кодирование данных

5.2.1 Общие положения

Данные можно кодировать с использованием любой комбинации из шести схем кодирования (таблица 1), при этом кодирование по схеме ASCII (КОИ-7/КОИ-8) является основной схемой. Остальные схемы кодирования вызываются из схемы кодирования ASCII (КОИ-7/КОИ-8) с последующим возвратом к этой же схеме. Следует учитывать эффективность уплотнения (число битов на знак данных), приведенную в таблице 1. Лучшей схемой для выбранного набора данных может оказаться не та, у которой на знак данных приходится наименьшее число битов. Если требуется наибольшая степень уплотнения, то надо принимать в расчет служебную информацию для переключения между схемами кодирования и наборами знаков внутри одной схемы кодирования (приложение Р). Следует также учитывать, что даже если число кодовых слов минимизировано, поток кодовых слов может нуждаться в расширении для полного заполнения символа. Дополнение осуществляют путем использования знаков-заполнителей.

5.2.2 Интерпретация знаков по умолчанию

Интерпретация знаков по умолчанию для знаков с десятичными значениями от 0 до 127 должна соответствовать версии КОИ-7 по ИСО/МЭК 646, а для знаков с десятичными значениями от 128 до 255 — версии КОИ-8 по ИСО 8859-1 «Латинский алфавит № 1». Графические представления знаков данных, приведенных в настоящем стандарте, соответствуют интерпретации по умолчанию. Эта интерпретация может быть изменена с помощью переключающих последовательностей интерпретации в расширенном канале (5.4). Интерпретацией по умолчанию является ECI 000003.

5.2.3 Схема кодирования ASCII (КОИ-7/КОИ-8)

Схема кодирования ASCII (КОИ-7/КОИ-8) представляет собой набор знаков по умолчанию для первого знака символа в символах любого размера. С помощью указанной схемы кодируют знаки ASCII (КОИ-7) и расширенного набора ASCII (КОИ-8), числовые данные двойной плотности и управляющие знаки символики. Управляющие знаки символики включают в себя функциональные знаки, знак-заполнитель и знаки-переключатели на другие кодовые наборы. Знаки данных ASCII (КОИ-7) кодируют как кодовые слова с десятичными значениями от 1 до 128 (десятичное значение знака КОИ-7 плюс 1). Знаки данных расширенного набора ASCII (КОИ-8) с десятичными значениями от 128 до 255 кодируют с использованием управляющего знака верхнего регистра (Upper Shift) (5.2.4.2). Пары цифр от 00 до 99 кодируют кодовыми словами от 130 до 229 (числовое значение плюс 130). Присвоенные значения кодовых слов для схемы кодирования ASCII (КОИ-7/КОИ-8) приведены в таблице 2.

Таблица 2 — Значения кодовых слов в схеме кодирования ASCII (КОИ-7/КОИ-8)

Значение кодового слова	Знак данных или функция
1—128	Знаки данных ASCII (КОИ-7) (десятичное значение знака +1)
129	Знак-заполнитель
130—229	Пары цифр от 00 до 99 (числовое значение + 130)
230	Знак фиксации схемы кодирования С40
231	Знак фиксации схемы кодирования по основанию 256
232	Знак FNC1
233	Знак структурированного соединения
234	Знак программирования устройства считывания
235	Знак верхнего регистра (переход к расширенному набору ASCII (КОИ-8))
236	Знак Макро 05
237	Знак Макро 06
238	Знак фиксации схемы кодирования X12
239	Знак фиксации схемы кодирования Text
240	Знак фиксации схемы кодирования EDIFACT
241	Знак интерпретации в расширенном канале (ECI)
242—255	Не подлежит использованию в схеме кодирования ASCII (КОИ-7/КОИ-8)

5.2.4 Управляющие знаки символики

В символах версии ECC 200 есть несколько специальных управляющих знаков символики, имеющих особое значение для схемы кодирования. Эти знаки должны использоваться для сообщения команды декодеру на выполнение определенных функций или передачи управляющему компьютеру специальных данных (5.2.4.1 — 5.2.4.9). Эти управляющие знаки символики, за исключением знаков с десятичными значениями от 242 до 255, присутствуют в кодовом наборе ASCII (КОИ-7/КОИ-8) (таблица 2).

5.2.4.1 Знаки фиксации схемы кодирования (Latch)

Для переключения из схемы кодирования ASCII (КОИ-7/КОИ-8) в любую иную схему кодирования используют знаки фиксации. Все кодовые слова после знака фиксации должны кодироваться в соответствии с новой схемой кодирования. Различные схемы кодирования имеют свои способы возврата к кодовому набору ASCII (КОИ-7/КОИ-8).

5.2.4.2 Знак верхнего регистра (Upper Shift)

Знак верхнего регистра используется в комбинации с знаком ASCII (КОИ-7) для кодирования знака расширенного набора ASCII (КОИ-8) с десятичными значениями от 128 до 255. Знак расширенного набора ASCII (КОИ-8), кодируемый в схемах кодирования ASCII (КОИ-7/КОИ-8), C40 или Text, требует наличия предшествующего знака верхнего регистра, после которого стоит знак ASCII (КОИ-7), десятичное значение которого уменьшено на 128. Эту пару кодируют в соответствии с правилами схемы кодирования. В схеме кодирования ASCII (КОИ-7/КОИ-8) знак верхнего регистра представлен кодовым словом со значением 235. Уменьшенное десятичное значение знака данных (т.е. десятичное значение знака расширенного набора ASCII (КОИ-8) минус 128) преобразуют в значение кодового слова путем прибавления к его значению единицы. Например, для кодирования знака ¥ (ДЕНЕЖНЫЙ ЗНАК ИЕНЫ, десятичное значение которого равно 165) следует после знака верхнего регистра (кодовое слово со значением 235) поставить знак ASCII (КОИ-7) с десятичным значением 37 (165 — 128), которое кодируется как кодовое слово со значением 38. При наличии протяженных последовательностей знаков данных расширенного набора ASCII (КОИ-8), более эффективное кодирование может быть достигнуто путем использования знака фиксации схемы кодирования по основанию 256.

5.2.4.3 Знак-заполнитель (Pad)

Если кодируемых данных, независимо от используемой схемы кодирования, не хватает для полного заполнения символа для данных, то оставшаяся часть символа для данных должна быть заполнена знаками-заполнителями. Знаки-заполнители должны использоваться исключительно для указанной цели. Перед вводом знака-заполнителя, необходимо вернуться к схеме кодирования ASCII (КОИ-7/КОИ-8) из любой другой используемой схемы кодирования.

Алгоритм рандомизации с шаблоном из 253 состояний применяют к знакам-заполнителям, ко всей последовательности знаков-заполнителей, начиная со второго знака-заполнителя и до конца символа (приложение В.1).

5.2.4.4 Знак интерпретации в расширенном канале (ECI)

Знак интерпретации в расширенном канале (ECI) используют для смены интерпретации, принятой по умолчанию, на иную интерпретацию, применяемую для кодирования данных. Протокол интерпретации в расширенном канале является общим для многих символик и его применение к символике версии ECC 200 более полно определено в 5.4. После знака ECI обязательно должны следовать одно, два или три кодовых слова, которые идентифицируют конкретную активизируемую ECI. Новая ECI действует до конца кодируемых данных или до тех пор, пока другой знак ECI не вызовет иную интерпретацию.

5.2.4.5 Знаки регистра (Shift) в схемах кодирования C40 и Text

В схемах кодирования C40 и Text используют три специальных знака, называемые знаками регистра, в качестве префикса к одному из 40 значений для кодирования примерно трех четвертей набора знаков ASCII (КОИ-7). Это позволяет добиться более компактного кодирования оставшихся знаков ASCII (КОИ-7) с помощью одиночных значений[1)].

5.2.4.6 Знак FNC1 как идентификатор альтернативного типа данных

Для кодирования данных, соответствующих специальным международным отраслевым стандартам, одобренных AIM Inc, знак FNC1 должен присутствовать в позиции первого или второго знака символа (либо пятой или шестой позиции данных в первом символе структурированного соединения символов). Знак FNC1, кодируемый в позиции любого иного знака символа, используют как разделитель полей, и он подлежит передаче как управляющий знак $^{G}_{S}$ (знак ASCII (КОИ-7) с десятичным значением 29).

[1)] *Без предшествующего знака «Регистр» перед каждым знаком.*

5.2.4.7 Знаки Макро (Macro)

Символика Data Matrix обеспечивает представление специальных международных отраслевых головной и конечной меток в одном знаке символа, которое сокращает число знаков символа, необходимых для кодирования данных в символе при использовании установленных структурированных форматов. Любой знак Макро применяют только в позиции первого знака символа. Эти знаки не должны использоваться вместе со структурированным соединением (Structured Append). Функции знаков Макро приведены в таблице 3. Головная метка должна быть включена в передаваемый поток данных в виде префикса, а конечная метка — суффикса[1]. Если используют идентификатор символики, то он должен предшествовать головной метке.

Таблица 3 — Функции знаков Макро

Значение кодового слова знака Макро	Обозначение знака	Интерпретация метки	
		Головная метка	Конечная метка
236	Макро 05	[ )>$^{R}_{S}$05$^{G}_{S}$ [2]	$^{R}_{S}E_{OT}$ [3]
237	Макро 06	[ )>$^{R}_{S}$06$^{G}_{S}$ [4]	$^{R}_{S}E_{OT}$

5.2.4.8 Знак структурированного соединения (Structured Append)

Знак структурированного соединения используют для указания того, что символ является частью последовательности символов структурированного соединения в соответствии с 5.6.

5.2.4.9 Знак программирования устройства считывания

Знак программирования устройства считывания указывает на то, что в символе закодировано сообщение, предназначенное для программирования устройства считывания. Знак программирования устройства считывания должен быть первым кодовым словом символа и не подлежит использованию совместно со структурированным соединением.

5.2.5 Схема кодирования С40

Схему кодирования С40 применяют для оптимизации кодирования данных, состоящих из последовательности, включающей прописные буквы латинского алфавита и числа (включая знак ПРОБЕЛ). Данная схема позволяет также кодировать и другие знаки путем использования знаков регистра в комбинации с другими знаками данных.

Знаки данных в схеме кодирования С40 разделены на четыре набора. Знаки из первого набора, называемого основным набором, содержат три специальных знака регистра, знак ПРОБЕЛ и знаки ASCII (КОИ-7) с A по Z и с 0 по 9. Каждому знаку данных соответствует единственное значение схемы кодирования С40 (далее — значение С40). Знаки других наборов присваивают одному из трех знаков регистра, которые указывают на один из трех оставшихся наборов и сопровождаются одним из значений С40 (приложение С, таблица С.1).

В результате первого этапа кодирования каждый знак данных преобразуют в одно значение С40 или в пару значений С40. Затем полную строку значений С40 разбивают на группы по три значения (если в конце данных остается одно или два значения, то применяют специальные правила, приведенные в 5.2.5.2.). После этого три значения (С1, С2, С3) кодируют как одно 16-битовое значение по формуле (1600×С1) + (40×С2) + С3 + 1. В завершение каждое 16-битовое значение кодируют в двух кодовых словах, представляющих собой восемь старших битов и восемь младших битов.

5.2.5.1 Переключение на схему кодирования С40 и обратно

На схему кодирования С40 можно переключиться из схемы кодирования ASCII (КОИ-7/КОИ-8), используя соответствующее кодовое слово фиксации схемы кодирования С40 со значением 230. Кодовое слово со значением 254, непосредственно следующее за парой кодовых слов в схеме кодирования С40, действует как отказ от фиксации (Unlatch) для возврата к схеме кодирования ASCII (КОИ-7/КОИ-8). В противном случае кодирование по схеме кодирования С40 сохраняется до окончания данных, закодированных в символе.

---

[1] *Суффикс является завершающим дополнением к потоку данных*

[2] *Знаки КОИ-7 (ASCII) с десятичными значениями 91, 41 ,62, 30 ,48 ,53, 29.*

[3] *Знаки КОИ-7 (ASCII) с десятичными значениям 30, 4.*

[4] *Знаки КОИ-7 (ASCII) с десятичными значениями 91, 41, 62, 30, 48, 54, 29.*

5.2.5.2 Правила кодирования С40

Каждая пара кодовых слов представляет собой 16-битовое значение, в котором первое кодовое слово соответствует восьми старшим битам, а второе — восьми младшим битам. Три значения С40 (С1, С2, С3) кодируют по формуле (1600×С1)+(40×С2)+С3+1.

В результате получают значения от 1 до 64000. Уплотнение трех значений С40 в два кодовых слова представлено на рисунке 2.

Исходные знаки данных	AIM
Полученные значения С40	14, 22, 26
Вычисление 16-битового значения	(1600×14)+(40×22)+26+1 =23307
Определение первого кодового слова: (16-битовое значение) div 256	23307 div 256 = 91
Определение второго кодового слова: (16-битовое значение) mod 256	23307 mod 256 =11
Итоговые кодовые слова	91, 11

Рисунок 2 — Пример кодирования по схеме кодирования С40

Для кодирования знаков, принадлежащих наборам Регистр 1 (Shift 1), Регистр 2 (Shift 2) и Регистр 3 (Shift 3), сначала следует закодировать соответствующий знак регистра, а затем — значение С40 для данных. Кодирование по схеме С40 может действовать до окончания кодовых слов символа, кодирующих данные.

В случае, если в символе остается только один или два знака символа до начала следования кодовых слов исправления ошибки, то следует придерживаться следующих правил:

a) если остаются два знака символа и кодированию подлежат три оставшихся значения С40 (которые могут включать как знаки данных, так и знаки регистра (Shift)), то эти три значения С40 кодируют в двух последних знаках символа. Заключительного кодового слова отказа от фиксации схемы кодирования (Unlatch) не требуется;

b) если остаются два знака символа и кодированию подлежат два оставшихся значения С40 (первое из которых может быть знаком регистра (Shift) или знаком данных, а второе должно представлять знак данных), то эти два оставшиеся значения С40 кодируют с добавлением значения заполнителя С40, равного 0 (из набора Регистр 1) в двух последних знаках символа. Кодового слова отказа от фиксации схемы кодирования (Unlatch) также не требуется;

c) если остаются два знака символа для кодирования одного оставшегося значения С40 (знака данных), то в первом из двух оставшихся знаков символа (предпоследнем знаке символа) кодируют отказ от фиксации схемы кодирования (Unlatch), а в последнем знаке символа кодируют знак данных по схеме кодирования ASCII (КОИ-7/КОИ-8);

d) если остается один знак символа для кодирования одного оставшегося значения С40 (знака данных), то в последнем знаке символа кодируют знак данных по схеме кодирования ASCII (КОИ-7/КОИ-8). Знак отказа от фиксации схемы кодирования (Unlatch) не кодируют, его наличие подразумевается перед последним знаком символа.

Во всех остальных случаях либо используют знак отказа от фиксации схемы кодирования (Unlatch) для выхода из схемы кодирования С40 перед окончанием символа, либо применяют символ большего размера для кодирования данных.

5.2.5.3 Использование знака верхнего регистра (Upper Shift) в схеме кодирования С40

В схеме кодирования С40 знак верхнего регистра (Upper Shift) не является функциональным знаком символики, а используется как знак регистра (Shift) внутри данного кодового набора. Для кодирования знаков расширенного набора ASCII (КОИ-8) с десятичными значениями от 128 до 255 необходимо закодировать три или четыре значения С40 в соответствии со следующими требованиями.

Если [десятичное значение знака расширенного набора ASCII (КОИ-8) минус 128] принадлежит основному набору, то используют запись:

[1 (значение знака Регистр 2 (Shift))] [30 (значение знака верхнего регистра (UperShift))] [V (десятичное значение знака расширенного набора ASCII (КОИ-8) минус 128)].

В противном случае запись приобретает следующий вид:

[1 (значение знака Регистр 2)] [30 (значение знака верхнего регистра)] [0, 1 или 2 (значения знаков Регистр 1, 2 или 3)] [V (десятичное значение знака расширенного набора ASCII (КОИ-8) минус 128)].

В данных записях число, приведенное в квадратных скобках, соответствует значению согласно приложению С.1, соответствующее значение С40 обозначено V.

5.2.6 Схема кодирования Text

Схема кодирования Text предназначена для кодирования обычного печатного текста, состоящего в основном из знаков нижнего регистра (строчных букв латинского алфавита, цифр, знака ПРОБЕЛ). По структуре она похожа на кодовый набор, используемый в схеме кодирования С40, за исключением того, что строчные буквы нижнего регистра кодируют напрямую (без переключения регистра). Знаки верхнего регистра (прописные буквы латинского алфавита, цифры, специальные графические знаки и знак ПРОБЕЛ) предваряют знаком регистра 3. Полный кодовый набор знаков схемы кодирования Text приведен в приложении С (таблица С.2).

5.2.6.1 Переключение на схему кодирования Text и обратно

На схему кодирования Text можно переключиться из схемы кодирования ASCII (КОИ-7/КОИ-8), используя соответствующее кодовое слово фиксации схемы кодирования с десятичным значением 239. Кодовое слово значением 254, непосредственно следующее за парой кодовых слов в схеме кодирования Text, действует как кодовое слово отказа от фиксации (Unlatch) для возврата в схему кодирования ASCII (КОИ-7/КОИ-8). В противном случае схема кодирования Text действует до окончания данных, кодируемых в символе.

5.2.6.2 Правила кодирования в схеме кодирования Text

Применяют те же правила, что и в схеме кодирования С40.

5.2.7 Схема кодирования ANSI X12

Схему кодирования ANSI X12 применяют для кодирования знаков, используемых при стандартном электронном обмене данными по ANSI X12, в которой три знака данных размещают с уплотнением в двух кодовых словах и которая в некоторой степени подобна схеме кодирования С40. Схема кодирования ANSI X12 позволяет кодировать буквы верхнего регистра (прописные латинские буквы), цифры, знак ПРОБЕЛ и три стандартных ограничительных и разделительных знака в соответствии с ANSI X12. Соответствие кодов по ANSI X12 приведено в таблице 4. В наборе кодируемых знаков по ANSI X12 отсутствуют знаки регистра (Shift).

Таблица 4 — Набор кодируемых знаков по ANSI X12

Значение знака X12	Кодируемые знаки	Десятичные значения знака ASCII (КОИ-7)
0	X12 ограничитель сегмента <CR>[1)]	13
1	X12 разделитель сегментов * (ЗВЕЗДОЧКА)	42
2	X12 разделитель подэлементов > (БОЛЬШЕ)	62
3	ПРОБЕЛ	32
4—13	от 0 до 9	48—57
14—39	от A до Z	65—90

5.2.7.1 Переключение на схему кодирования ANSI X12 и обратно

На схему кодирования ANSI X12 можно переключиться из схемы кодирования ASCII (КОИ-7/КОИ-8), используя соответствующее кодовое слово фиксации схемы кодирования (Latch) (значение 238). Кодовое слово значением 254, непосредственно следующее за парой кодовых слов схемы кодирования ANSI X12, действует как кодовое слово отказа от фиксации (Unlatch) для возврата в схему кодирования ASCII (КОИ-7/КОИ-8). В противном случае схема кодирования ANSI X12 действует до окончания данных, кодируемых в символе.

[1)] *Управляющий знак CR соответствует обозначению знака ВК «ВОЗВРАТ КАРЕТКИ» по ГОСТ 27465-87 «Системы обработки информации. Символы. Классификация, наименование и обозначение».*

5.2.7.2 Правила кодирования в соответствии со схемой кодирования ANSI X12

Применяют правила, установленные для схемы кодирования С40. Исключение составляет окончание кодирования данных ANSI X12. Если знаки данных не полностью заполняют пары кодовых слов, то сразу за последней полной парой кодовых слов следует использовать переключение в схему кодирования ASCII (КОИ-7/КОИ-8) с помощью кодового слова значением 254 и продолжить использование схемы кодирования ASCII (КОИ-7/КОИ-8) за исключением случая, когда остается единственный конечный знак символа (кодовое слово) перед первым кодовым словом исправления ошибки. Этот единственный знак символа кодируется по схеме кодирования ASCII (КОИ-7/КОИ-8) без использования кодового слова отказа от фиксации (Unlatch).

5.2.8 Схема кодирования EDIFACT

Схема кодирования EDIFACT включает в себя 63 знака КОИ-7 (ASCII) с десятичными значениями от 32 до 94, а также знак отказа от фиксации (двоичное значение 011111) для возврата в схему кодирования ASCII (КОИ-7/КОИ-8). Схема кодирования EDIFACT позволяет кодировать четыре знака данных в трех кодовых словах. Знаки данных включают в себя все цифры, буквы латинского алфавита и специальные графические знаки (знаки пунктуации), определенные в наборе знаков «EDIFACT Level A» без знаков регистра (Shift), используемых в схеме кодирования С40.

5.2.8.1 Переключение на схему кодирования EDIFACT и обратно

На схему кодирования EDIFACT можно переключиться из схемы кодирования ASCII (КОИ-7/КОИ-8), используя соответствующее кодовое слово фиксации схемы кодирования (Latch) значением 240. Знак отказа от фиксации в схеме кодирования EDIFACT следует использовать в качестве ограничителя окончания схемы кодирования EDIFACT для возврата в схему кодирования ASCII (КОИ-7/КОИ-8).

5.2.8.2 Правила кодирования в соответствии со схемой кодирования EDIFACT

Набор знаков в схеме кодирования EDIFACT приведен в приложении С, таблица С.3. Существует простое соответствие между 6-битовыми значениями знаков по EDIFACT и 8-битовыми байтами знака расширенного набора ASCII (КОИ-8). При построении 6-битового значения знака по EDIFACT исключают два бита старших разрядов 8-битового байта в соответствии с рисунком 3. Строки из четырех знаков со значениями по EDIFACT кодируют в три кодовых слова. В процессе простого кодирования два бита старших разрядов удаляют из 8-битового байта. Оставшийся 6-битовый байт является значением по EDIFACT и должен быть непосредственно закодирован в кодовом слове (рисунок 4).

Знак данных	Значение знака расширенного набора ASCII (КОИ-8)		Значение знака по EDIFACT
	Десятичное значение	8-битовое двоичное значение	
A	65	01000001	000001
9	57	00111001	111001
Примечание — В процессе декодирования, если начальный бит (6-й разряд) равен 1, то для построения 8-битового байта требуется вставить в качестве префикса биты 00. Если начальный бит (6-й разряд) равен нулю, то для построения 8-битового байта надо вставить в качестве префикса биты 01. Исключением является знак со значением по EDIFACT 011111, который является управляющим знаком символики отказа от фиксации (Unlatch) для возврата в схему кодирования ASCII (КОИ-7/КОИ-8).			

Рисунок 3 — Соответствие значений знаков по EDIFACT и значений 8-битовых байтов

Знаки данных	D	A	T	A
Исходные двоичные значения (по таблице С.3)	00 01 00	00 00 01	01 01 00	00 00 01
Разделение по три 8-битовых байта	00 01 00 00	00 01 01 01	00 00 00 01	
Итоговые значения кодовых слов	16	21	1	

Рисунок 4 — Пример кодирования по EDIFACT

Когда кодирование EDIFACT завершается знаком отказа от фиксации схемы кодирования (Unlatch), любые биты, оставшиеся в одиночном знаке символа, следует заполнять нулями. Схема кодирования ASCII (КОИ-7/КОИ-8) начинается со следующего знака символа. Если схема кодирования EDIFACT действует до конца символа, и до первого знака исправления ошибки осталось закодировать только одно или два кодовых слова, оставшихся за последним триплетом кодовых слов по схеме кодирования EDIFACT, их следует кодировать по схеме кодирования ASCII (КОИ-7/КОИ-8) без использования знака отказа от фиксации (Unlatch).

5.2.9 Схема кодирования по основанию 256

Схему кодирования по основанию 256 используют для кодирования любых 8-битовых байтов данных, включая интерпретации в расширенном канале (ECI), и двоичных данных. Интерпретация, используемая по умолчанию, определена в 5.2.2. Алгоритм рандомизации с шаблоном из 255 состояний применяют к каждой последовательности по основанию 256, встречающейся в закодированных данных (приложение В.2). Схема начинает действовать после знака фиксации схемы кодирования по основанию 256 и заканчивается на последнем знаке, определенном длиной поля в схеме кодирования по основанию 256.

5.2.9.1 Переключение на схему кодирования по основанию 256 и обратно

На схему кодирования по основанию 256 можно переключиться из схемы кодирования ASCII (КОИ-7/КОИ-8), используя соответствующее кодовое слово фиксации схемы кодирования значением 231. По окончании данных, закодированных в соответствии со схемой кодирования по основанию 256, возврат к схеме кодирования ASCII (КОИ-7/КОИ-8) осуществляется автоматически. Обращение к интерпретации в расширенном канале (ECI), отличающейся от принятой по умолчанию, должно быть выполнено до переключения на схему кодирования по основанию 256. Последовательность ECI не требуется располагать непосредственно перед переключением в схему кодирования по основанию 256.

5.2.9.2 Правила кодирования в соответствии со схемой кодирования по основанию 256

После переключения на схему кодирования по основанию 256 первые одно (*d1*) или два (*d1*, *d2*) кодовых слова устанавливают длину поля данных в байтах. Определение степени соответствия между длиной поля и значениями *d1* и *d2* приведено в таблице 5. Далее записываются значения данных в байтах.

Таблица 5 — Длина поля в схеме кодирования по основанию 256

Длина поля	Значения *d1*, *d2*	Допустимые значения *d*
До конца символа	*d1* = 0	*d1* = 0
От 1 до 249	*d1* = заданная длина	*d1* = от 1 до 249
От 250 до 1555	*d1* = (заданная длина DIV 250) +249	*d1* = от 250 до 255
	d2 = заданная длина MOD 250	*d2* = от 0 до 249

5.3 Рекомендации пользователям

Символика версии ECC 200 предлагает гибкие способы кодирования данных. К альтернативным наборам знаков следует обращаться с использованием протокола интерпретации в расширенном канале (ECI). Данные могут быть закодированы в символ квадратной или прямоугольной формы. Если длина сообщения превышает емкость символа, то оно может быть закодировано с использованием последовательности структурированного соединения нескольких (до 16) отдельных, но логически связанных символов версии ECC 200 (5.6).

5.3.1 Выбор пользователем интерпретации в расширенном канале (ECI)

Использование альтернативной интерпретации в расширенном канале (ECI) для задания определенной кодовой страницы (набора) или более специфичной интерпретации данных требует вызова дополнительных кодовых слов для активизации этой возможности. Использование протокола интерпретации в расширенном канале (ECI) (5.4) обеспечивает возможность кодирования в данных знаков алфавитов, отличающихся от латинского (по ИСО/МЭК 8859-1 Латинский алфавит № 1), поддерживаемого интерпретацией по умолчанию (последовательность ECI 000003).

5.3.2 Выбор пользователем формы и размера символа

Версия ECC 200 имеет двадцать четыре квадратных и шесть прямоугольных конфигураций символа. Можно выбрать подходящий размер и форму символа, в зависимости от требований к его практическому применению; технические требования к данным конфигурациям приведены в 5.5.

5.4 Интерпретация в расширенном канале

Протокол интерпретации в расширенном канале (ECI) позволяет включать в выходной поток данных знаки различных интерпретаций, отличающиеся от набора знаков по умолчанию. Протокол ECI единообразно определен для ряда символик. В символике Data Matrix поддерживаются четыре распространенных типа интерпретаций:

a) международные наборы знаков (или кодовые страницы);

b) интерпретации общего назначения, такие как шифрование и уплотнение;

c) определяемые пользователем интерпретации для замкнутых систем применения;

d) управляющая информация для структурированного соединения в небуферизованном режиме.

Протокол интерпретации в расширенном канале полностью установлен в стандарте AIM Inc. ITS/04-001 «Интерпретации в расширенном канале. Часть 1» («International Technical Specification — Extended Channel Interpretation — Parth 1»). Протокол обеспечивает последовательный метод установления специфических интерпретаций значений байтов перед печатью и после декодирования. Конкретную интерпретацию в расширенном канале идентифицируют с помощью 6-разрядного числа, которое в символике Data Matrix кодируют знаком ECI, за которым следует от одного до трех кодовых слов. Специальные интерпретации приведены в документе AIM Inc. «Интерпретации в расширенном канале. Часть 3» («Extended Chanel Interpretations — Part 3 — Register»). Интерпретация в расширенном канале может использоваться только с устройствами считывания, позволяющими передавать идентификаторы символики. Устройства считывания, которые не могут передавать идентификаторы символики, не обеспечивают передачу данных из любого символа, содержащего ECI. Исключение может быть сделано только в случае, если интерпретация в расширенном канале может быть полностью обработана самим устройством считывания.

Протокол интерпретации в расширенном канале используют только в символах версии ECC 200. Заданная интерпретация в расширенном канале может быть вызвана в любом месте закодированного сообщения.

5.4.1 Кодирование интерпретации в расширенном канале

Разнообразные схемы кодирования символики Data Matrix версии ECC 200 (таблица 1) могут применяться при любой интерпретации в расширенном канале. Вызов ECI может быть осуществлен только из схемы кодирования ASCII (КОИ-7/КОИ-8), после которого допускается переключение между любыми схемами кодирования. Используемый способ кодирования строго определен 8-битовыми значениями данных и он не зависит от действующей ECI. Например последовательность знаков с десятичными значениями в диапазоне от 48 до 57 может быть наиболее эффективно закодирована в цифровом режиме, даже если они не будут интерпретироваться как числа. Назначение ECI вводят с помощью кодового слова значением 241 (знак ECI) в схеме кодирования ASCII (КОИ-7/КОИ-8). Одно, два или три дополнительных кодовых слова используют для кодирования номера назначения ECI (ECI Assignment member). Правила кодирования приведены в таблице 6.

Таблица 6 — Кодирование номеров назначения ECI в символике версии ECC 200

Номер назначения ECI	Последовательность кодовых слов	Значения кодовых слов	Область значений
От 000000 до 000126	C_0	241	
	C_1	*ECI_no +1*[1]	C_1 = (от 1 до 127)
От 000127 до 016382	C_0	241	
	C_1	(*ECI_no* – 127)div254 + 128	C_1 = (от 128 до 191)
	C_2	(*ECI_no* – 127) mod 254 + 1	C_2 = (от 1 до 254)
От 0016383 до 999999	C_0	241	
	C_1	(*EC1_no* – 16383)div64516 + 192	C_1 = (от 192 до 207)
	C_2	[(*ECI_no* – 16383) div 254] mod 254 + 1	C_2 = (от 1 до 254)
	C_3	(*ECI_no* – 16383) mod 254 + 1	C_3 = (от 1 до 254)

[1] *ECI_no +1 — заданный номер назначения ECI.*

Следующие примеры приведены для иллюстрации кодирования:

номер назначения ECI = 015000

Кодовые слова:

[241] [(15000 – 127) div 254 + 128] [(15000 – 127) mod 254 + 1]=[241][58 + 128][141 + 1]=
=[241][186][142]

номер назначения ECI = 090000

Кодовые слова:

[241][(90000 – 16383) div 64516 + 192][((90000 – 16383)div 254)mod 254 + 1][(90000 – 16383) mod 254 + + 1] = [241][1 + 192][289 mod 254 + 1][211 + 1] = [241][193][36][212]

5.4.2 ECI и структурированное соединение

ECI могут появляться в любом месте сообщения, закодированного в одиночном символе или в символе структурированного соединения (5.6) набора символов Data Matrix. Любая активизированная ECI сохраняет действие либо до конца закодированных данных, либо до появления другой ECI. Таким образом, интерпретация в заданной ECI может распространяться на два или более символов.

5.4.3 Протокол после декодирования

Протокол передачи данных ECI определен в 11.4. При применении интерпретаций в расширенном канале следует использовать идентификаторы символики (11.5) и соответствующий идентификатор символики должен передаваться перед декодированными данными.

5.5 Атрибуты символа версии ECC 200

5.5.1 Размер и емкость символа

В символике версии ECC 200 доступны 24 квадратных и 6 прямоугольных символов, указанные в таблице 7.

Таблица 7 — Атрибуты символов ECC 200

Размер[a] символа		Область данных		Размер координатной матрицы	Общее число кодовых слов		Число кодовых слов в блоке Рида-Соломона		Число чередующихся блоков	Максимальная емкость символа для данных			Кодовые слова исправления ошибок, %	Максимальное число восстановленных кодовых слов[b], ошибок/стираний
Число строк	Число столбцов	Размер	Число областей данных		данных	исправления ошибок	данных	исправления ошибок		число числовых разрядов	число латинских букв и цифр[d]	число байтов		
Символы квадратной формы														
10	10	8×8	1	8×8	3	5	3	5	1	6	3	1	62,5	2/0
12	12	10×10	1	10×10	5	7	5	7	1	10	6	3	58,3	3/0
14	14	12×12	1	12×12	8	10	8	10	1	16	10	6	55,6	5/7
16	16	14×14	1	14×14	12	12	12	12	1	24	16	10	50	6/9
18	18	16×16	1	16×16	18	14	18	14	1	36	25	16	43,8	7/11
20	20	18×18	1	18×18	22	18	22	18	1	44	31	20	45	9/15
22	22	20×20	1	20×20	30	20	30	20	1	60	43	28	40	10/17
24	24	22×22	1	22×22	36	24	36	24	1	72	52	34	40	12/21
26	26	24×24	1	24×24	44	28	44	28	1	88	64	42	38,9	14/25
32	32	14×14	4	28×28	62	36	62	36	1	124	91	60	36,7	18/33
36	36	16×16	4	32×32	86	42	86	42	1	172	127	84	32,8	21/39
40	40	18×18	4	36×36	114	48	114	48	1	228	169	112	29,6	24/45
44	44	20×20	4	40×40	144	56	144	56	1	288	214	142	28	28/53
48	48	22×22	4	44×44	174	68	174	68	1	348	259	172	28,1	34/65
52	52	24×24	4	48×48	204	84	102	42	2	408	304	202	29,2	42/78

Окончание таблицы 7

Размер[a] символа		Область данных		Размер координатной матрицы	Общее число кодовых слов		Число кодовых слов в блоке Рида-Соломона		Число чередующихся блоков	Максимальная емкость символа для данных			Кодовые слова исправления ошибок, %	Максимальное число восстановленных кодовых слов[b], ошибок/стираний
Число строк	Число столбцов	Размер	Число областей данных		данных	исправления ошибок	данных	исправления ошибок		число числовых разрядов	число латинских букв и цифр[d]	число байтов		
64	64	14×14	16	56×56	280	112	140	56	2	560	418	277	28,6	56/106
72	72	16×16	16	64×64	368	144	92	36	4	736	550	365	28,1	72/132
80	80	18×18	16	72×72	456	192	114	48	4	912	682	453	29,6	96/180
88	88	20×20	16	80×80	576	224	144	56	4	1152	862	573	28	112/212
96	96	22×22	16	88×88	696	272	174	68	4	1392	1042	693	28,1	136/260
104	104	24×24	16	96×96	816	336	136	56	6	1632	1222	813	29,2	168/318
120	120	18×18	36	108×108	1050	408	175	68	6	2100	1573	1047	28	204/390
132	132	20×20	36	120×120	1304	496	163	62	8	2608	1954	1301	27,6	248/472
144	144	22×22	36	132×132	1558	620	156	62	8[c]	3116	2335	1555	28,5	310/590
							155	62	2[c]					
Символы прямоугольной формы														
8	18	6×16	1	6×16	5	7	5	7	1	10	6	3	58,3	3/0
8	32	6×14	2	6×28	10	11	10	11	1	20	13	8	52,4	5/0
12	26	10×24	1	10×24	16	14	16	14	1	32	22	14	46,7	7/11
12	36	10×16	2	10×32	22	18	22	18	1	44	31	20	45,0	9/15
16	36	14×16	2	14×32	32	24	32	24	1	64	46	30	42,9	12/21
16	48	14×22	2	14×44	49	28	49	28	1	98	72	47	36,4	14/25

[a] В размер символа не включена свободная зона.

[b] По 5.7.3.

[c] В символе наибольшего размера (144×144) первые восемь блоков по Риду-Соломону включают в 218 кодовых слов, кодирующих 156 кодовых слов данных. Последние два блока включают 217 кодовых слов (155 слов данных). Все блоки содержат 62 кодовых слова исправления ошибок.

[d] В соответствии с правилами схем кодировании Text или C40 без использования знаков фиксации и регистра; при кодировании с помощью других схем это значение может значительно варьироваться в зависимости от сочетания и группирования наборов знаков.

5.5.2 Включение направляющих шаблонов в символы большого размера

Согласно таблице 7 символы квадратной формы, размерами (в модулях) 32 × 32 и более, и четыре прямоугольных символа размерами (в модулях) 8 × 32, 12 × 36, 16 × 36 и 16 × 48 имеют две или более области данных (data regions). Эти области данных ограничивают направляющими шаблонами (приложение D). Символы квадратной формы делят на 4, 16 или 36 областей данных (приложение D, рисунки D.1, D.2 и D.3). Прямоугольные символы делят на две области данных (приложение D, рисунок D.4). Чередующиеся темные модули направляющего шаблона должны быть расположены на верхней и правой границах области данных и идентифицировать четные столбцы и строки.

5.6 Структурированное соединение

5.6.1 Основные принципы

В структурированном формате может присутствовать до 16 символов версии ECC 200 для кодирования сообщения большого объема. Символ является частью структурированного соединения, что отмечает-

ся кодовым словом значением 233 в позиции первого знака символа. Непосредственно за ним следуют три кодовых слова структурированного соединения. Первое кодовое слово является индикатором позиции символа в группе, второе и третье предназначены для идентификация файла.

5.6.2 Индикатор позиции символа

Кодовое слово индикатора позиции символа задает положение конкретного символа внутри группы (до 16) символов версии ECC 200 в формате структурированного соединения в виде значения «*m* из общего числа *n* символов». Первые четыре бита в данном кодовом слове указывают позицию данного символа как двоичное значение равное (*m — 1*). Последние 4 бита задают общее число символов, подлежащих объединению в формате структурированного соединения как двоичное значение, равное (*17 — n*). 4-битовые комбинации должны соответствовать установленным в таблице 8.

Таблица 8 — Значения битов позиции символа в структурированном соединении

Позиция символа	Значения битов с 1 по 4	Общее число символов	Значения битов с 5 по 8	Позиция символа	Значения битов с 1 по 4	Общее число символов	Значения битов с 5 по 8
1	0000			9	1000	9	1000
2	0001	2	1111	10	1001	10	0111
3	0010	3	1110	11	1010	11	0110
4	0011	4	1101	12	1011	12	0101
5	0100	5	1100	13	1100	13	0100
6	0101	6	1011	14	1101	14	0011
7	0110	7	1010	15	1110	15	0010
8	0111	8	1001	16	1111	16	0001

Пример — Порядок кодирования кодового слова индикатора позиции символа для третьего символа в группе из семи символов:

Значение битов в третьей позиции символа: 0010

Общее число символов 7: 1010

Комбинация битов: 00101010

Значение кодового слова: 42

5.6.3 Идентификация файла

Идентификацию файла задают значениями двух кодовых слов. Каждое кодовое слово идентификации файла может иметь значение от 1 до 254, что допускает 64516 различных вариантов идентификации файла. Идентификация файла предназначена для повышения вероятности того, что только логически связанные символы обрабатываются как часть единого сообщения.

5.6.4 Структурированное соединение и знак FNC1

Если структурированное соединение используется в сочетании со знаком FNC1 (5.2.4.6), то первые четыре кодовых слова следует применять для структурированного соединения, а пятое и шестое доступны для использования знака FNC1. Знак FNC1 не должен повторяться в этих же позициях во втором и последующем символах, если только он не используется в качестве разделителя полей.

5.6.5 Буферизованные и небуферизованные операции

Сообщение, содержащееся в рамках последовательности структурированного соединения, может быть целиком накоплено в буфере устройства считывания до своего полного ввода и передано после того, как считаны все символы. В качестве альтернативы устройство считывания может передавать декодированные данные из каждого символа по мере их считывания. В этой небуферизованной операции протокол ECI для структурированного соединения, установленный в стандарте AIM ITS 04/001, часть 1, определяет управляющий блок, который должен вставляться в качестве префикса перед началом передаваемых данных каждого символа.

5.7 Обнаружение и исправление ошибок

5.7.1 Исправление ошибок Рида-Соломона

В символах версии ECC 200 используют исправление ошибок Рида-Соломона.

Для символов версии ECC 200 с общим числом кодовых слов менее 255 кодовые слова исправления ошибки вычисляют с помощью кодовых слов данных без процедуры чередования.

Для символов версии ECC 200 с общим числом кодовых слов более 255 кодовые слова исправления ошибки вычисляют с помощью кодовых слов данных с использованием процедуры чередования (приложение А). Каждый символ версии ECC 200 характеризуется особым числом кодовых слов данных и исправления ошибок, которые разделены в определенном числе блоков (таблица 7) и к которым применяется процедура чередования (приложение А).

Полиномиальные арифметические вычисления для символов версии ECC 200 должны проводиться с использованием побитового арифметического устройства для битовых операций по модулю 2 и арифметического устройства для байтовых операций по модулю 100101101 (десятичное значение 301). Это поле Галуа 2^8, где 100101101 соответствует простому минимальному многочлену поля $x^8+x^5+x^3+x^2+1$. Используют 16 различных порождающих многочленов для вычисления соответствующих кодовых слов исправления ошибок, приведеных в приложении Е, раздел Е.1.

5.7.2 Генерация кодовых слов исправления ошибок

Кодовые слова исправления ошибок являются остатком от деления кодовых слов данных на полиномиальную функцию g(x), используемую для кодов Рида-Соломона (приложение Е, раздел Е.1).

Примечание — При вычислении полином данных символа сначала должен быть дополнительно умножен на x^k, после чего проводят деление многочлена на многочлен.

Кодовые слова данных являются коэффициентами полинома с коэффициентом при наивысшей степени равным первому кодовому слову данных и с коэффициентом при низшей степени равным последнему кодовому слову данных перед первым кодовым словом исправления ошибок. Коэффициент при наивысшей степени оставшейся части полинома является первым кодовым словом исправления ошибок и коэффициент при нулевой степени является последним кодовым словом исправления ошибок и последним кодовым словом. Это может быть выполнено с помощью схемы деления, приведенной на рисунке 5. Регистры от b_0 до b_{k-1} инициализируют нулями. Существуют две стадии генерации кодирования. На первой стадии при положении ключа в нижней позиции кодовые слова данных передаются как на выход, так и на схему. Первая стадия завершается за *n* синхронизирующих импульсов. На второй стадии (*n* + 1, *n* + *k* синхронизирующих импульсов) при положении ключа в верхнем положении, кодовые слова исправления ошибок $\varepsilon_{k-1},...,\varepsilon_0$ генерируются путем выдачи значений из всех регистров по порядку с сохранением нулевых данных на входе. Кодовые слова на выходе регистра сдвига должны быть расположены в порядке, в котором они будут размещены в символе. При использовании процедуры чередования указанные кодовые слова не будут размещены последовательно в знаках символа (приложение А).

Примечание — *n* и *k* определены в 3.2 как число кодовых слов данных и число кодовых слов исправления ошибок соответственно.

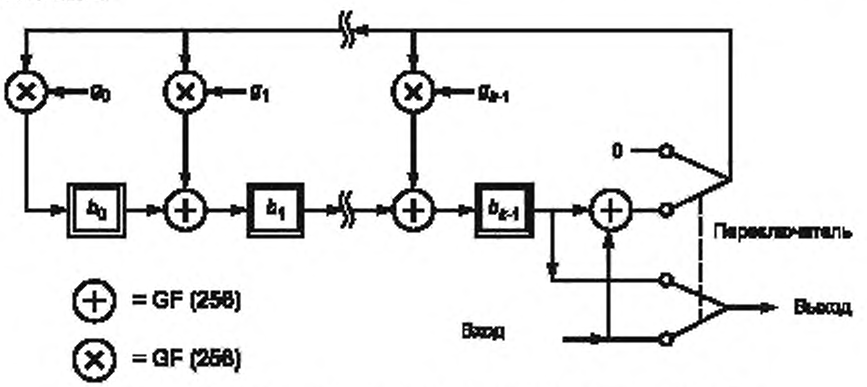


Рисунок 5 — Схема генерации кодовых слов исправления ошибки

5.7.3 Возможности исправления ошибок

Кодовые слова исправления ошибок позволяют исправлять два типа ошибочных кодовых слов: стирания (ошибочные кодовые слова с известными позициями) и ошибки (ошибочные кодовые слова с неизвестными позициями). Стирание представляет собой не сканированный или не подающийся декодирова-

нию знак символа. Ошибка представляет собой неправильно декодированный знак символа. Число стираний и ошибок, которые одновременно можно исправить, вычисляют по следующей формуле

$$e + 2t \le d - p,$$

где e — число стираний;

t — число ошибок;

d — число кодовых слов исправления ошибок;

p — число кодовых слов, зарезервированных для обнаружения ошибок.

В общем случае $p = 0$. Однако если большая часть возможности исправления ошибок, свойственных коду, используется для исправления стираний, то возрастает возможность необнаружения ошибки. Если число стираний больше половины числа кодовых слов исправления ошибок, то $p = 3$. Для символов небольших размеров (10×10, 12×12, 8×18, 8×32) не следует использовать исправление стираний ($e = 0$ и $p = 1$).

5.8 Формирование символа

При заданной последовательности кодовых слов, рассмотренной в предыдущих разделах, символ версии ECC 200 формируется следующим образом:

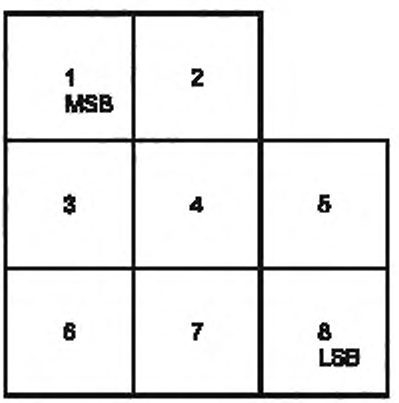
a) размещение модулей кодовых слов в координатной матрице;

b) подстановка модулей направляющего шаблона, при необходимости;

c) размещение модулей шаблона поиска по периметру символа.

5.8.1 Размещение знаков символа

Каждый знак символа должен включать в себя восемь номинально квадратных модулей, каждый из которых отображает один двоичный разряд. Темный модуль соответствует единице, светлый — нулю. Восемь модулей, упорядоченных слева-направо и сверху-вниз составляют форму знака символа (рисунок 6). Вследствие того, что форма знака символа (рисунок 6) не может точно вписаться в границы символа, некоторые знаки символа разбивают на части. Порядок размещения знаков символа определен программой на языке программирования С (приложение F).



LSB — младший значащий разряд (Least significant bit); MSB — старший значащий разряд (Most significant bit)

Рисунок 6 — Представление кодового слова в знаке символа версии ECC 200

5.8.2 Размещение модулей направляющего шаблона (Alinement Pattern)

Этот этап применяют только для крупных матриц квадратной формы размерами (в модулях) от 32 × 32 и более, а также прямоугольной формы размерами (в модулях) от 8 × 32, 12 × 36 и более. Для выбранного формата символа координатную матрицу разбивают на области данных размерами, установленными в таблице 7. Области данных отделяют друг от друга направляющими шаблонами шириной два модуля. В результате некоторые знаки символа будут разделены между двумя смежными областями данных. Для матриц квадратной формы направляющие шаблоны размещают между областями данных

горизонтально и вертикально парами. Общее число пар направляющих шаблонов 2, 6 или 10 (приложение D, рисунки D.1 — D.3). Для матриц прямоугольной формы между областями данных размещают только один вертикальный направляющий шаблон (приложение D, рисунок D.4).

5.8.3 Размещение модулей шаблона поиска (Finder Pattern)

Для формирования шаблона поиска модули следует размещать по периметру матрицы (4.3.1).

6 Требования к символам версии ECC 000-140

6.1 Рекомендации по применению

Для разрабатываемых прикладных приложений и открытых систем рекомендуется использовать символику версии ECC 200 (раздел 5). Неизвестны случаи, когда символы версии ECC 200 были бы менее устойчивыми к повреждениям, чем символы версии ECC 000-140 того же размера.

6.2 Порядок кодирования

В настоящем подразделе приведено общее описание порядка кодирования, в следующих подразделах — более подробное описание. Пример кодирования для символа уровня ECC 050 приведен в приложении Q.

Преобразование данных пользователя в символ версии ECC 000-140 проводят в следующем порядке:

Этап 1. Кодирование данных

Проводят анализ входных данных пользователя для определения совокупности различных типов знаков, подлежащих кодированию. Для максимальной эффективности уплотнения должна быть выбрана оптимальная схема кодирования самого низкого уровня, способная закодировать данные. Если пользователь не устанавливает размер матрицы, то выбирают наименьший размер для размещения данных. Результатом этого этапа является двоичный поток закодированных данных (Encoded Data Bit Stream).

Этап 2. Формирование префикса данных

Двоичный поток префикса данных формируют из поля идентификатора формата, поля величины контроля циклической избыточности (CRC) и двоичного поля длины данных. Указанный двоичный поток префикса данных добавляют в виде префикса к двоичному потоку закодированных данных для создания незащищенного двоичного потока (Unprotected Data Bit Stream).

Этап 3. Обнаружение и исправление ошибок

Обрабатывают незащищенный двоичный поток в соответствии с установленным пользователем алгоритмом сверточного кодирования для создания защищенного двоичного потока (Protected Bit Stream). Этот этап не выполняют для символов уровня ECC 000.

Этап 4. Построение головной и конечной меток

К защищенному двоичному потоку добавляют в виде префикса головную метку, содержащую только двоичное поле контроля и исправления ошибок (ECC). К защищенному двоичному потоку присоединяют также конечную метку, содержащую биты-заполнители (нулевые биты). Защищенный двоичный поток, к которому добавлены головная и конечная метки, называют нерандомизированным двоичным потоком (Unrandomised Bit Stream).

Этап 5. Шаблонная рандомизация

Нерандомизированный двоичный поток обрабатывают с помощью алгоритма рандомизации и создают рандомизированный двоичный поток (Randomised Bit Stream).

Этап 6. Размещение модулей в матрице

Модули размещают в матрице так, чтобы образовался шаблон поиска. Рандомизированный двоичный поток размещают в матрице помодульно в соответствии с алгоритмом размещения модулей данных (приложение H). Различные двоичные потоки в процессе кодирования приведены на рисунке 7.

6.3 Кодирование данных

Данные должны быть закодированы с использованием одной из шести схем кодирования (таблица 9). Схему кодирования устанавливают для всего символа, поэтому выбор наиболее подходящей схемы кодирования может значительно влиять на число битов, необходимых для кодирования исходных данных. Одни и те же данные могут быть представлены в символах версии ECC 000-140 различными способами путем использования различных схем кодирования. Наборы знаков для всех схем кодирования, за исключением схемы с 8-битовыми байтами, приведены в приложении I. Схему кодирования с 8-битовыми байта-

ми определяет пользователь. Наиболее эффективной схемой кодирования для использования является схема с наименьшим основанием, с помощью которой можно закодировать все знаки сообщения. Таким образом, если все знаки могут быть закодированы по схеме кодирования по основанию 27, неэффективно использовать схемы кодирования по основаниям 37 и 41 или схему кодирования ASCII (КОИ-7/КОИ-8).

Чтобы определить наиболее подходящую схему кодирования, необходимо проанализировать данные, предназначенные для кодирования. Наборы знаков каждой из схем кодирования с основанием N необходимо последовательно сравнивать с набором знаков, подлежащих кодированию, начиная с набора

Таблица 9 — Схемы кодирования для символов версии ECC 000-140

Схема кодирования	Знаки	Число битов на один знак
По основанию 11	Цифровые данные	3,5
По основанию 27	Прописные буквы	4,8
По основанию 37	Прописные буквы и цифры	5,25
По основанию 41	Прописные буквы, цифры и специальные графические знаки	5,5
ASCII (КОИ-7)	Полный набор 128 знаков ASCII (КОИ-7)	7
8-битовый байт	Определяются пользователем	8

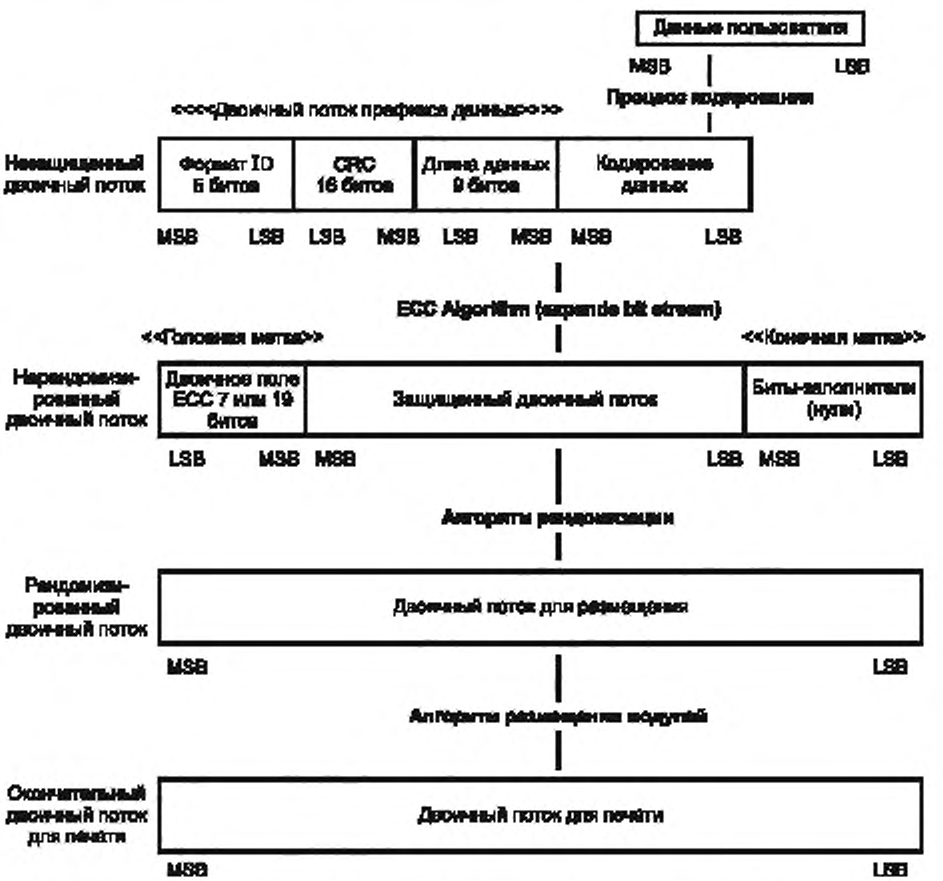


Рисунок 7 — Двоичные потоки в процессе кодирования символов версии ECC 000-140

знаков схемы кодирования по основанию 11. Если данный набор знаков подходит, то следует использовать именно его, если нет, то сравнение должно быть продолжено для наборов знаков схем кодирования по основаниям 27, 37 и 41 до тех пор, пока не будет определена подходящая схема с наименьшим числом в основании. Если знаки данных выходят за пределы возможностей набора знаков схемы кодирования по основанию 41, то необходимо использовать набор знаков ASCII (КОИ-7), пока знаки не выходят за установленные пределы, в противном случае следует использовать набор 8-битовых байтов.

Для всех схем кодирования каждую уплотненную последовательность от 4 до 24 битов длиной размещают в двоичном потоке закодированных данных в обратном порядке, начиная с самого младшего двоичного разряда в первой позиции, т.е. формируют каждую отдельную уплотненную последовательность, затем порядок меняют на обратный и добавляют в двоичный поток закодированных данных, т.е. формируют полный уплотненный двоичный поток, который затем (как целое) меняет порядок на обратный.

Подробности каждой схемы кодирования приведены в следующих подпунктах.

6.3.1 Схема кодирования по основанию 11 — кодирование цифр

В схеме кодирования по основанию 11 (цифры) кодируют шесть знаков данных как 21 бит с плотностью кодирования 3,5 битов на один знак данных. Кодируемый набор знаков схемы кодирования по основанию 11 позволяет кодировать следующие 11 знаков:

- цифры от 0 до 9;
- знак ПРОБЕЛ.

Данные кодируют в два этапа. На первом этапе фактические знаки данных должны быть заменены их кодовыми значениями по основанию 11 в соответствии с приложением I. На втором этапе должно быть произведено уплотнение кодовых значений по основанию 11 путем преобразования от схемы кодирования по основанию 11 к схеме кодирования по основанию 2 (приложение I, раздел I.1).

6.3.2 Схема кодирования по основанию 27 — кодирование прописных букв

В схеме кодирования по основанию 27 (прописные буквы) кодируют пять знаков данных в 24 битах с плотностью кодирования 4,8 битов на один знак данных. Кодируемый набор знаков по основанию 27 позволяет кодировать следующие 27 знаков:

- прописные латинские буквы от A до Z;
- знак ПРОБЕЛ.

Кодирование данных проводят в два этапа. На первом этапе фактические знаки данных должны быть заменены их кодовыми значениями по основанию 27 (приложение I). На втором этапе должно быть произведено уплотнение кодовых значений по основанию 27 путем преобразования от схемы кодирования по основанию 27 к схеме кодирования по основанию 2 (приложение I, раздел I.2).

6.3.3 Схема кодирования по основанию 37 — кодирование прописных букв и цифр

В схеме кодирования по основанию 37 (прописные буквы и цифры) кодируют четыре знака данных в 21 бите с плотностью кодирования 5,25 битов на один знак данных. Кодируемый набор знаков по основанию 37 позволяет кодировать следующие 37 знаков:

- прописные латинские буквы от A до Z;
- цифры от 0 до 9;
- знак ПРОБЕЛ.

Данные кодируют в два этапа. На первом этапе фактические знаки данных должны быть заменены их кодовыми значениями по основанию 37 (приложение I). На втором этапе должно быть произведено уплотнение кодовых значений по основанию 37 путем преобразования от схемы кодирования по основанию 37 к схеме кодирования по основанию 2 (приложение I, раздел I.3).

6.3.4 Схема кодирования по основанию 41 — кодирование прописных букв, цифр и специальных графических знаков

В схеме кодирования по основанию 41 (прописные буквы, цифры и специальные графические знаки) кодируют четыре знака данных в 22 битах с плотностью кодирования 5,5 битов на один знак данных. Кодируемый набор знаков по основанию 41 позволяет кодировать следующий 41 знак:

- прописные латинские буквы от A до Z;
- цифры от 0 до 9;
- знак ПРОБЕЛ;
. (ТОЧКА);
, (ЗАПЯТАЯ);
- (МИНУС или ДЕФИС);
/ (ДРОБНАЯ ЧЕРТА).

Данные кодируют в два этапа. На первом этапе фактические знаки данных должны быть заменены их кодовыми значениями по основанию 41 (приложение I). На втором этапе должно быть произведено уплотнение кодовых значений по основанию 41 путем преобразования от схемы кодирования по основанию 41 к схеме кодирования по основанию 2 (приложение I, раздел I.4).

6.3.5 Схема кодирования ASCII (КОИ-7)

В схеме кодирования ASCII (КОИ-7) кодируют все 128 знаков по ИСО/МЭК 646[1)]. Каждый знак данных кодируют как 7-битовый байт, эквивалентный десятичному значению, приведенному в приложении I, таблице I.1, графе ASCII (КОИ-7).

6.3.6 Схема кодирования 8-битовыми байтами

Схему кодирования 8-битовыми байтами используют в замкнутых прикладных системах, где интерпретацию данных определяет пользователь. Каждый знак данных должен кодироваться как 8-битовый байт.

6.4 Выбор пользователем уровня исправления ошибок

6.4.1 Выбор уровня исправления ошибок

Символы версии ECC 000-140 предлагают пять уровней исправления ошибок на основе использования сверточных кодов (таблица 10). В прикладных системах применения важно понимать, что повышение уровня исправления ошибок приводит к пропорциональному увеличению числа битов в выходном сообщении (и, следовательно, к увеличению размера символа) и обеспечивает различные уровни исправления ошибок.

Таблица 10 — Уровень исправления ошибок, уровень повреждений и увеличение числа битов

Уровень кода исправления ошибок	Максимально возможный уровень повреждений, %	Увеличение числа битов пользователя к ECC 000, %
000	Отсутствует	Отсутствует
050	2,8	33
080	5,5	50
100	12,6	100
140	25	300

6.4.2 Прочие уровни исправления ошибок на основе алгоритмов сверточных кодов

До публикации настоящего стандарта в Data Matrix в прикладных применениях использовались иные уровни исправления ошибок, основанные на алгоритмах сверточного кодирования. Информация нетиповых уровнях исправления ошибок доступна в AIM Inc. Символы с подобными уровнями исправления ошибок не соответствуют требованиям настоящего стандарта.

6.5 Формирование незащищенного двоичного потока

На рисунке 7 показано, что незащищенный двоичный поток имеет префикс данных двоичного потока в виде префикса к кодированным битам данных. Определения составляющих частей префикса данных двоичного потока приведены в следующих пунктах.

6.5.1 Двоичное поле идентификатора формата

Идентификатор формата определяет схему кодирования данных. Идентификатор формата имеет десятичное значение для целей определения и 5-битовую величину сегмента для кодирования (таблица 11).

[1)] *Набор знаков по ANSI INCITS 4-1986 (R2007) Information Systems - Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII) (Информационные системы — Кодированные наборы знаков — 7-битовый американский национальный стандартный код для обмена информацией (7- битовый ASCII).*

Таблица 11 — Кодирование идентификатора формата

Идентификатор формата	Схема кодирования	Значение 5-битового сегмента
		MSB LSB
1	По основанию 11	00000
2	По основанию 27	00001
3	По основанию 41	00010
4	По основанию 37	00011
5	ASCII (КОИ-7)	00100
6	8-битовых байтов	00101

6.5.2 Двоичное поле контроля циклической избыточности (CRC)

Двоичное поле контроля циклической избыточности (CRC) генерируют с помощью алгоритма CRC. Значение CRC генерируется перед началом кодирования из исходных данных пользователя, представленных в виде 8-битовых байтов, и поэтому используется для независимого контроля ошибок данных пользователя. Полное описание процедуры генерирования значения CRC приведено в приложении J.

6.5.3 Двоичное поле длины данных

Двоичное поле длины данных имеет размер 9 битов и представляет в двоичном виде число кодируемых знаков данных пользователя.

6.5.4 Построение префикса данных

Двоичный поток префикса данных имеет длину 30 битов и формируется в соответствии с рисунком 8.

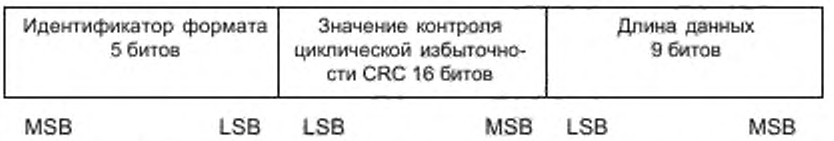


Рисунок 8 — Структура префикса двоичного потока данных

Примечание — Некоторые двоичные поля начинаются с самого старшего разряда (MSB), другие — с самого младшего разряда (LSB).

6.5.5 Завершение незащищенного двоичного потока данных

Кодированные двоичные данные добавляют в качестве суффикса к двоичному потоку префикса данных для формирования незащищенного двоичного потока данных.

6.6 Построение нерандомизированного двоичного потока

Нерандомизированный двоичный поток состоит из трех составных частей (рисунок 7):

a) головной метки;

b) защищенного двоичного потока;

c) конечной метки.

Формирование указанных составных частей проводят в соответствии с требованиями, приведенными в следующих пунктах.

6.6.1 Построение головной метки

Головная метка нерандомизированного двоичного потока содержит двоичное поле кода исправления ошибок (ECC), которое определяет структуру сверточного кода, используемого для защиты данных, закодированных в символе. Двоичное поле кода исправления ошибок (ECC) имеет в длину 7 или 19 битов (таблица 12).

Таблица 12 — Двоичное поле кода исправления ошибок (ECC)

Уровень кода исправления ошибок	Идентификатор двоичного сегмента MSB LSB
000	1111110
050	0001110000000001110
080	1110001110000001110
100	1111111110000001110
140	1111110001110001110

6.6.2 Применение сверточного кодирования для создания защищенного двоичного потока

Необходимо выбрать один из пяти уровней исправления ошибок. Критерии выбора приведены в 6.4. Для символа уровня ECC 000 не применяют ни один из уровней исправления ошибок, поэтому незащищенный двоичный поток автоматически становится защищенным двоичным потоком. Для прочих четырех уровней исправления ошибок применяют сверточное кодирование, что приводит к увеличению места, занимаемого входными данными, пропорционально по всей длине. Обработка незащищенного двоичного потока на соответствующем конечном автомате исправления ошибок и считывание результата должны создавать кодированный двоичный поток. Схемы четырех конечных автоматов для символов версии ECC 050-140 приведены в приложении К.

6.6.3 Построение конечной метки

К защищенному двоичному потоку добавляют конечную метку, содержащую биты-заполнители (нули). Биты-заполнители должны быть добавлены в конец двоичного потока для обеспечения того, чтобы квадратный корень общего числа битов в нерандомизированном двоичном потоке был равен нечетному целому числу в интервале от 7 до 47. Эта процедура обеспечивает квадратную форму символа.

6.6.4 Построение нерандомизированного двоичного потока

Защищенный двоичный поток с головной и конечной метками называют нерандомизированным двоичным потоком (рисунок 7).

6.7 Шаблонная рандомизация

Нерандомизированный двоичный поток обрабатывают с применением алгоритма шаблонной рандомизации, в результате чего возникает рандомизированный двоичный поток. Алгоритм шаблонной рандомизации включает в себя операцию поразрядного исключающего ИЛИ (XOR), приложенную к нерандомизированному двоичному потоку по всей его длине, и образцу рандомизации двоичного потока (приложение L), начиная с позиции самого старшего двоичного разряда (MSB).

6.8 Размещение модулей в матрице

Размер сторон сетки модулей данных должен быть равен нечетному числу (от 7 до 47), вычисляемому как квадратный корень согласно 6.6.3. Рандомизированный двоичный поток размещают в матрице модуль за модулем в соответствии с сеткой размещения модулей данных (приложение Н). Шаблон поиска (4.3.1) должен быть размещен так, чтобы образовывать внешние границы сетки модулей данных.

7 Размеры символов

7.1 Размеры

Размеры символов Data Matrix должны соответствовать следующим требованиям:

размер X — размер модуля должен быть установлен в стандарте по применению с учетом методов сканирования и технологии нанесения символа;

шаблон поиска — ширина шаблона поиска должна быть равна размеру X,

направляющий шаблон — ширина направляющего шаблона должна быть равна 2X,

свободная зона — минимальный размер свободной зоны вокруг символа Data Matrix должен быть равен размеру X со всех четырех сторон. Для прикладных систем, в которых необходимо уменьшить влияние областей, расположенных в непосредственной близости от символа, создающих повышенные отражающие помехи, рекомендуется оставлять свободную зону размером от 2X до 4X.

8 Качество печати символов

Качество печати символов Data Matrix должно оцениваться в соответствии с требованиями, установленными в ИСО/МЭК 15415 с учетом дополнений и изменений, приведенных в настоящем разделе.

Некоторые методы маркировки не позволяют наносить символы, полностью соответствующие требованиям настоящего стандарта, без применения специальных мер. Дополнительные рекомендации по адаптации любой системы печати для производства требуемых символов Data Matrix приведены в приложении Т.

8.1 Параметры качества символа

8.1.1 Повреждение фиксированного шаблона

Методы измерения и оценки параметра повреждения фиксированного шаблона установлены в приложении М.

Примечание — Согласно ИСО/МЭК 15415 (приложение А) измерения и величины, установленные в приложении М настоящего стандарта, приведены взамен установленным в ИСО/МЭК 15415 (приложение А).

8.1.2 Качество сканирования и полный класс символа

Класс качества сканирования определяют как наименьшую величину из классов отдельных параметров — контраста, модуляции, повреждения фиксированного шаблона, декодирования, осевой неоднородности, неоднородности сетки модулей и неиспользованного исправления ошибок для отдельного изображения символа. Полный класс символа вычисляют как среднеарифметическое значение отдельных классов качества сканирования для нескольких тестовых изображений символа.

8.1.3 Неоднородность сетки

Идеальную сетку модулей рассчитывают с использованием четырех угловых точек реальной полученной сетки для каждой области данных и разделением ее на одинаковые ячейки по обеим осям.

8.1.4 Декодирование

Для получения класса декодирования следует использовать рекомендуемый алгоритм декодирования (раздел 9). Неспособность рекомендуемого алгоритма декодирования успешно декодировать символ оценивают как класс декодирования 0.

8.2 Измерения в процессе контроля

Для проведения необходимых измерений с целью мониторинга и контроля процесса производства символов Data Matrix используют средства и методы, описанные в приложении R. Указанные методы не позволяют точно оценить качество печати нанесенных символов (дополнительные рекомендации приведены выше в этом пункте, а также в приложении М), но использование каждого метода по отдельности и всех вместе может быть рекомендовано для производства надлежащих символов в процессе их печати.

9 Рекомендуемый алгоритм декодирования для символики Data Matrix

Данный рекомендуемый алгоритм декодирования[1)] позволяет находить на изображении символы Data Matrix и производить их декодирование следующим образом:

a) определяют значения параметров размеров и формируют цифровое изображение:

1) задают расстояние d_{min}, равное 7,5 диаметрам апертуры, заданной в системе применения, которое считают минимальной длиной стороны «L-образного» шаблона поиска;

2) задают расстояние g_{max}, равное 7,5 диаметров апертуры. Это расстояние считают максимальным промежутком в «L-образном» шаблоне поиска, допускаемым алгоритмом поиска на этапе b);

3) задают расстояние m_{min}, равное 1,25 диаметров апертуры, которое считают номинальным наименьшим размером модуля;

4) формируют черно-белое изображение, используя порог, определенный по методике, установленной в ИСО/МЭК 15415;

b) осуществляют поиск горизонтальной и вертикальной линии сканирования для двух внешних «L-образных» границ Data Matrix:

1) продлевают горизонтальную линию сканирования по обе стороны от центральной точки изображения, и, следуя вдоль этой линии, находят все точки перехода черное/белое и белое/черное. Для каждой точки перехода, найденной на линии сканирования и приведенной к границе пикселя (называемой далее точкой старта), производят следующее:

i) следуют от точки старта вверх вдоль границы перехода черный-белый, пиксель за пикселем до точки, расположенной на расстоянии 3,5 m_{min} от точки старта, или до той точки, в которой линия границы поворачивает вниз;

ii) следуют от точки старта вниз вдоль границы перехода черный-белый, пиксель за пикселем до точки, расположенной на расстоянии 3,5 m_{min} от точки старта, или до той точки, в которой линия границы поворачивает вверх;

iii) если при движении вверх достигли точки, отстоящей на 3,5 m_{min} от точки старта:

I) проводят линию А, соединяющую конечные точки вертикальной границы перехода;

II) проверяют, чтобы отклонение промежуточной точки границы от прямой линии А находилось в пределах 0,5 m_{min} и данная точка границы перехода находилась дальше от точки старта, чем предыдущая точка границы перехода. Если это условие выполняется, то продолжают выполнение этапа, указанного в перечислении iii). В противном случае переходят к выполнению этапа, указанного в перечислении 1) IV), до достижения края границы перехода в противоположное направление;

[1)] *Разработаны и иные алгоритмы декодирования с аналогичными функциями.*

III) продолжают следовать вдоль границы перехода вверх до расстояния 0,5 m_{min} от линии А. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии, большем или равном m_{min} от последней точки границы перехода вдоль линии границы перехода и сохраняют ее как конечную точку границы перехода. Эту точку следует рассматривать как одно из предполагаемых положений границ внешнего края L-образных границ;

IV) продолжают следовать вниз вдоль границы перехода до расстояния 0,5 m_{min} от линии А. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии, большем или равным m_{min} от последней точки границы перехода, и сохраняют ее как конечную точку границы перехода. Эта точка должна быть расположена на предполагаемой границе перехода, и ее следует рассматривать как одно из предполагаемых положений границы внешнего края L-образных границ;

V) вычисляют новую откорректированную линию А1, которая является «наиболее приближенной» линией для границы перехода, определенной на двух предыдущих этапах. «Наиболее приближенную» линию вычисляют с использованием алгоритма линейной регрессии (используя конечные точки для выбора зависимой оси, т.е. если они ближе к горизонтальной оси, зависимая ось — ось x) для каждой точки. На «наиболее приближенной» прямой линии отмечают отрезок, ограниченный точками p1 и p2 , которые являются ближайшими к найденным выше конечным точкам границы перехода;

VI) сохраняют две конечные точки отрезка линии А1 — p1 и p2. Так же сохраняют значение цвета левой стороны края границы перехода, видимое при движении от p1 к p2;

iv) если этап, указанный в перечислении iii), закончился неудачей или невозможно продолжить движение вниз на 3,5 m_{min} на этапе, указанном в перечислении iii) IV), проверяют, достигнута ли снизу граница перехода на расстоянии 3,5 m_{min} от точки старта. Если да, повторяют операции этапа, указанные в перечислении iii) , но не вверх, а вниз;

v) если этапы, указанные в перечислениях iii) и iv), закончились неудачей, проверяют, находятся ли верхняя и нижняя границы перехода на расстоянии не менее 2 m_{min} от точки старта. Если достигнуты верхняя и нижняя границы перехода, то включают в формируемую границу перехода сегменты вверх и вниз на расстоянии 2 m_{min} и повторяют операции этапа, указанные в перечислении iii), но с добавлением границы перехода;

vi) повторяют вышеуказанный процесс для следующей точки перехода на линии сканирования, начиная с этапа, указанного в перечислении i), до достижении края изображения;

2) продлевают линию сканирования вертикально в обоих направлениях от центральной точки изображения. Находят линию сегментов с использованием той же логической процедуры, что и на этапе, указанном в перечислении 1, одновременно следуя от каждой границы перехода символа влево, а затем вправо;

3) среди сохраненных отрезков линий А1 осуществляют поиск пар отрезков, удовлетворяющих следующим четырем условиям:

i) ближайшие конечные точки двух отрезков должны находиться друг от друга на расстоянии не более g_{max};

ii) два вышеуказанных отрезка должны быть параллельными с отклонением не более 5°;

iii) два вышеуказанных отрезка должны быть одного цвета, если отрезки имеют одинаковое направление от p1 к p2, или противоположного цвета, если направления этих отрезков противоположны;

iv) формируют две временные линии, продолжая каждый из двух рассматриваемых отрезков по достижению на их продолжении точки, ближайшей к конечной точке отрезка другой линии. Проверяют, чтобы обе временные линии были отделены менее чем на 0,5m_{min} от любой иной точки каждой линии;

4) для каждой пары линий, соответствующих требованиям этапа, указанного в перечислении 3), заменяют эту пару отрезков линий на один удлиненный отрезок линии А1 путем выбора «наиболее приближенной» линии по четырем конечным точкам пары рассматриваемых коротких отрезков линий. Также запоминают значение цвета левой стороны границы перехода новой удлиненной линии, рассматриваемой от конечной точки p1 до конечной точки p2;

5) повторяют этапы, указанные в перечислениях 3) и 4), до тех пор, пока возможно комбинировать пары линии А1:

6) выбирают отрезки линии, длиннее d_{min}. Помечают эти линии как предполагаемые L-образные стороны;

7) находят среди полученных пар предполагаемых L-образных сторон две линии, которые должны соответствовать следующим трем критериям:

i) ближайшие точки этих линий должны находиться друг от друга на расстоянии менее 1,5g_{max};

ii) эти две линии должны были взаимно перпендикулярны с погрешностью до 5°;

iii) внутренняя сторона угла, образованного этими линиями, имела один и тот же цвет. Следует иметь ввиду, что если одна или обе линии простираются в обе стороны от точки их пересечения, то два или четыре образованных L-образных шаблона должны быть проверены на соответствие цвету и минимальной длине d_{min} для укороченной стороны или сторон, прежде чем они могут стать предполагаемыми L-образными границами;

8) для каждой пары линий — предполагаемых L-образных границ, найденных на этапе, указанном в перечислении 7), формируют предполагаемую L-образную структуру путем продления отрезков до точки их пересечения;

9) если предполагаемая L-образная структура была сформирована из отрезков линий белого цвета внутри угла L-образной структуры, формируют инвертированное по цвету изображение для декодирования. Предпринимают попытки декодировать символ, начиная с нормального или инвертированного изображения, выбирая в качестве начального этап, указанный в перечислении d), используя каждую предполагаемую L-образную структуру, определенную на этапе, указанном в перечислении 8) как L-образный шаблон поиска. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении c);

c) продолжают подбирать отрезки линий A1 и предполагаемые L-образные структуры аналогично предыдущим этапам, также продолжают поиски предполагаемых L-образных структур, используя горизонтальное и вертикальное смещение линий сканирования по отношению к предыдущим линиям сканирования:

1) используя новую горизонтальную линию сканирования, проведенную на расстоянии $3m_{min}$ выше от центральной горизонтальной линии, повторяют в том же порядке действия этапа, указанного в перечислении b) 1), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислениях от b)3) до b)9). Если декодирование не удалось выполнить, переходят к следующему этапу, указанному в перечислении 2);

2) используя новую вертикальную линию сканирования, проведенную на расстоянии $3m_{min}$ слева от центральной вертикальной линии сканирования, повторяют действия этапа, указанного в перечислении b) 2), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислениях от b) 3) до b) 9). Если декодирование не удалось выполнить, переходят к следующему этапу, указанному в перечислении 3);

3) повторяют действия этапа, указанного в перечислении 1), используя новой горизонтальную линию сканирования, расположенную на расстоянии $3m_{min}$ ниже центральной горизонтальной линии сканирования. Если декодирование не удалось выполнить, повторяют действия этапа, указанного в перечислении 2), но со сдвигом новой вертикальной линии сканирования на $3m_{min}$ вправо от центральной вертикальной линии сканирования. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении 4);

4) продолжают производить горизонтальные и вертикальные линии сканирования, как это предусмотрено на этапах, указанных в перечислениях 1) — 3), на $3m_{min}$ вверх, затем влево, затем вниз, затем от ранее произведенных линий сканирования до успешного декодирования символа или до достижения края изображения;

d) первоначально считают, что область-кандидат содержит символ квадратной формы. Если декодировать область как символ квадратной формы не удается, пытаются найти и декодировать символ прямоугольной формы, начиная с этапа, указанного в перечислении j). Для символа квадратной формы сначала формируют нормализованную схему переходов для равных сторон области-кандидата, чтобы найти шаблон поиска с чередующимися модулями:

1) проецируют через область-кандидат линию, делящую пополам внутренний угол, образованный L-образными сторонами, определенный на этапе, указанном в перечислении b) 5) (рисунок 9). Определяют две равные области, образованные этой разделительной линией (биссектрисой), как левую и правую области со стороны угла L;

2) для каждой стороны формируют так называемую «линию поиска», расположенную на расстоянии d_{min} от вершины угла L-образной структуры и параллельную другой ее стороне, и продолжают эту линию до биссектрисы согласно рисунку 9;

3) сдвигают каждую «линию поиска» от вершины угла L-образной структуры (рисунок 9), удлиняя каждую «линию поиска», чтобы они всегда начинались от стороны угла L-образной структуры и заканчивались на биссектрисе, сохраняя линии поиска параллельными противоположным сторонам угла L-образной структуры. Каждый раз, когда каждая сторона сдвигается на один пиксель изобра-

жения, указывают на схеме сумму числа переходов от черного к белому и от белого к черному, умноженную на длину наибольшей стороны L-образной структуры и разделенную на текущую длину линии поиска, измеренную между двумя граничными линиями:

T = (число переходов) × (максимальная длина стороны L-образной структуры)/(длина линии поиска).

Эта формула нормализует значение T, предупреждая его увеличение по мере увеличения длины линии поиска.

Продолжают вычислять значения T до тех пор, пока линия поиска не будет длиннее наибольшей оси предполагаемой области-кандидата символа на 50 %;

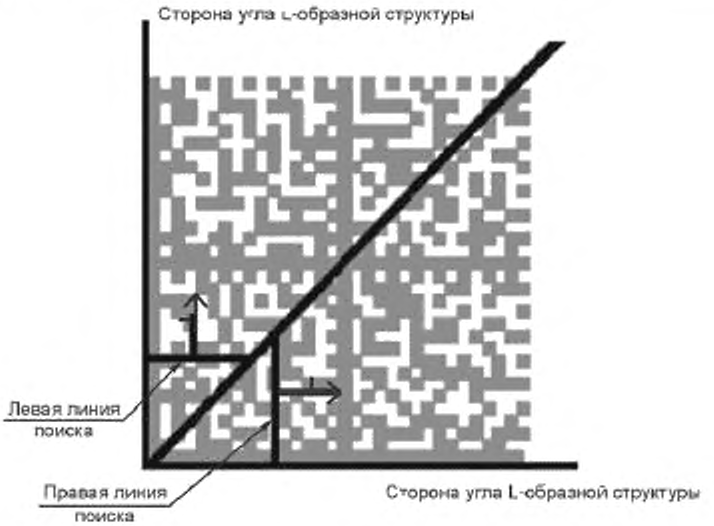


Рисунок 9 — Направления движения линий поиска

4) строят график зависимости T для каждой стороны (области), где на оси ординат Y указано значение T, а на оси абсцисс X — расстояние линии поиска от вершины угла L-образной структуры. Пример графика приведен на рисунке 10;

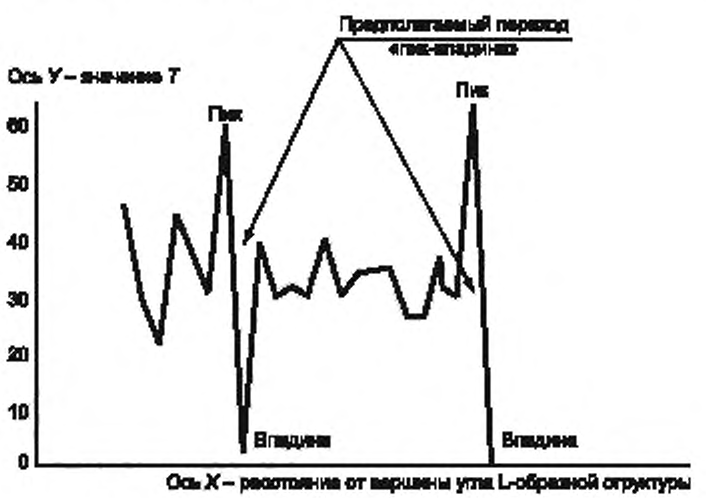


Рисунок 10 — Пример графика зависимости T от увеличения длины линии поиска

5) рассматривают график T с правой стороны (области), начиная с наименьших значений по оси X, с постепенным увеличением значений по этой оси. Находят первое место резкого падения значения на графике T, где значение T_S (T_S — максимальное значение одной из двух величин — ноль и T-1) во впадине меньше 15 % значения T в точке пика. Если пик или впадина на графике имеют плоский участок (плато), выбирают точку на пике или впадине, ближайшую к линии падения на графике. Линия поиска в точке пика может соответствовать стороне шаблона поиска с чередующимися модулями. Линия поиска во впадине может соответствовать внутренней части однородной темной линии или светлой свободной зоне;

6) находят пик и впадину на графике для левой стороны (области), которые в наибольшей степени подходят по координате X к пику и впадине на графике для правой стороны. Если значения X для пика или впадины на графике для левой стороны отличаются более чем на 15 % от соответствующих значений для пика или впадины на графике для правой стороны, то бракуют значения для пика и впадины на графике для правой стороны и продолжают поиск с этапа, указанного в перечислении d) 5), для следующего пика и впадины. Значение, равное 15 %, соответствует максимально разрешенному сокращению;

7) линия поиска, соответствующая впадине на графике для правой стороны, линия поиска, соответствующая впадине на графике для левой стороны, и две стороны угла L-образной структуры очерчивают возможную область данных символа Data Matrix. Проводят обработку этой области данных согласно этапу, указанному в перечислении e). Если декодирование выполнено неудачно, бракуют значения для пика и впадины на графике для правой стороны и продолжают поиск, начиная с этапа, указанного в перечислении d) 5), для следующего пика и впадины;

e) для каждой из двух сторон чередующегося шаблона находят линию, проходящую через центр чередующихся темных и светлых модулей:

1) для каждой стороны формируют прямоугольную область, ограниченную линиями поиска для пика и впадины как двумя длинными сторонами прямоугольной области и стороной L-образной структуры и линией для впадины для другой стороны как короткими сторонами прямоугольной области (рисунок 11);

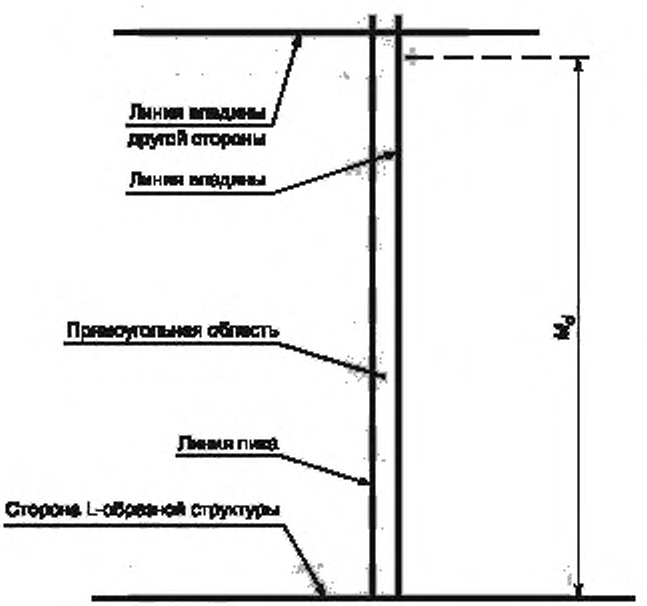


Рисунок 11 — Построение прямоугольной области

2) в пределах этой прямоугольной области, находят пары границ между пикселями на стороне с «зубцами»:

i) проводят контрольные линии, параллельные линии впадины, первая из которых совпадает с этой линией, и определяют все переходы цвета в направлении, перпендикулярном к контрольным линиям. Выбирают только переходы от темного к светлому или от светлого к темному, где первый цвет соответствует преобладающему цвету на изображении вдоль линии впадины;

ii) если число найденных цветовых переходов меньше 15 % числа пикселей, составляющих линию впадины, и контрольная линия не является линией пика, сдвигают контрольную линию в сторону линии пика приблизительно на один пиксель и повторяют действия этапа, указанного в перечислении i). Если выполнено условие наличия 15 % числа цветовых переходов к числу пикселей или достигнута линия пика, переходят к следующему этапу;

iii) вычисляют предварительную «наиболее приближенную линию» с помощью алгоритма линейной регрессии, используя точки на границах между выбранными парами пикселей;

iv) отбрасывают 25 % точек, наиболее отдаленных от предварительной «наиболее приближенной линии». Вычисляют окончательную «наиболее приближенную линию» с помощью алгоритма линейной регрессии с использованием оставшихся 75 % точек. Эта линия должна проходить вдоль внешней стороны чередующегося шаблона и указана на рисунке 12 как «наиболее приближенная линия»;

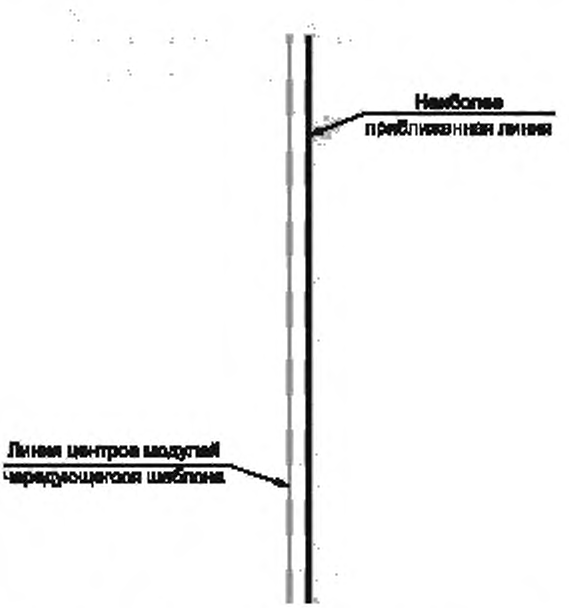


Рисунок 12 — Линия центров модулей чередующегося шаблона

3) для каждой стороны строят линию, параллельную линии, определенной на этапе, указанном в перечислении e)2), смещенную в сторону вершины угла L-образной структуры на длину «пиковой» линии поиска, разделенную на удвоенное число переходов на этой линии поиска:

Смещение = длина линии пика/(число переходов · 2).

Каждая из этих двух построенных линий должна соответствовать линии, проведенной через середины модулей шаблона чередующихся модулей (или средней линии шаблона чередующихся модулей) для этой стороны (рисунок 13).

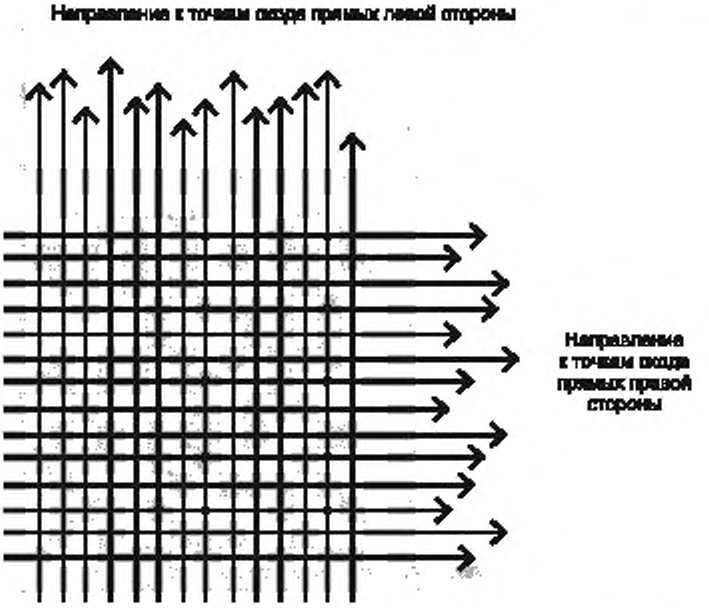


Рисунок 13 — Формирование реальной сетки модулей

f) для каждой стороны определяют число модулей данных на стороне символа квадратной формы или области данных:

1) ограничивают линию, проходящую через центры модулей чередующегося шаблона, сформированную на этапе, указанном в перечислении e) 3), с одной стороны границей L-образной структуры, и с другой стороны средней линией другого шаблона чередующихся модулей, определенной на этапе, указанном в перечислении e) 3). Длину этой линии обозначают M_d (рисунок 11);

2) вдоль ограниченной средней линии измеряют расстояния от края до края между всеми подобными границами всех двухэлементных пар, т.е. пар элементов темный/светлый и светлый/темный;

3) вычисляют среднеарифметическое значение всех измеренных расстояний от края до края и устанавливают текущую оценку расстояния от края до края *EE_Dist* как искомое значение;

4) бракуют все пары элементов, у которых измеренные расстояния от края до края отличаются более чем на 25 % от *EE_Dist*;

5) вычисляют среднеарифметическое значение оставшихся измеренных расстояний для данной стороны и обозначают его E_{avg};

6) вычисляют число модулей данных *dm* по формуле

$$dm = (Md \cdot 2 / E_{avg}) — 1{,}5,$$

где *dm* округляют до ближайшего целого числа;

7) если *dm* имеет разные значения для двух сторон, бракуют значения для пика и впадины для правой стороны и продолжают поиск с этапа, указанного в перечислении d) 5), для следующих пика и впадины. В противном случае *dm* считают искомым размером области данных квадратной формы;

g) для каждой стороны находят центральные точки модулей в шаблоне чередующихся модулей:

1) используя измеренные расстояния пар элементов, не забракованные на этапе, указанном в перечислении f) 4), вычисляют среднее приращение ширины темного элемента (штриха) при печати (по вертикали или горизонтали в зависимости от стороны сегмента) по формуле как среднее приращение ширины темного элемента (штриха) при печати для пар элементов (темный-светлый или штрих-пробел, в которой «штрих» — это ширина темнного элемента, а «пробел» — ширина светлого элемента в оставшейся паре элементов):

ink_spread = Average ((bar — ((bar + space)/2)) / ((bar + space)/2)),

где *ink_spread* — среднее приращение темного элемента (штриха) при печати;

Average — обозначение среднеарифметического значения;

bar — ширина темного элемента (штриха);

space — ширина светлого элемента (пробела);

2) вычисляют центр темного элемента (штриха) в паре элементов, занимающих среднее положение, используя следующее смещение в сторону темного элемента (штриха) от внешнего края темного элемента (штриха) в паре элементов, занимающих среднее положение:

$$\text{offset} = (EE_Dist \times (1 + ink_spread)) / 4,$$

где offset — смещение;

EE_Dist — среднеарифметическое значение измеренных расстояний от края до края;

ink_spread — среднее приращение темного элемента (штриха) при печати;

3) начиная от центра темного элемента (штриха) пары элементов, занимающих среднее положение, из этапа, указанного в перечислении f) 3), и продолжая процесс в направлении светлого элемента (пробела) из пары элементов, до конца ограниченной средней линии, вычисляют центр каждого элемента, выделенного белыми точками на темном фоне (рисунок 14) с выполнением следующих действий:

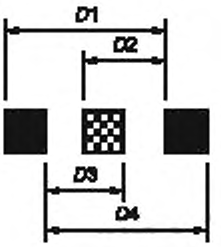


Рисунок 14 — Измерения расстояний от края до края для определения центра каждого элемента

Примечание — На рисунке 14 показаны три темных (штриха) и два светлых элемента (пробела). Если элемент, центр которого вычисляют, светлый (пробел), то на схеме должны быть представлены три светлых элемента (пробела) вместо темных (штрихов) и два темных элемента (штриха) вместо светлых (пробелов). Для светлых элементов (пробелов), смежных с конечными элементами средней линии, измерения расстояний *D1* или *D4* не проводят, поскольку они оказываются за границами символа или измеряемого сегмента.

i) вычисляют точку p1, находящуюся на средней линии на расстоянии *EE_Dist/2* от предыдущего вычисленного центра элемента в направлении нового элемента;

ii) вычисляют значения d_1 — d_4:

$d_1 = D1 / 2$,

$d_2 = D2$,

$d_3 = D3$,

$d_4 = D4 / 2$;

iii) если одно из значений d_1 — d_4 находится в пределах 25 % от *EE_Dist*, выбирают одно из значений d_i, ближайшее к *EE_Dist*, и устанавливают новое значение *EE_Dist* как среднее между текущим значением *EE_Dist* и выбранным d_i из диапазона d_1 — d_4:

I) если выбрано значение d_1 или d_4, определяют соответствующий край *D1* или *D4*, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние (*ink_spread/2*) (*EE_Dist/2*) в соответствующем направлении (т.е. если приращение ширины темного элемента (штриха) *ink_spread* положительная величина, смещение края должно быть в сторону светлого элемента (пробела), заключенного в пределах значений *D1* или *D4* , и, если отрицательная, смещение должно быть в противоположную сторону от светлого элемента). Вычисляют точку p2, находящуюся на средней линии на расстоянии 0,75 выбранного значения d_1 или d_4 от этого смещенного края в сторону элемента, центр которого должен быть вычислен;

II) если выбрано значение d_2 или d_3, определяют соответствующий край *D2* или *D3*, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние (*ink_spread/2*) (*EE_Dist/2*) в соответствующем направлении (т.е. если приращение ширины тем

ного элемента (штриха) при печати является положительным значением, смещение края должно быть проведено в сторону светлого элемента (пробела), заключенного в пределах значений *D*2 или *D*3, и, если отрицательная, смещение должно быть в противоположную сторону от светлого элемента). Вычисляют точку p2 , находящуюся на средней линии на расстоянии 0,25 выбранного значения d_2 или d_3 от смещенного края в сторону элемента, центр которого следует вычислить;

III) считают, что центр элемента находится точно посередине между точками p1 и p2;

iv) в противном случае, если ни одно из значений d_1, d_2, d_3, d_4 не находится в пределах 25 % *EE_Dist*, оставляют текущее значение *EE_Dist*, используют p1 как центр нового элемента и переходят к определению следующего элемента;

4) начиная с темного элемента в паре элементов, занимающих среднее положение, и продолжая в противоположном направлении по отношению к определенному на этапе, указанном в перечислении 3), вплоть до окончания ограниченной средней линии, вычисляют центры каждого элемента, используя порядок действий, установленный для этапа, указанного в перечислении для этапа 3);

h) составляют пробную сетку модулей данных в области данных, проводя линии из центров модулей шаблона чередующихся модулей:

1) для каждой стороны продолжают каждую линию, построенную на этапе, указанном в перечислении e) 3) и линию противоположной стороны L-образной структуры для формирования точки схода двух почти параллельных линий;

2) из каждой точки схода проводят лучи, проходящие через центры модулей, построенные на этапе, указанном в перечислении g, в направлении, близком к перпендикулярному к линии, полученной на этапе, указанном в перечислении e)3);

3) точки пересечения этих двух направлений лучей, близких к перпендикулярным, должны соответствовать центрам модулей данных в области данных (рисунок 13);

i) продолжают заполнение остальных областей данных:

1) в процессе составления области данных формируют новую L-образную структуру для следующей части данных левее или выше, используя одну из двух следующих процедур:

i) a. Если новая область данных по-прежнему ограничена с одной стороны исходной L-образной структурой, полученной на этапе, указанном в перечислении b), повторяют этап, указанный в перечислении c), устанавливая новую область данных и используя множество точек, выбранных на этапе, указанном в перечислении e) 2), и множество точек на стороне L-образной структуры из этапа, указанного в перечислении b)2), которые находятся за пределами линии, полученной на этапе, указанной в перечислении e)2);

ii) b. Если новая область данных ограничена с двух сторон другими областями данных, повторяют порядок действий с этапа, указанного в перечислении c), для определения новой области данных с помощью множества точек, выбранных на этапе, указанном в перечислении e)2), для каждой области данных, которая примыкает и ограничивает новую область данных с двух сторон;

2) если область данных не соответствует по числу модулей ранее полученным областям данных, символ корректируют путем его уменьшения до ближайшего большего числа областей, допускаемых стандартом для символа;

3) декодируют символ с одной или несколькими областями данных, начиная с последовательности действий, установленных на этапе, указанном в перечислении k);

j) находят части данных прямоугольного символа:

1) для каждой стороны L-образной структуры передвигают линию, перпендикулярную этой стороне, осуществляя сканирование по длине другой стороны L-образной структуры. После того как каждая сторона будет сдвинута на пиксель составляют график суммарного числа переходов от темного к светлому и от светлого к темному:

T = (число переходов) · (максимальная длина линии L-образной структуры) / (длина линии сканирования).

Сканирование продолжают до тех пор, пока параллельная линия сканирования не продвинется на длину перпендикулярной стороны L-образной структуры плюс 10 %;

2) по графику, начиная от его начала, для каждого направления находят первый случай резкого падения графика, где значение *T* во впадине менее 15 % значения *T* в точке пика. Если пик или впадина на графике имеют плоский участок (плато), выбирают точку на пике или во впадине, ближайшую к линии падения. Линия впадины в этой точке может представлять сторону символа или области данных;

3) находят линии шаблонов чередующихся модулей для каждой стороны области подобно процедуре для этапа, указанного в перечислении e);

4) составляют пробную сетку модулей для области данных или символа аналогично этапам, указанным в перечислениях f), g) и h). Пропускают этап, указанный в перечислении f) 6), который необходим только для символа квадратной формы;

5) если найденная область данных не является надлежащим символом прямоугольной формы, пытаются сформировать новую область данных, используя следующие далее действительные переходы пик/впадина;

6) составляют все дополнительные области данных как предусмотрено на этапе, указанном в перечислении i);

7) если установлена одна или несколько действительных областей данных, предпринимаются попытки декодировать символ в соответствии с перечислениями k) и l). Если область(ти) не является действительной, или декодирование не удалось выполнить, игнорируют эту область как предполагаемую действительную;

k) если число модулей данных является четным числом или символ имеет надлежащую прямоугольную форму, выполняют его декодирование, используя алгоритм исправления ошибок Рида-Соломона:

1) определяют модули данных в предполагаемых центрах сетки. Темный модуль соответствует единице, светлый модуль — нулю;

2) преобразуют группы по восемь модулей по определенным шаблонам кодовых слов в 8-битовые значения знаков символа;

3) выполняют процедуру исправления ошибок Рида-Соломона с полученными значениями знаков символа;

4) декодируют знаки символа в знаки данных в соответствии с установленными схемами декодирования;

l) если число модулей данных является нечетным числом, то декодируют символ, используя алгоритм сверточного исправления ошибок кода:

1) определяют модули данных в предполагаемых центрах сетки. Темный модуль соответствует единице, светлый модуль — нулю;

2) применяют черно-белую выравнивающую маску;

3) используя соответствующую таблицу расположения битов, преобразуют данные в двоичный поток;

4) затем применяют алгоритм сверточного исправления ошибок;

5) преобразуют битовый поток в знаки данных, используя соответствующую схему декодирования;

6) выполняют проверку правильности контрольной суммы CRC.

10 Рекомендации для пользователя

10.1 Интерпретация для визуального чтения

Поскольку символы Data Matrix могут содержать тысячи знаков, интерпретация для визуального чтения знаков данных в виде расположенного рядом обычного текста может оказаться неудобной. Как альтернатива, символ может сопровождать краткий описательный текст, а не весь закодированный. Размер знаков и тип шрифта не устанавливаются, и такое сообщение может быть напечатано в любом месте вблизи символа Data Matrix. Текст интерпретации для визуального чтения не должен накладываться ни на символ Data Matrix , ни на свободные зоны вокруг него.

10.2 Способность к автоматическому распознаванию

Символика Data Matrix может наряду с некоторыми другими символиками использоваться в среде автоматического распознавания (приложение S).

10.3 Системные подходы

Прикладные системы применения Data Matrix должны рассматриваться в качестве целостных системных решений (приложение T).

11 Передаваемые данные

Данный раздел описывает типовой протокол передачи данных для совместимых устройств считывания. Такие устройства считывания могут быть запрограммированы на поддержку иных вариантов передачи. Все закодированные знаки данных должны быть включены в передаваемые данные. Управляющие знаки символики и знаки исправления ошибок не передаются. Более сложные вопросы интерпретации данных рассмотрены далее в настоящем разделе.

11.1 Протокол для знака FNC1 (только для символики версии ECC200)

Когда знак FNC1 находится в позиции первого знака символа (или в позиции пятого знака символа в первом символе последовательности структурированного соединения), это является признаком соответствия данных типовому формату идентификаторов применения (AI) GS1. Знак FNC1 в любой более дальней позиции такого символа выполняет функцию разделителя полей. Данный протокол должен обеспечивать передачу идентификатора символики. Первый знак FNC1 не должен быть представлен в передаваемых данных, хотя на его присутствие указывает использование в идентификаторе символики знака модификации со значением 2 (11.5).

Когда знак FNC1 используют как разделитель полей, он должен быть представлен в передаваемом сообщении как управляющий знак $<^{G}{}_{S}>$ КОИ-7 (ASCII) с десятичным значением 29.

11.2 Протокол для знака FNC1 во второй позиции (только для символики версии ECC200)

Когда знак FNC1 находится в позиции второго знака символа (или в позиции шестого знака символа в первом символе последовательности структурированного соединения), это является признаком того, что данные соответствуют определенному международному отраслевому стандартному формату. Указанный протокол должен обеспечивать передачу идентификатора символики. Первый знак FNC1 не должен быть представлен в передаваемых данных, хотя на его присутствие указывает использование соответствующего знака-модификатора со значением 3 в идентификаторе символики (11.5). Данные, закодированные в первом знаке символа, должны передаваться как обычно в начале данных. Когда знак FNC1 используется как разделитель полей, он должен быть представлен в передаваемом сообщении как управляющий знак $<^{G}{}_{S}>$ ASCII (КОИ-7) с десятичным значением 29.

11.3 Протокол для знаков Macro в первой позиции (только для символики версии ECC 200)

Данный протокол используют для кодирования с уплотнением двух специальных головных и конечных меток сообщения в символах Data Matrix версии ECC 200.

Когда знак Macro находится в первой позиции, должны передаваться вводная и заключительная части сообщений. Если первый знак символа содержит кодовое слово со значением 236 (т.е. закодированный знак Macro 05), то кодируемым данным должна предшествовать вводная часть $[\,)^{R}{}_{S}05^{G}{}_{S}$, за которой следуют данные. Если первый знак символа содержит кодовое слово со значением 237 (т.е. закодированный знак Macro 06), то кодируемым данным должна предшествовать вводная часть $[\,)^{R}{}_{S}06^{G}{}_{S}$, за которой следуют данные. В обоих случаях после данных должна быть передана заключительная часть $^{R}{}_{S}\ ^{E}O_{T}$.

11.4 Протокол для интерпретаций в расширенном канале ECI (только для символики версии ECC200)

В системах, которые поддерживают интерпретации в расширенном канале (ECI), необходимо использовать префиксы идентификаторов символики при каждой передаче данных. Если присутствует кодовое слово ECI, его следует передавать как управляющий знак с десятичным значением 92_{DEC} (или с шестнадцатеричным значением $5C_{HEX}$), представляемый знаком «\» (ОБРАТНАЯ ДРОБНАЯ ЧЕРТА) в интерпретации, принимаемой по умолчанию. Следующие кодовые слова преобразуют в 6-разрядные значения в соответствии с правилами преобразования, обратными приведенным в таблице 6[1)]. Полученные 6-разрядные значения передают как знаки с соответствующими десятичными значениями ASCII (КОИ-7) (от 48 до 57). Прикладное программное обеспечение после распознавания последовательности \nnnnnn должно интерпретировать все последующие знаки как знаки интерпретации ECI, установленной в соответствии с указанным 6-разрядным значением. Эта интерпретация действует до окончания кодируемых данных или до появления другой последовательности ECI.

Если знак ОБРАТНАЯ ДРОБНАЯ ЧЕРТА (байт с десятичным значением 92_{DEC}) должен быть использован в кодируемых данных, то его передача должна осуществляться следующим образом. В случае, когда знак ASCII (КОИ-7 со значением 92_{DEC}) встречается как знак данных, должны быть переданы два

[1)] *Обратное преобразование — это перевод последовательности кодовых слов в номер назначения ECI.*

байта с этим же значением, таким образом в случае одиночного применения знак действует как управляющий знак, а появление сдвоенных знаков свидетельствует о появлении знака данных.

Примеры:

Закодированные данные A\\B\C.

Передаваемые данные A\\\\B\\C.

Использование идентификатора символики обеспечивает правильность интерпретации управляющего знака в данном прикладном применении.

11.5 Идентификатор символики

ИСО/МЭК 15424 предусматривает типовую процедуру указания символики, которая была считана, наряду с набором вариантов, установленных в декодере, и специальными свойствами, которые могут быть включены в символ.

После того как структура данных (включая использование любых ECI), идентифицирована, декодер должен добавить соответствующий идентификатор символики в виде префикса к передаваемым данным. Идентификатор символики также необходим в случае, если одна или несколько интерпретаций в расширенном канале (ECI) появляются в символе или при использовании знака FNC1 в соответствии с 11.1 и 11.2. Идентификаторы символики и значения возможных вариантов, которые возможны для применения в символике Data Matrix, приведены в приложении N.

11.6 Пример передаваемых данных

В данном примере сообщение, состоящее из двух знаков «¶Ж» должно быть закодировано в символе версии ECC 200 с помощью схемы кодирования ASCII (КОИ-7/КОИ-8). Знак «¶» представляют байтом с десятичным значением 182 в наборе знаков по умолчанию Data Matrix (номер назначения ECI 000003 соответствует набору знаков по ИСО 8859-1). Буква кириллицы «Ж» отсутствует в ECI 000003, но представлена в ИСО 8859-5[1)] (номер назначения ECI 000007) байтом с тем же десятичным значением 182. Полное сообщение, следовательно, может быть представлено, путем вставки переключения к ECI 000007 после первого знака следующим образом.

Символ кодирует сообщение

< ¶ > < переключение к ECI 000007 > < Ж >,

используя следующую последовательность кодовых слов Data Matrix:

[знак верхнего регистра] [55] [ECI] [8] [знак верхнего регистра] [55]

с десятичными значениями [235], [55], [241], [8], [235], [55].

Примечание 1 — Знак верхнего регистра с последующим кодовым словом, имеющим значение 55, кодирует байт с десятичным значением 182.

Примечание 2 — Номер назначения ECI в символе Data Matrix кодируют как ECI+1.

Декодер передает байты со следующими значениями (включая префикс идентификатора символики, вариант символики, знак-модификатор 4, указывающий на использование протокола ECI):

93, 100, 52, 182, 92, 48, 48, 48, 48, 48, 55, 182.

В графических знаках эта запись будет выглядеть следующим образом в интерпретации по умолчанию:

]d4¶\000007¶

Декодер отвечает за передачу сигнала переключения к ECI 000007, но не за интерпретацию результата. Программное обеспечение с поддержкой ECI в приемной прикладной системе удалит управляющую последовательность ECI 000007, и буква «Ж» будет представлена способом, принятым в системе (т.е. путем изменения шрифта в файле распечатки). Конечным результатом будет соответствие исходному сообщению с буквой «Ж».

[1)] *Набор 8-битовых знаков (версия КОИ-8) по ИСО/МЭК 8859-5 приведен в приложении V.*

Приложение А
(обязательное)

Процесс чередования в символике версии ECC 200

А.1 Пояснительная схема

Рассмотрим пример символа размером 72 × 72 модулей, для которого необходимо четыре уровня чередования, чтобы закодировать 368 кодовых слов данных и 144 кодовых слова исправления ошибок. Кодовые слова делят на четыре блока по 92 кодовых слова данных и 36 кодовых слов исправления ошибок с общей длиной каждого блока — 128 кодовых слов.

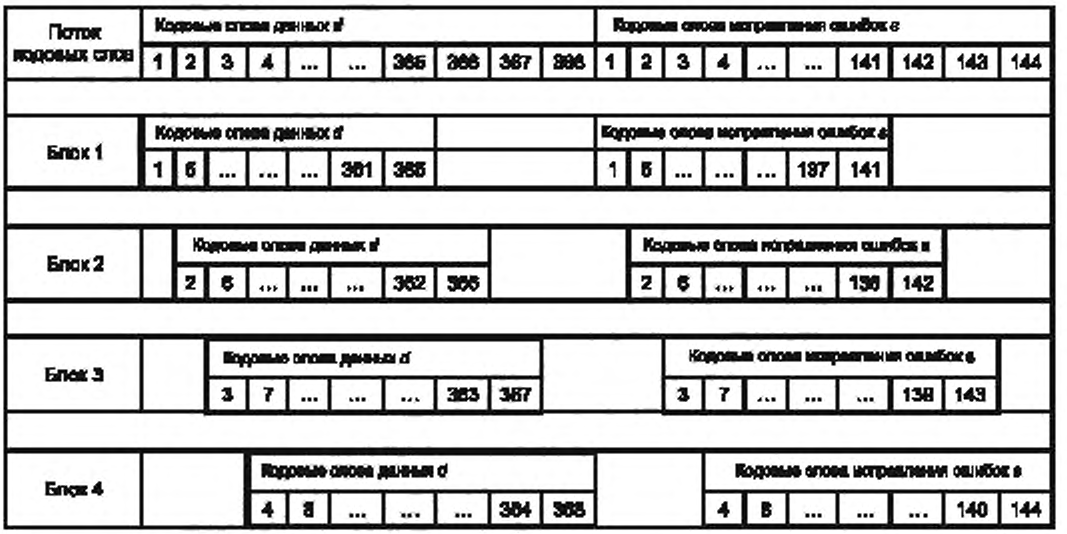


Рисунок А.1 — Схема процесса чередования для символа размером 72 × 72

А.2 Начальная последовательность для чередования в символах различных размеров

Последовательность чередующихся кодовых слов данных и кодовых слов исправления ошибок приведена в таблице А.1.

Таблица А.1 — Последовательность кодовых слов данных и исправления ошибок для символов разных размеров

Размер символа	Блок Рида-Соломона	Последовательность кодовых слов данных			Последовательность кодовых слов исправления ошибок		
52 × 52	1	1, 3, 5	...	201, 203	1, 3, 5	...	81, 83
	2	2, 4, 6	...	202, 204	2, 4, 6	...	82, 84
64 × 64	1	1, 3, 5	...	277, 279	1, 3, 5	...	109, 111
	2	2, 4, 6	...	278, 280	2, 4, 6	...	110, 112
72 × 72	1	1, 5, 9	...	361, 365	1, 5, 9	...	137, 141
	2	2, 6, 10	...	362, 366	2, 6, 10	...	138, 142
	3	3, 7, 11	...	363, 367	3, 7, 11	...	139, 143
	4	4, 8, 12	...	364, 368	4, 8, 12	...	140, 144
80 × 80	1	1, 5, 9	...	449, 453	1, 5, 9	...	185, 189

Окончание таблицы А.1

Размер символа	Блок Рида-Соломона	Последовательность кодовых слов данных			Последовательность кодовых слов исправления ошибок		
	2	2, 6, 10	...	450, 454	2, 6, 10	...	186, 190
	3	3, 7, 11	...	451, 455	3, 7, 11	...	187, 191
	4	4, 8, 12	...	452, 456	4, 8, 12	...	188, 192
88 × 88	1	1, 5, 9	...	569, 573	1, 5, 9	...	217, 221
	2	2, 6, 10	...	570, 574	2, 6, 10	...	218, 222
	3	3, 7, 11	...	571, 575	3, 7, 11	...	219, 223
	4	4, 8, 12	...	572, 576	4, 8, 12	...	220, 224
96 × 96	1	1, 5, 9	...	689, 693	1, 5, 9	...	265, 269
	2	2, 6, 10	...	690, 694	2, 6, 10	...	266, 270
	3	3, 7, 11	...	691, 695	3, 7, 11	...	267, 271
	4	4, 8, 12	...	692, 696	4, 8, 12	...	268, 272
104 × 104	1	1, 7, 13	...	805, 811	1, 7, 13	...	325, 331
	2	2, 8, 14	...	806, 812	2, 8, 14	...	326, 332
	3	3, 9, 15	...	807, 813	3, 9, 15	...	327, 333
	4	4, 10, 16	...	808, 814	4, 10, 16	...	328, 334
	5	5, 11, 17	...	809, 815	5,11,17	...	329, 335
	6	6, 12, 18	...	810, 816	6, 12, 18	...	330, 336
120 × 120	1	1, 7, 13	...	1039, 1045	1, 7, 13	...	397, 403
	2	2, 8, 14	...	1040, 1046	2, 8, 14	...	398, 404
	3	3, 9, 15	...	1041, 1047	3, 9, 15	...	399, 405
	4	4, 10, 16	...	1042, 1048	4, 10, 16	...	400, 406
	5	5, 11, 17	...	1043, 1049	5, 11, 17	...	401, 407
	6	6, 12, 18	...	1044, 1050	6, 12, 18	...	402, 408
132 × 132	1	1, 9, 17	...	1289, 1297	1, 9, 17	...	481, 489
	2	2, 10, 18	...	1290, 1298	2, 10, 18	...	482, 490
	3	3, 11, 19	...	1291,1299	3, 11, 19	...	483, 491
	4	4, 12, 20	...	1292, 1300	4, 12, 20	...	484, 492
	5	5, 13, 21	...	1293, 1301	5, 13, 21	...	485, 493
	6	6, 14, 22	...	1294, 1302	6, 14, 22	...	486, 494
	7	7, 15, 23	...	1295, 1303	7, 15, 23	...	487, 495
	8	8, 16, 24	...	1296, 1304	8, 16, 24	...	488, 496
144 × 144	1	1, 11, 21	...	1541, 1551	1, 11, 21	...	601, 611
	2	2, 12, 22	...	1542, 1552	2, 12, 22	...	602, 612
	3	3, 13, 23	...	1543, 1553	3, 13, 23	...	603, 613
	4	4, 14, 24	...	1544, 1554	4, 14, 24	...	604, 614
	5	5, 15, 25	...	1545, 1555	5, 15, 25	...	605, 615
	6	6, 16, 26	...	1546, 1556	6, 16, 26	...	606, 616
	7	7, 17, 27	...	1547, 1557	7, 17, 27	...	607, 617
	8	8, 18, 28	...	1548, 1558	8, 18, 28	...	608, 618
	9	9, 19, 29	...	1549	9, 19, 29	...	609, 619
	10	10, 20, 30	...	1550	10, 20, 30	...	610, 620

Приложение В
(обязательное)

Шаблонная рандомизация в символике версии ECC 200

Алгоритмы шаблонной рандомизации преобразуют кодовое слово в заданной позиции на входе в новое рандомизированное (псевдослучайное) кодовое слово на выходе.

В.1 Алгоритм 253 состояний

Указанный алгоритм добавляет псевдослучайное число к значению кодового слова-заполнителя. Псевдослучайное число всегда будет в диапазоне от 1 до 253, а рандомизированное значение кодового слова-заполнителя будет в диапазоне от 1 до 254.

Переменная позиция кодового слова-заполнителя является номером кодового слова исходных данных от начала закодированных данных.

В.1.1 Алгоритм рандомизации 253 состояний[1)]

```
INPUT (Pad_codeword_value, Pad_codeword_position)
pseudo_random_number = ( (149 × Pad_codeword_position) mod 253) + 1
temp_variable = Pad_codeword_value + pseudo_random_number
IF (temp_variable <= 254)
    OUTPUT (randomised_Pad_codeword_value = temp_variable)
ELSE
    OUTPUT (randomised_Pad_codeword_value = temp_variable – 254)
```

В.1.2 Алгоритм дерандомизации 253 состояний[1)]

```
INPUT (randomised_Pad_codeword_value, Pad_codeword_position)
pseudo_random_number = ( (149 × Pad_codeword_position) mod 253 ) + 1
temp_variable = randomised_Pad_codeword_value - pseudo_random_number
IF (temp_variable >= 1)
    OUTPUT (Pad_codeword_value = temp_variable)
ELSE
    OUTPUT (Pad_codeword_value = temp_variable + 254)
```

В.2 Алгоритм 255 состояний

Указанный алгоритм добавляет псевдослучайное число к значению кодового слова в схеме кодирования по основанию 256. Псевдослучайное число всегда будет находиться в диапазоне от 1 до 255, а рандомизированное значение кодового слова в схеме кодирования по основанию 256 — в диапазоне от 0 до 255.

Переменная позиция кодового слова по основанию 256 (Base256_codeword_position) является номером кодового слова исходных данных от начала кодированных данных.

[1)] *Соответствие обозначений алгоритма рандомизации и дерандомизации 253 состояний:*
Pad_codeword_value — значение кодового слова-заполнителя;
Pad_codeword_position — позиция кодового слова-заполнителя;
pseudo_random_number — псевдослучайное число;
temp_variable — временная переменная;
randomised_Pad_codeword_value — рандомизированное значение кодового слова-заполнителя.

B.2.1 Алгоритм рандомизации 255 состояний[1)]

```
INPUT (Base256_codeword_value, Base256_codeword_position)
pseudo_random_number = ( (149 × Base256_codeword_position ) mod 255) + 1
temp_variable = Base256_codeword_value + pseudo_random_number
IF (temp_variable <= 255)
    OUTPUT (randomised_Base256_codeword_value = temp_variable)
ELSE
    OUTPUT (randomised_Base256_codeword_value = temp_variable – 256)
```

B.2.2 Алгоритм дерандомизации 255 состояний[1)]

```
INPUT (randomised_Base256_codeword_value, Base256_codeword_position)
pseudo_random_number = ( (149 × Base256_codeword_position ) mod 255) + 1
temp_variable=randomised_Base256_codeword_value - pseudo_random_number
IF (temp_variable >= 0)
    OUTPUT (Base256_codeword_value = temp_variable)
ELSE
    OUTPUT (Base256_codeword_value = temp_variable + 256)
```

[1)] *Соответствие обозначений алгоритма рандомизации и дерандомизации 255 состояний:*
Base256_codeword_value — значение кодового слова по основанию 256;
Base256_codeword_position — позиция кодового слова по основанию 256;
pseudo_random_number — псевдослучайное число;
temp_variable — временная переменная;
randomised_Base256_codeword_value — рандомизированное значение кодового слова по основанию 256.

Приложение С
(обязательное)

Наборы кодируемых знаков символики версии ECC 200

Таблица С.1 — Набор знаков в схеме кодирования С40

Значение в схеме	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
0	Регистр 1 (Shift 1)		NUL	0	!	33	`	96
1	Регистр 2 (Shift 2)		SOH	1	"	34	a	97
2	Регистр 3 (Shift 3)		STX	2	#	35	b	98
3	ПРОБЕЛ (Space)	32	ETX	3	$	36	c	99
4	0	48	EOT	4	%	37	d	100
5	1	49	ENQ	5	&	38	e	101
6	2	50	ACK	6	'	39	f	102
7	3	51	BEL	7	(	40	g	103
8	4	52	BS	8	)	41	h	104
9	5	53	HT	9	*	42	i	105
10	6	54	LF	10	+	43	j	106
11	7	55	VT	11	,	44	k	107
12	8	56	FF	12	-	45	l	108
13	9	57	CR	13	.	46	m	109
14	A	65	SO	14	/	47	n	110
15	B	66	SI	15	:	58	o	111
16	C	67	DLE	16	;	59	p	112
17	D	68	DC1	17	<	60	q	113
18	E	69	DC2	18	=	61	r	114
19	F	70	DC3	19	>	62	s	115
20	G	71	DC4	20	?	63	t	116
21	H	72	NAK	21	@	64	u	117
22	I	73	SYN	22	[	91	v	118
23	J	74	ETB	23	\	92	w	119
24	K	75	CAN	24	]	93	x	120
25	L	76	EM	25	^	94	y	121
26	M	77	SUB	26	_	95	z	122
27	N	78	ESC	27	FNC1		{	123
28	O	79	FS	28			|	124

Окончание таблицы С.1

Значение в схеме	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
29	P	80	GS	29			}	125
30	Q	81	RS	30	Верхний регистр (Upper Shift)		~	126
31	R	82	US	31			DEL	127
32	S	83						
33	T	84						
34	U	85						
35	V	86						
36	W	87						
37	X	88						
38	Y	89						
39	Z	90						

Примечание — Соответствие между десятичным значением знака ASCII (КОИ-7) и значением знака в схеме кодирования С40 остается неизменным вне зависимости от действующей интерпретации в расширенном канале (ECI).

Таблица С.2 — Набор знаков в схеме кодирования Text

Значение в схеме	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
0	Регистр 1 (Shift 1)	1	NUL	0	!	33	`	96
1	Регистр 2 (Shift 2)	2	SOH	1	"	34	A	65
2	Регистр 3 (Shift 3)	3	STX	2	#	35	B	66
3	ПРОБЕЛ (Space)	32	ETX	3	$	36	C	67
4	0	48	EOT	4	%	37	D	68
5	1	49	ENQ	5	&	38	E	69
6	2	50	ACK	6	'	39	F	70
7	3	51	BEL	7	(	40	G	71
8	4	52	BS	8	)	41	H	72
9	5	53	HT	9	*	42	I	73
10	6	54	LF	10	+	43	J	74
11	7	55	VT	11	,	44	K	75
12	8	56	FF	12	-	45	L	76

Окончание таблицы С.2

Значение в схеме	Основной набор		Набор регистра 1 (Shift 1)		Набор регистра 2 (Shift 2)		Набор регистра 3 (Shift 3)	
	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение	Знак	Десятичное значение
13	9	57	CR	13	.	46	M	77
14	a	97	SO	14	/	47	N	78
15	b	98	SI	15	:	58	O	79
16	c	99	DLE	16	;	59	P	80
17	d	100	DC1	17	<	60	Q	81
18	e	101	DC2	18	=	61	R	82
19	f	102	DC3	19	>	62	S	83
20	g	103	DC4	20	?	63	T	84
21	h	104	NAK	21	@	64	U	85
22	i	105	SYN	22	[	91	V	86
23	j	106	ETB	23	\	92	W	87
24	k	107	CAN	24	]	93	X	88
25	l	108	EM	25	^	94	Y	89
26	m	109	SUB	26	_	95	Z	90
27	n	110	ESC	27	FNC1		{	123
28	o	111	FS	28			|	124
29	p	112	GS	29			}	125
30	q	113	RS	30	Верхний регистр (Upper Shift)		~	126
31	r	114	US	31			DEL	127
32	s	115						
33	t	116						
34	u	117						
35	v	118						
36	w	119						
37	x	120						
38	y	121						
39	z	122						

Примечание — Соответствие между десятичным значением знака ASCII (КОИ-7) и значением знака в схеме кодирования Text остается неизменным вне зависимости от действующей интерпретации в расширенном канале (ECI).

Таблица С.3 — Набор знаков в схеме кодирования EDIFACT

Знак данных			Двоичное значение по EDIFACT	Знак данных			Двоичное значение по EDIFACT
Знак	Десятичное значение	Двоичное значение		Знак	Десятичное значение	Двоичное значение	
@	64	01000000	000000	ПРОБЕЛ (space)	32	00100000	100000
A	65	01000001	000001	!	33	00100001	100001
B	66	01000010	000010	«	34	00100010	100010
C	67	01000011	000011	#	35	00100011	100011
D	68	01000100	000100	$	36	00100100	100100
E	69	01000101	000101	%	37	00100101	100101
F	70	01000110	000110	&	38	00100110	100110
G	71	01000111	000111	'	39	00100111	100111
H	72	01001000	001000	(	40	00101000	101000
I	73	01001001	001001	)	41	00101001	101001
J	74	01001010	001010	*	42	00101010	101010
K	75	01001011	001011	+	43	00101011	101011
L	76	01001100	001100	,	44	00101100	101100
M	77	01001101	001101	-	45	00101101	101101
N	78	01001110	001110	.	46	00101110	101110
O	79	01001111	001111	/	47	00101111	101111
P	80	01010000	010000	0	48	00110000	110000
Q	81	01010001	010001	1	49	00110001	110001
R	82	01010010	010010	2	50	00110010	110010
S	83	01010011	010011	3	51	00110011	110011
T	84	01010100	010100	4	52	00110100	110100
U	85	01010101	010101	5	53	00110101	110101
V	86	01010110	010110	6	54	00110110	110110
W	87	01010111	010111	7	55	00110111	110111
X	88	01011000	011000	8	56	00111000	111000
Y	89	01011001	011001	9	57	00111001	111001
Z	90	01011010	011010	:	58	00111010	111010
[	91	01011011	011011	;	59	00111011	111011
\	92	01011100	011100	<	60	00111100	111100
]	93	01011101	011101	=	61	00111101	111101
^	94	01011110	011110	>	62	00111110	111110
Отказ от фиксации (Unlatch)		01011111	011111	?	63	00111111	111111

Примечание — Соответствие между десятичным значением знака ASCII (КОИ-7) и значением по EDIFACT остается неизменным вне зависимости от действующей интерпретации в расширенном канале (ECI).

Приложение D
(обязательное)

Направляющие шаблоны символов версии ECC 200

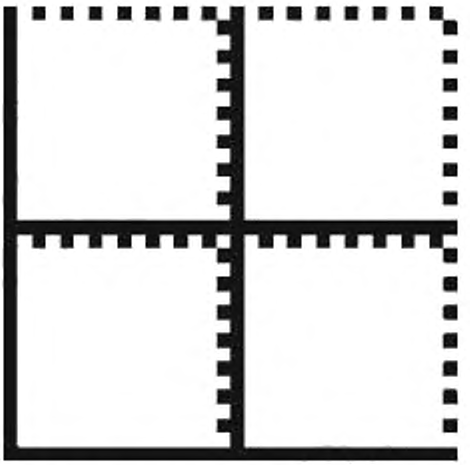


Рисунок D.1 — Конфигурация направляющего шаблона для символа квадратной формы 32 × 32 модуля

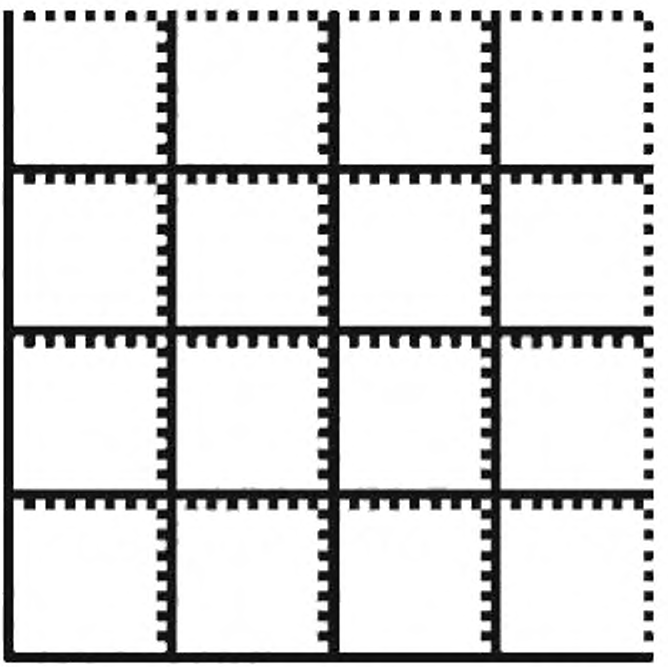


Рисунок D.2 — Конфигурация направляющего шаблона для символа квадратной формы 64 × 64 модуля

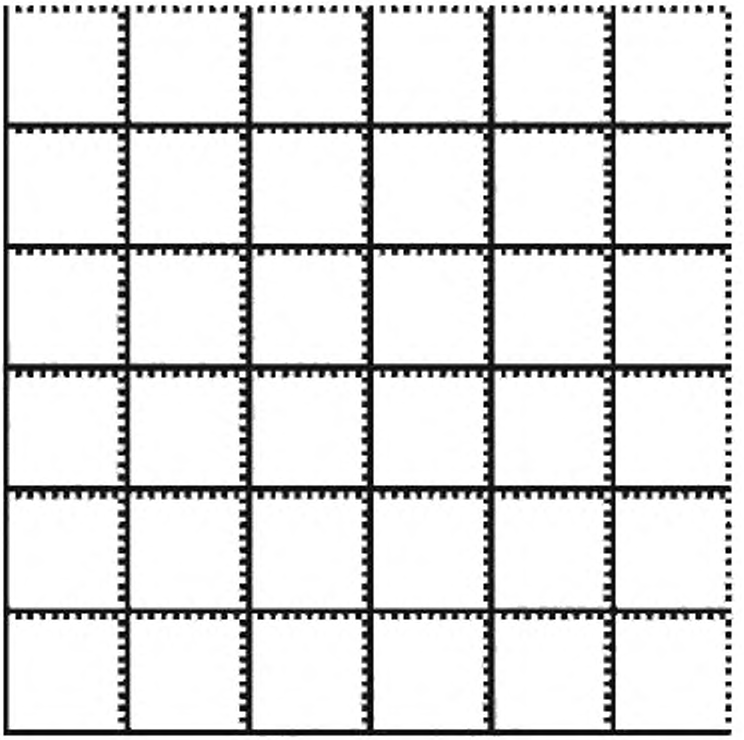


Рисунок D.3 — Конфигурация направляющего шаблона для символа квадратной формы 120 × 120 модулей

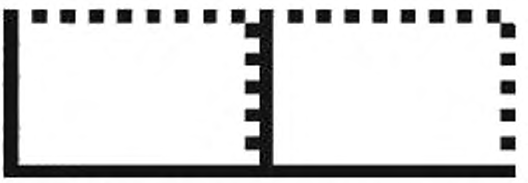


Рисунок D.4 — Конфигурация направляющего шаблона для символа прямоугольной формы 12 × 36 модулей

Приложение Е
(обязательное)

Алгоритм обнаружения и исправления ошибок Рида-Соломона для символики версии ECC 200

Е.1 Порождающие полиномы для кодовых слов исправления ошибок

Кодовые слова исправления ошибок являются коэффициентами остатка от деления на порождающий полином $g(x)$ произведения полинома данных символа $d(x)$ на x^k. Каждый порождающий полином является произведением полиномов первой степени: $x - 2^1$, $x - 2^2$,..., $x - 2^n$, где n — показатель степени порождающего полинома.

Например, порождающий полином пятой степени представляет собой

$(x + 2)(x + 4)(x + 8)(x + 16)(x + 32) = x^5 + (2 + 4 + 8 + 16 + 32)\,x^4 + ((2 \times 4) + (2 \times 8) + (2 \times 16) + (2 \times 32) + (4 \times 8) + (4 \times 16) + (4 \times 32) + (8 \times 16) + (8 \times 32) + (16 \times 32))\,x^3 + ((2 \times 4 \times 8) + (2 \times 4 \times 16) + (2 \times 4 \times 32) + (2 \times 8 \times 16) + (2 \times 8 \times 32) + (2 \times 16 \times 32) + (4 \times 8 \times 16) + (4 \times 8 \times 32) + (4 \times 16 \times 32) + (8 \times 16 \times 32))\,x^2 + ((2 \times 4 \times 8 \times 16) + (2 \times 4 \times 8 \times 32) + (2 \times 4 \times 16 \times 32) + (2 \times 8 \times 16 \times 32) + (4 \times 8 \times 16 \times 32))\,x + (2 \times 4 \times 8 \times 16 \times 32) = x^5 + 62\,x^4 + 111\,x^3 + 15\,x^2 + 48\,x + 228.$

Следует обратить внимание на то, что арифметика в этом поле Галуа не является обычной целочисленной арифметикой: операция «–» эквивалентна операции «+», которая представляет собой выполнение операции «исключающего ИЛИ» («exclusive-or») в этом поле, а операция умножения представляет собой побитовую операцию взятия по модулю 100101101 результата побитового умножения двух сомножителей.

Полином-делитель для порождения пяти проверочных знаков представляет собой

$g(x) = x^5 + 62\,x^4 + 111\,x^3 + 15\,x^2 + 48\,x + 228.$

Полином-делитель для порождения семи проверочных знаков представляет собой

$g(x) = x^7 + 254\,x^6 + 92\,x^5 + 240\,x^4 + 134\,x^3 + 144\,x^2 + 68\,x + 23.$

Полином-делитель для порождения 10 проверочных знаков представляет собой

$g(x) = x^{10} + 61\,x^9 + 110\,x^8 + 255\,x^7 + 116\,x^6 + 248\,x^5 + 223\,x^4 + 166\,x^3 + 185\,x^2 + 24\,x + 28.$

Полином-делитель для порождения 11 проверочных знаков представляет собой

$g(x) = x^{11} + 120\,x^{10} + 97\,x^9 + 60\,x^8 + 245\,x^7 + 39\,x^6 + 168\,x^5 + 194\,x^4 + 12\,x^3 + 205\,x^2 + 138\,x + 175.$

Полином-делитель для порождения 12 проверочных знаков представляет собой

$g(x) = x^{12} + 242\,x^{11} + 100\,x^{10} + 178\,x^9 + 97\,x^8 + 213\,x^7 + 142\,x^6 + 42\,x^5 + 61\,x^4 + 91\,x^3 + 158\,x^2 + 153\,x + 41.$

Полином-делитель для порождения 14 проверочных знаков представляет собой

$g(x) = x^{14} + 185\,x^{13} + 83\,x^{12} + 186\,x^{11} + 18\,x^{10} + 45\,x^9 + 138\,x^8 + 119\,x^7 + 157\,x^6 + 9\,x^5 + 95\,x^4 + 252\,x^3 + 192\,x^2 + 97\,x + 156.$

Полином-делитель для порождения 18 проверочных знаков представляет собой

$g(x) = x^{18} + 188\,x^{17} + 90\,x^{16} + 48\,x^{15} + 225\,x^{14} + 254\,x^{13} + 94\,x^{12} + 129\,x^{11} + 109\,x^{10} + 213\,x^9 + 241\,x^8 + 61\,x^7 + 66\,x^6 + 75\,x^5 + 188\,x^4 + 39\,x^3 + 100\,x^2 + 195\,x + 83.$

Полином-делитель для порождения 20 проверочных знаков представляет собой

$g(x) = x^{20} + 172\,x^{19} + 186\,x^{18} + 174\,x^{17} + 27\,x^{16} + 82\,x^{15} + 108\,x^{14} + 79\,x^{13} + 253\,x^{12} + 145\,x^{11} + 153\,x^{10} + 160\,x^9 + 188\,x^8 + 2\,x^7 + 168\,x^6 + 71\,x^5 + 233\,x^4 + 9\,x^3 + 244\,x^2 + 195\,x + 15.$

Полином-делитель для порождения 24 проверочных знаков представляет собой

$g(x) = x^{24} + 193\,x^{23} + 50\,x^{22} + 96\,x^{21} + 184\,x^{20} + 181\,x^{19} + 12\,x^{18} + 124\,x^{17} + 254\,x^{16} + 172\,x^{15} + 5\,x^{14} + 21\,x^{13} + 155\,x^{12} + 223\,x^{11} + 251\,x^{10} + 197\,x^9 + 155\,x^8 + 21\,x^7 + 176\,x^6 + 39\,x^5 + 109\,x^4 + 205\,x^3 + 88\,x^2 + 190\,x + 52.$

Полином-делитель для порождения 28 проверочных знаков представляет собой:

$g(x) = x^{28} + 255\,x^{27} + 93\,x^{26} + 168\,x^{25} + 233\,x^{24} + 151\,x^{23} + 120\,x^{22} + 136\,x^{21} + 141\,x^{20} + 213\,x^{19} + 110\,x^{18} + 138\,x^{17} + 17\,x^{16} + 121\,x^{15} + 249\,x^{14} + 34\,x^{13} + 75\,x^{12} + 53\,x^{11} + 170\,x^{10} + 151\,x^9 + 37\,x^8 + 174\,x^7 + 103\,x^6 + 96\,x^5 + 71\,x^4 + 97\,x^3 + 43\,x^2 + 231\,x + 211.$

Полином-делитель для порождения 36 проверочных знаков представляет собой

$g(x) = x^{36} + 112\,x^{35} + 81\,x^{34} + 98\,x^{33} + 225\,x^{32} + 25\,x^{31} + 59\,x^{30} + 184\,x^{29} + 175\,x^{28} + 44\,x^{27} + 115\,x^{26} + 119\,x^{25} + 95\,x^{24} + 137\,x^{23} + 101\,x^{22} + 33\,x^{21} + 68\,x^{20} + 4\,x^{19} + 2\,x^{18} + 18\,x^{17} + 229\,x^{16} + 182\,x^{15} + 80\,x^{14} + 251\,x^{13} + 220\,x^{12} + 179\,x^{11} + 84\,x^{10} + 120\,x^{9} + 102\,x^{8} + 181\,x^{7} + 162\,x^{6} + 250\,x^{5} + 130\,x^{4} + 218\,x^{3} + 242\,x^{2} + 127\,x + 245.$

Полином-делитель для порождения 42 проверочных знаков представляет собой

$g(x) = x^{42} + 5\,x^{41} + 9\,x^{40} + 5\,x^{39} + 226\,x^{38} + 177\,x^{37} + 150\,x^{36} + 50\,x^{35} + 69\,x^{34} + 202\,x^{33} + 248\,x^{32} + 101\,x^{31} + 54\,x^{30} + 57\,x^{29} + 253\,x^{28} + x^{27} + 21\,x^{26} + 121\,x^{25} + 57\,x^{24} + 111\,x^{23} + 214\,x^{22} + 105\,x^{21} + 167\,x^{20} + 9\,x^{19} + 100\,x^{18} + 95\,x^{17} + 175\,x^{16} + 8\,x^{15} + 242\,x^{14} + 133\,x^{13} + 245\,x^{12} + 2\,x^{11} + 122\,x^{10} + 105\,x^{9} + 247\,x^{8} + 153\,x^{7} + 22\,x^{6} + 38\,x^{5} + 19\,x^{4} + 31\,x^{3} + 137\,x^{2} + 193\,x + 77.$

Полином-делитель для порождения 48 проверочных знаков представляет собой

$g(x) = x^{48} + 19\,x^{47} + 225\,x^{46} + 253\,x^{45} + 92\,x^{44} + 213\,x^{43} + 69\,x^{42} + 175\,x^{41} + 160\,x^{40} + 147\,x^{39} + 187\,x^{38} + 87\,x^{37} + 176\,x^{36} + 44\,x^{35} + 82\,x^{34} + 240\,x^{33} + 186\,x^{32} + 138\,x^{31} + 66\,x^{30} + 100\,x^{29} + 120\,x^{28} + 88\,x^{27} + 131\,x^{26} + 205\,x^{25} + 170\,x^{24} + 90\,x^{23} + 37\,x^{22} + 23\,x^{21} + 118\,x^{20} + 147\,x^{19} + 16\,x^{18} + 106\,x^{17} + 191\,x^{16} + 87\,x^{15} + 237\,x^{14} + 188\,x^{13} + 205\,x^{12} + 231\,x^{11} + 238\,x^{10} + 133\,x^{9} + 238\,x^{8} + 22\,x^{7} + 117\,x^{6} + 32\,x^{5} + 96\,x^{4} + 223\,x^{3} + 172\,x^{2} + 132\,x + 245.$

Полином-делитель для порождения 56 проверочных знаков представляет собой

$g(x) = x^{56} + 46\,x^{55} + 143\,x^{54} + 53\,x^{53} + 233\,x^{52} + 107\,x^{51} + 203\,x^{50} + 43\,x^{49} + 155\,x^{48} + 28\,x^{47} + 247\,x^{46} + 67\,x^{45} + 127\,x^{44} + 245\,x^{43} + 137\,x^{42} + 13\,x^{41} + 164\,x^{40} + 207\,x^{39} + 62\,x^{38} + 117\,x^{37} + 201\,x^{36} + 150\,x^{35} + 22\,x^{34} + 238\,x^{33} + 144\,x^{32} + 232\,x^{31} + 29\,x^{30} + 203\,x^{29} + 117\,x^{28} + 234\,x^{27} + 218\,x^{26} + 146\,x^{25} + 228\,x^{24} + 54\,x^{23} + 132\,x^{22} + 200\,x^{21} + 38\,x^{20} + 223\,x^{19} + 36\,x^{18} + 159\,x^{17} + 150\,x^{16} + 235\,x^{15} + 215\,x^{14} + 192\,x^{13} + 230\,x^{12} + 170\,x^{11} + 175\,x^{10} + 29\,x^{9} + 100\,x^{8} + 208\,x^{7} + 220\,x^{6} + 17\,x^{5} + 12\,x^{4} + 238\,x^{3} + 223\,x^{2} + 9\,x + 175.$

Полином-делитель для порождения 62 проверочных знаков представляет собой

$g(x) = x^{62} + 204\,x^{61} + 11\,x^{60} + 47\,x^{59} + 86\,x^{58} + 124\,x^{57} + 224\,x^{56} + 166\,x^{55} + 94\,x^{54} + 7\,x^{53} + 232\,x^{52} + 107\,x^{51} + 4\,x^{50} + 170\,x^{49} + 176\,x^{48} + 31\,x^{47} + 163\,x^{46} + 17\,x^{45} + 188\,x^{44} + 130\,x^{43} + 40\,x^{42} + 10\,x^{41} + 87\,x^{40} + 63\,x^{39} + 51\,x^{38} + 218\,x^{37} + 27\,x^{36} + 6\,x^{35} + 147\,x^{34} + 44\,x^{33} + 161\,x^{32} + 71\,x^{31} + 114\,x^{30} + 64\,x^{29} + 175\,x^{28} + 221\,x^{27} + 185\,x^{26} + 106\,x^{25} + 250\,x^{24} + 190\,x^{23} + 197\,x^{22} + 63\,x^{21} + 245\,x^{20} + 230\,x^{19} + 134\,x^{18} + 112\,x^{17} + 185\,x^{16} + 37\,x^{15} + 196\,x^{14} + 108\,x^{13} + 143\,x^{12} + 189\,x^{11} + 201\,x^{10} + 188\,x^{9} + 202\,x^{8} + 118\,x^{7} + 39\,x^{6} + 210\,x^{5} + 144\,x^{4} + 50\,x^{3} + 169\,x^{2} + 93\,x + 242.$

Полином-делитель для порождения 68 проверочных знаков представляет собой

$g(x) = x^{68} + 186\,x^{67} + 82\,x^{66} + 103\,x^{65} + 96\,x^{64} + 63\,x^{63} + 132\,x^{62} + 153\,x^{61} + 108\,x^{60} + 54\,x^{59} + 64\,x^{58} + 189\,x^{57} + 211\,x^{56} + 232\,x^{55} + 49\,x^{54} + 25\,x^{53} + 172\,x^{52} + 52\,x^{51} + 59\,x^{50} + 241\,x^{49} + 181\,x^{48} + 239\,x^{47} + 223\,x^{46} + 136\,x^{45} + 231\,x^{44} + 210\,x^{43} + 96\,x^{42} + 232\,x^{41} + 220\,x^{40} + 25\,x^{39} + 179\,x^{38} + 167\,x^{37} + 202\,x^{36} + 185\,x^{35} + 153\,x^{34} + 139\,x^{33} + 66\,x^{32} + 236\,x^{31} + 227\,x^{30} + 160\,x^{29} + 15\,x^{28} + 213\,x^{27} + 93\,x^{26} + 122\,x^{25} + 68\,x^{24} + 177\,x^{23} + 158\,x^{22} + 197\,x^{21} + 234\,x^{20} + 180\,x^{19} + 248\,x^{18} + 136\,x^{17} + 213\,x^{16} + 127\,x^{15} + 73\,x^{14} + 36\,x^{13} + 154\,x^{12} + 244\,x^{11} + 147\,x^{10} + 33\,x^{9} + 89\,x^{8} + 56\,x^{7} + 159\,x^{6} + 149\,x^{5} + 251\,x^{4} + 89\,x^{3} + 173\,x^{2} + 228\,x + 220.$

Е.2 Алгоритм обнаружения и исправления ошибок

Алгоритм Петерсона-Горенштейна-Зирлера может быть использован для исправления ошибок в декодированных символах версии ECC 200.

Нижеуказанные вычисления следуют этому алгоритму исправления ошибок, используя кодовые слова исправления ошибок Рида-Соломона.

Стирания должны быть исправлены как ошибки путем первоначального заполнения позиций всех стертых кодовых слов фиктивными значениями.

Все вычисления проводят с помощью арифметических операций $GF(2^8)$. Сложение и вычитание в этом поле соответствует проведению двоичной операции «исключающего ИЛИ» («exclusive-or» – XOR). Умножение и деление могут быть выполнены с помощью таблиц логарифмов и антилогарифмов.

Составляют полином знаков символа $C(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + ... + C_1x^1 + C_0$,

где n коэффициентов полинома являются считанными кодовыми словами, причем C_{n-1} относится к первому знаку символа, а n — общее число знаков символа.

Вычисляют i величин синдромов от S_0 до S_{i-1} путем вычисления полинома $C(x)$ при $x = 2^k$ для k = от 1 до i, где i — число кодовых слов исправления ошибок в символе.

Составляют и решают систему j уравнений с j неизвестными от L_0 до L_{j-1}, используя i синдромов:

$$S_0L_0 + S_1L_1 + ... + S_{j-1}L_{j-1} = S_j;$$

$$S_1L_0 + S_2L_1 + ... + S_jL_{j-1} = S_{j+1};$$

$$\vdots$$

$$\vdots$$

$S_{j-1}L_0 + S_jL_1 + ... + S_{2j-2}L_{j-1} = S_{2j-1}$;

где $j = i / 2$.

Составляют полином указания местонахождения ошибок

$L(x) = L_{j-1}x^j + L_{j-2}x^{j-1} + ... + L_0x + 1$.

из j величин L, вычисленных выше. Вычисляют $L(x)$ для $x = 2^k$, где k = от 0 до $n - 1$, где n — общее число кодовых слов в символе.

Как только $L(2^k) = 0$, позиция ошибки определяется как $(n - 1) - k$. Если найдено больше местоположений ошибок, чем значение j, данный символ невозможно исправить.

Сохраняют местоположения ошибок в m переменных указателях местоположения ошибок от E_0 до E_{m-1}, где m — число найденных местоположений ошибок. Составляют и решают систему m уравнений с m неизвестными от X_0 до X_{m-1} (значения ошибок), используя переменные указатели местоположения ошибок и первые m синдромов S:

$E_0X_0 + E_1X_1 + ... + E_{m-1}X_{m-1} = S_0$;

$E_0^2X_0 + E_1^2X_1 + ... + E_{(m-1)}^2X_{m-1} = S_1$;

$E_0^3X_0 + E_1^3X_1 + ... + E_{(m-1)}^3X_{m-1} = S_2$;

$\vdots$

$E_0^mX_0 + E_1^mX_1 + ... + E_{(m-1)}^mX_{m-1} = S_{m-1}$.

Добавляют значения ошибок от X_0 до X_{m-1} к значениям знаков символа в соответствующих местоположениях ошибок от E_0 до E_{m-1} для исправления ошибок.

Примечание — $E_0, ..., E_{m-1}$ являются корнями полинома указателя местоположения ошибок.

Указанный алгоритм, написанный на языке программирования С, можно приобрести в международной организации AIM Inc. на дискете разработчиков Data Matrix [3].

Е.3 Вычисление кодовых слов исправления ошибки

Следующий пример программы на языке программирования С вычисляет кодовые слова исправления ошибки для заданной входной строки данных длиной «nd», записанной в целочисленном массиве wd[ ]. Функция ReedSolomon( ) сначала генерирует таблицы логарифмов и антилогарифмов для поля Галуа размером «gf» (для символов версии ECC 200 оно равно 2^8) с примитивным полиномом «pp» (для символов версии ECC 200, равным 301), затем использует их в функции prod( ), сначала для вычисления коэффициентов порождающего полинома степени «nc», а затем для вычисления «nc» дополнительных проверочных кодовых слов, которые добавляют к кодовым словам данных в массиве wd[ ].

```
/* «prod(x, y, log ,alog, gf)» returns the product «x» times «y» */[1]
int prod(int x, int y, int *log, int *alog, int gf)   {
    if (!x  ||  !y) return 0;
    ELSE return alog[(log[x] + log[y])   %   (gf-1)];
}
/* «ReedSolomon(wd, nd, nc, gf.pp)» takes «nd» data codeword values in wd[ ] */
/* and adds on «nc» check codewords, all within GF(gf) where «gf» is a */
/* power of 2 and «pp» is the value of its prime modulus polynomial */[2]
void ReedSolomon(int *wd, int nd, int nc, int gf, int pp) {
    int i, j, k, *log,*alog,*c;
/* allocate, then generate the log & antilog arrays: */[3]
    log = malloc(sizeof(int) * gf);
    alog = malloc(sizeof(int) * gf);
    log[0] = 1-gf; alog[0] = 1;
for (i = 1; i < gf; i++) {
    alog[i] = alog[i-1] * 2;
    if (alog[i] >= gf) alog[i] ^= pp;
    log[alog[i]] = i;
```

---

[1] *«prod(x, y, log, alog, gf)» возвращает произведение «x» на «y».*

[2] *«ReedSolomon(wd,nd,nc,gf.pp)» берет «nd» значений кодовых слов данных из wd[ ] и добавляет к «nc» проверочным словам все в поле Галуа GF(gf), где «gf» является степенью 2, а «pp» представляет собой значение примитивного полинома.*

[3] *Выделяют память, затем генерируют массивы логарифмов и антилогарифмов.*

```
    }
    /* allocate, then generate the generator polynomial coefficients: */ [1]
        c = malloc(sizeof(int) * (nc+1));
        for (i=1; i<=nc; i++) c[i] = 0; c[0] = 1;
        for (i=1; i<=nc; i++) {
            c[i] = c[i-1];
            for (j=i-1; j>=1; j—) {
                c[j] = c[j-1] ^ prod(c[j],alog[i],log,alog,gf);
            }
            c[0] = prod(c[0],alog[i],log,alog,gf);
        }
    /* clear, then generate «nc» checkwords in the array wd[ ] : */ [2]
        for (i=nd; i<=(nd+nc); i++) wd[i] = 0;
        for (i=0; i<nd; i++) {
            k = wd[nd] ^ wd[i] ;
            for (j=0; j<nc; j++) {
                wd[nd+j] = wd[nd+j+l] ^ prod(k,c[nc-j-1],log, alog,gf);
            }
        }
        free(c);
        free(alog);
        free(log);
    }
```

[1] *Выделяют память, затем генерируют коэффициенты порождающего полинома.*

[2] *Очищают память, затем генерируют «nc» кодовых слов в массиве wd[ ].*

Приложение F
(обязательное)

Размещение знаков символа в символе версии ECC 200

F.1 Программа размещения знаков символа

Следующая программа на языке программирования С генерирует схемы размещения знаков символа:

```
#include <stdio.h>
#include <alloc.h>
int nrow, ncol, * array;
/* «module» places «chr+bit» with appropriate wrapping within array[ ] */[1]
void module (int row, int col, int chr, int bit)
{ if (row < 0) { row += nrow; col += 4 - ((nrow+4)%8); }
    if (col < 0) { col += ncol; row += 4 - ((ncol+4)%8); }
    array[row*ncol+col] = 10*chr + bit;
}
/* «utah» places the 8 bits of a utah-shaped symbol character in ECC200 */[2]
void utah(int row, int col, int chr)
{ module(row-2, col-2, chr, 1);
    module(row-2,col-1,chr, 2);
    module(row-l, col-2, chr, 3);
    module(row-1, col-1, chr, 4);
    module(row-1, col, chr, 5);
    module(row, col-2, chr, 6);
    module(row, col-1, chr, 7);
    module(row, col, chr, 8);
}
/* «cornerN» places 8 bits of the four special corner cases in ECC200*/[3]
void corner1(int chr)
{ module(nrow-1, 0, chr, 1);
    module(nrow-1, 1, chr, 2);
    module(nrow-1, 2, chr, 3);
    module(0,ncol-2,chr,4);
    module(0,ncol-1, chr, 5);
    module(1,ncol-1, chr, 6);
    module(2,ncol-1, chr, 7);
    module(3,ncol-1, chr, 8);
}
void corner2 (int chr)
{ module(nrow-3, 0, chr, 1);
    module(nrow-2, 0, chr, 2);
    module(nrow-1, 0, chr, 3);
    module(0, ncol-4, chr, 4);
    module(0, ncol-3 ,chr ,5);
    module(0, ncol-2, chr, 6);
    module(0, ncol-1, chr, 7);
    module(1, ncol-1, chr ,8);
}
void corner3(int chr)
{ module(nrow-3, 0, chr, 1);
```

---

[1] *Процедура «module» записывает в матрицу array[ ] числа вида «10* chr+bit», где chr — порядковый номер знака символа, а bit — номер бита данного знака символа, которые записывают, начиная с первого знака и бита.*

[2] *Процедура «Utah» размещает 8 битов знака символа, имеющего форму штата Юта, в символе версии ECC200.*

[3] *Процедура «corner» размещает 8 битов для четырех специальных угловых случаев в символе версии ECC200.*

```
        module(nrow-2, 0, chr, 2);
        module(nrow-1, 0, chr, 3);
        module(0, ncol-2, chr, 4);
        module(0, ncol-1, chr, 5);
        module(1, ncol-1, chr, 6);
        module(2, ncol-1, chr, 7);
        module(3, ncol-1, chr, 8);
    }
    void corner4(int chr)
    { module(nrow-1,0, chr,1);
        module(nrow-1, ncol-1,chr,2);
        module(0, ncol-3, chr,3);
        module(0, ncol-2, chr,4);
        module(0, ncol-1, chr,5);
        module(1, ncol-3, chr,6);
        module(1, ncol-2, chr,7);
        module(1, ncol-1, chr,8);
    }
    /* «ecc200» fills an nrow x ncol array with appropriate values for ECC200*/[1]
    void ecc200(void)
    { int row, col, chr;
    /* First, fill the array[ ] with invalid entries*/[2]
        for (row=0; row<nrow; row++) {
            for (col=0; col<ncol; col++) {
              array[row*ncol+col] = 0;
            }
        }
    /* Starting in the correct location for character #1, bit 8,. */[3]
        chr = 1; row = 4; col = 0;
        do {
    /* repeatedly first check for one of the special corner cases, then... */[4]
            if ((row == nrow) && (col == 0) ) corner1 (chr++) ;
            if ((row == nrow-2) && (col == 0) && (ncol%4) ) corner2 (chr++) ;
            if ((row == nrow-2) && (col == 0) && (ncol%8 == 4)) corner3 (chr++) ;
            if ((row == nrow+4) && (col == 2) && (!(ncol%8) ) ) corner4 (chr++);
    /* sweep upward diagonally, inserting successive characters,... */[5]
        do {
            if ((row < nrow) && (col >= 0) && (!array[row*ncol+col]))
                utah (row, col, chr++);
            row -= 2; col += 2;
        } while ((row >= 0) && (col < ncol)) ;
        row += 1; col += 3;
    /* & then sweep downward diagonally, inserting successive haracters,... */[6]
        +
        do {
            if ((row >= 0) && (col < ncol) && (!array[row*ncol+col]))
                utah (row, col, chr++);
            row += 2; col -= 2;
        } while ((row < nrow) && (col >= 0) ) ;
        row += 3; col += 1;
    /* ... until the entire array is scanned */[7]
```

---

[1] *Процедура «ecc200» заполняет nrow × ncol массив соответствующими значениями для символа версии ECC200.*

[2] *Сначала заполняют массив array[ ] недопустимыми значениями.*

[3] *Фиксируют начальное положение модуля в матрице для знака с порядковым номером (chr) 1 и бита с номером 8 и начинают заполнять матрицу.*

[4] *Повторно (в цикле) проверяют на предмет одного из специальных угловых случаев, тогда ... .*

[5] *Продвигаясь вверх по диагонали, вставляют последовательные знаки,... .*

[6] *Затем продвигаясь вниз по диагонали, вставляют последовательные знаки... .*

[7] *...пока не просмотрят весь массив.*

```
    } while ((row < nrow) || (col < ncol));
/* Lastly, if the lower righthand corner is untouched, fill in fixed pattern */ [1]
    if (!array[nrow*ncol-1]) {
        array[nrow*ncol-1] = array[nrow*ncol-ncol-2] = 1;
    }
}

/* «main» checks for valid command line entries, then computes & displays array  */ [2]
void main (int argc, char *argv[ ])
{ int x, y, z;
    if (argc =< 3) {
        printf(«Command line: ECC200 #_of_Data_Rows #_of_Data_Columns\n»);
        } ELSE {
          nrow = ncol = 0;
          nrow = atoi(argv[1]); ncol = atoi(argv[2]);
          if ((nrow >= 6) && (–nrow&0x0l) && (ncol >= 6) && (–ncol&0x01)) {
              array = malloc(sizeof(int) * nrow * ncol);
              ECC200 ( ):
              for (x=0; x<nrow; x++) {
                   for (y=0; y<ncol; y++) {
                       z = array[x*ncol+y];
                       if (z == 0) printf (« WHI»);
                         ELSE if (z == 1) printf («BLK») ;
                           ELSE printf («%3d.%d»,z/10,z%10) ;
                   }
                   Printf («\n») ;
              }
              Free (array) ;
            }
        }
}
```

F.2 Правила размещения знаков символа

F.2.1 Нетипичная форма знака символа

Ввиду того, что знаки символа типичной формы не всегда могут быть размещены в границах модулей данных в символе и в некоторых его углах, необходим небольшой набор знаков символа нетипичной формы. Установлено шесть условий размещения — два условия размещения на границах, которые действуют во всех форматах символа, и четыре различных условия размещения в углах, которые применимы для отдельных форматов символа:

a) одну часть знака символа нетипичной формы размещают на одной стороне символа и другую — на противоположной. Это условие применяют к двум базовым формам знака символа (рисунок F.1). Варианты таких компоновок зависят от взаимосвязи между строками на левой и правой границах (таблица F.1);

b) одну часть знака символа размещают на верхней границе символа, а другую — на нижней. Это условие применяют к двум базовым формам знака символа (рисунок F.2). Варианты такой компоновки зависят от взаимосвязи между столбцами на верхней и нижней границах (таблица F.1);

c) четыре знака символа нетипичной формы распределяют по двум или трем углам (рисунки F.3 — F.6). Знаки символа нетипичной формы размещают на противоположных границах. Число таких пар возрастает прямо пропорционально периметру координатной матрицы. Базовые комбинации представлены на рисунках F.1 и F.2. На рисунке F.1 модули a8 и a7 находятся в одной и той же строке, так же как и модули b7 и b6. На рисунке F.2 модули c6 и c3 находятся в одном и том же столбце, так же как и модули d3 и d1. Существует семь вариантов размещения знаков символа на границах, которые определяют относительное положение знаков символа по вертикали (рисунок F.1), положение по горизонтали (рисунок F.2) и условия размещения в углах.

---

[1] *И наконец, если не заполнен нижний правый угол, заполняют его, используя фиксированный шаблон.*

[2] *Процедура «main» проверяет правильность командной строки, затем вычисляет и отображает массив array.*

Таблица F.1 — Факторы, определяющие вариант размещения на границах

Вариант размещения на границах	Взаимосвязь модулей a8 и a7 в строках	Взаимосвязь модулей c6 и c3 в столбцах	Рисунок, поясняющий условия в углах	Используемые отображающие матрицы	Рисунки приложения F с примером
1	Строка с модулем a7 = строке с модулем a8	Столбец с модулем c3 = столбцу с модулем c6	нет	Квадратные: 8^2, 16^2, 24^2,32^2, 40^2, 48^2,56^2, 64^2, 72^2, 80^2, 88^2, 96^2 и 120^2	F.9 и F.16
2	Строка с модулем a7 = строке с модулем a8 – 2	Столбец с модулем c3 = столбцу с модулем c6 – 2	нет	Квадратные: 10^2 и 18^2	F.10 и F.17
3	Строка с модулем a7 = строке с модулем a8 + 4	Столбец с модулем c3 = столбцу с модулем c6 + 4	F.3	Квадратные: 12^2, 20^2, 28^2, 36^2, 44^2, 108^2 и 132^2	F.11 и F.18
4	Строка с модулем a7 = строке с модулем a8 + 2	Столбец с модулем c3 = столбцу с модулем c6 + 2	F.4	Квадратные: 14^2 и 22^2	F.12 и F.19
5	Строка с модулем a7 = строке с модулем a8	Столбец c3 = столбцу с модулем c6 + 2	F.5	Прямоугольные: 6×16 и 14×32	F.13
6	Строка с модулем a7 = строке с модулем a8	Столбец с модулем c3 = столбцу с модулем c6 – 2	нет	Прямоугольные: 10×24 и 10×32	F.14
7	Строка с модулем a7 = строке с модулем a8 + 4	Столбец с модулем c3 = столбцу с модулем c6 + 2	F.6	Прямоугольные: 6×28 и 14×44	F.15

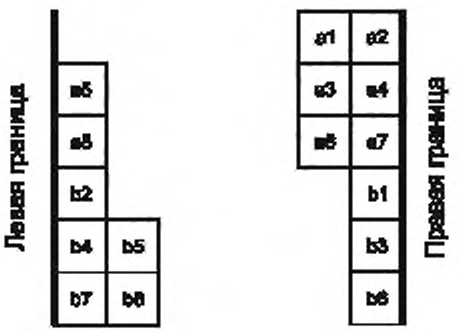


Рисунок F.1 — Правый и левый знаки символа



Рисунок F.2 — Верхний и нижний знаки символа

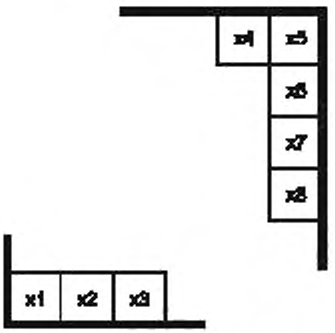


Рисунок F.3 — Первое условие размещения знака символа в углах

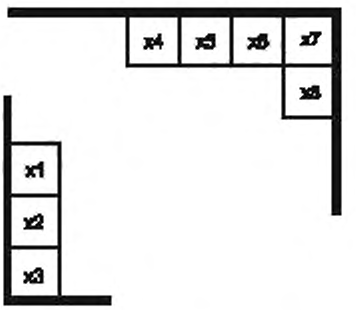


Рисунок F.4 — Второе условие размещения знака символа в углах

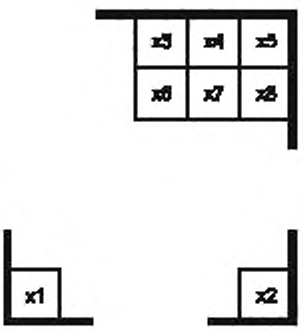


Рисунок F.5 — Третье условие размещения знака символа в углах

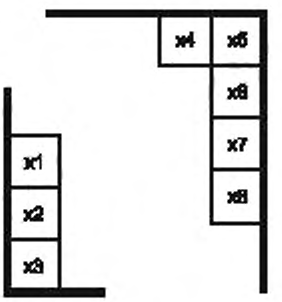


Рисунок F.6 — Четвертое условие размещения знака символа в углах

Примечание 1 — Для идентификации знака символа используют алгебраические обозначения, поскольку они варьируются в зависимости от формата символа.

Примечание 2 — Угловые знаки идентифицируют по модулю в левом нижнем и правом верхнем углах.

F.2.2 Размещение знаков символа

Знаки символа помещают в матрицу следующим образом:

a) создают координатную матрицу:

1) в небольших символах с единственной областью данных эта область совпадает с координатной матрицей;

2) в больших символах, имеющих более одной области данных, координатная матрица соответствует совокупному размеру смежных областей данных. Таким образом, координатная матрица не содержит разделительных направляющих шаблонов. Например, символ формата 36 × 36 имеет четыре области данных 16 × 16 которые, примыкая друг к другу, образуют координатную матрицу 32 × 32. Размер координатной матрицы для каждого формата символа приведен в таблице 7. Варианты размещения на границах приведены в таблице F.1;

b) второй знак символа размещают в верхней левой позиции, где его модули образуют последовательность битов (модулей) (рисунок F.11). Используют обозначение 2.1 для идентификации первого модуля второго знака символа. Этот модуль находится в верхней строке и в крайнем левом столбце каждой координатной матрицы. Последовательность массивов модулей, изображенная на рисунке F.7, является постоянной для всех координатных матриц.

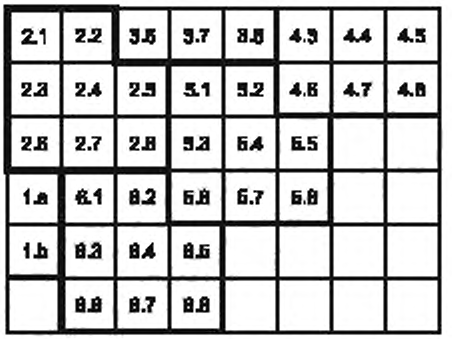


Рисунок F.7 — Начальная последовательность размещения модулей

Примечание — Величины a и b зависят от размера координатной матрицы.

c) размещение модулей в углах должно соответствовать таблице F.1 и соответствующим рисункам (от рисунка F.3 до рисунка F.6). Построение знаков символа типичной формы продолжают, сопрягая формы, как это показано выше для знаков символа 2, 5 и 6. Нетипичные знаки символа располагают в соответствии с таблицей F.1. Этот процесс дает в результате полное покрытие координатной матрицы знаками символа, большинство из которых не пронумерованы;

d) порядок следования знаков символа определен следующим образом. Знаки символа размещают вдоль параллельных диагональных линий с крайними точками левой нижней и правой верхней, наклоненных под углом 45° к границам символа, которые проходят через центры восьмых модулей знаков;

e) первую диагональную линию начинают как линию, проходящую через восьмой модуль первого знака символа, за исключением случая использования координатной матрицы размером 6 × 28, когда условие размещения в углах (рисунок F.6) определяет значения модулей в первом знаке символа (т.е. модуль, обозначенный на рисунке F.7 как 1.b, представляет модуль 1.2). Диагональную линию продолжают через модули 2.8 и 3.8;

f) в этой точке диагональная линия пересекает границу верхней строки. Следующую диагональную линию начинают с точки четвертого модуля справа от точки пересечения с верхней границей в верхней строке или, в случае использования координатной матрицы размером 8 × 8, в точке третьего модуля справа и одного модуля вниз, т.е. начало диагональной линии смещено вправо на четыре модуля. Знаки символа нумеруют по очередности их следования вдоль пути размещения, пересекающего восьмые модули. Таким образом, следующие знаки определены нисходящей диагональной линией, пересекающей модули 4.8, 5.8, 6.8 и так далее;

g) путь размещения знаков символа (рисунок F.8) продолжают по диагональным линиям, смещенным на четыре модуля вправо (или на четыре модуля вниз, или на комбинацию этих вариантов) от предыдущей диагональной линии. Первая и все нечетные диагональные линии отображают порядок следования знаков символа по направлению снизу вверх и слева направо. Вторая и все четные диагональные линии отображают порядок следования знаков символа по направлению сверху вниз и справа налево;

h) когда на пути размещения встречают условие для размещения знака символа нетипичной формы, который целиком не может быть размещен в границах координатной матрицы, часть этого знака символа продолжают на противоположной стороне матрицы, что приводит к необходимости нумерации противоположных частей таких знаков символа до того, как путь размещения пересечет отделенную часть. Например, в координатной матрице (рисунок F.8) отдельные части знаков символа три и семь пронумерованы до того, как путь размещения пересек их. Таким образом, вдоль пути размещения нумеруют только непронумерованные знаки символа.

Данные условия размещения на границах и в углах определены в таблице F.1, что подтверждается рисунком F.8 для знаков символа 1, 3, 4 и 7. Условия размещения в углах также влияют на порядок нумерации. Нумерацию нижнего левого угла:

- проводят (рисунок F.3) непосредственно перед знаком символа, расположенным над ним (примеры приведены на рисунках F.11 и F.18);
- проводят (рисунок F.4) непосредственно перед знаком символа, расположенным над ним (примеры приведены на рисунках F.12 и F.19);
- проводят (рисунок F.5) непосредственно после знака символа, расположенного справа от него (пример приведен на рисунке F.13);
- проводят (рисунок F.6) непосредственно перед знаком символа, расположенным над ним (пример приведен на рисунке F.15).

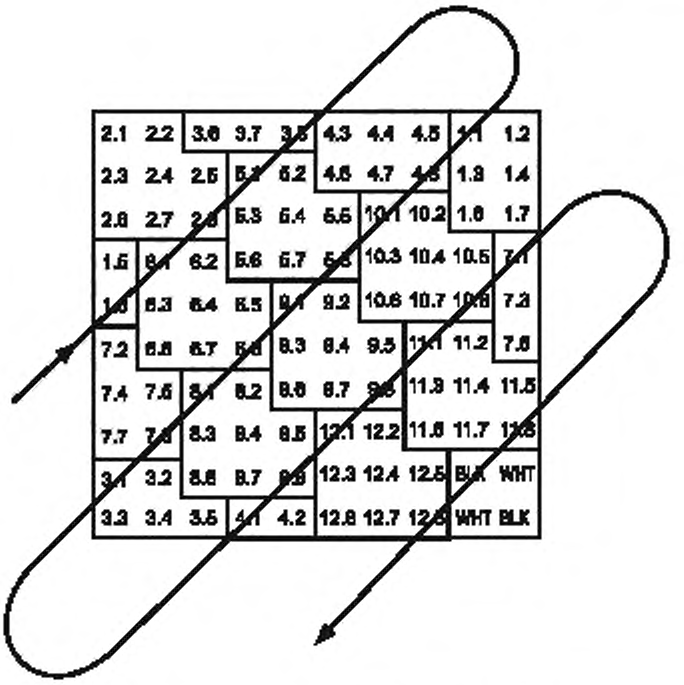


Рисунок F.8 — Последовательность размещения знаков символа

Остающиеся модули угла нумеруют перед тем, как путь размещения пересечет их.

i) процедуру размещения продолжают до тех пор, пока не будут размещены все знаки символа, и заканчивают в нижнем правом углу координатной матрицы. В координатных матрицах, имеющих четыре размера (10 × 10, 14 × 14, 18 × 18 и 22 × 22), остается область 2 × 2 в правом нижнем углу. Верхний левый и нижний правый модули этой области — темные (номинально кодирующие двоичную единицу) (рисунок F.8).

Типовые координатные матрицы, сформированные согласно этой процедуре, приведены в F.3. На рисунках F.9 — F.15 представлены варианты размещения на границах с 1 по 7 соответственно. На рисунках F.16 — F.19 представлены другие примеры для вариантов размещения с 1 по 4. Программа на языке программирования С, способная отображать все кодируемые биты в соответствующей координатной матрице, приведена в F.1.

F.3 Примеры размещения знаков символа в символах версии ECC 200

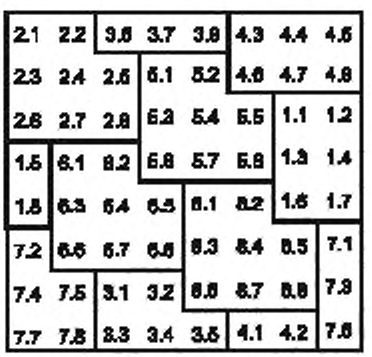


Рисунок F.9 — Размещение кодовых слов в квадратной координатной матрице размером 8 × 8

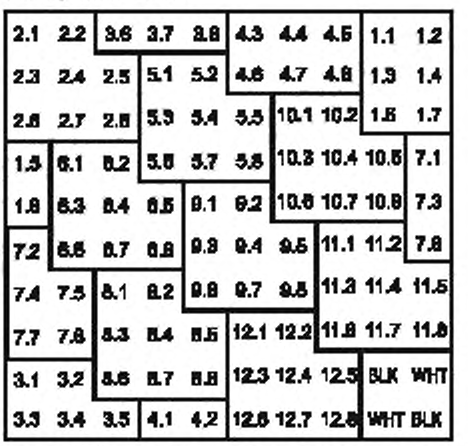


Рисунок F.10 — Размещение кодовых слов в квадратной координатной матрице размером 10 × 10

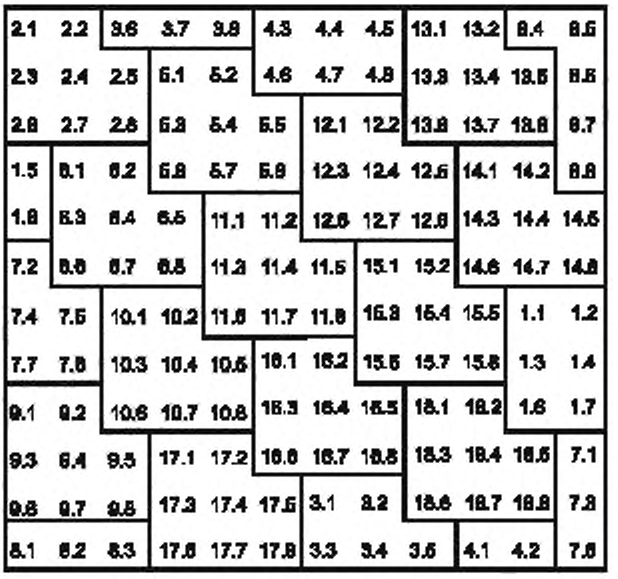


Рисунок F.11 — Размещение кодовых слов в квадратной координатной матрице размером 12 × 12

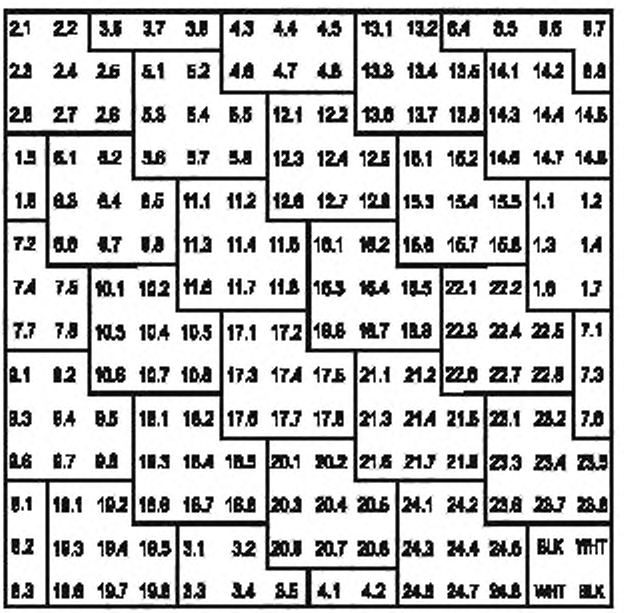


Рисунок F.12 — Размещение кодовых слов в квадратной координатной матрице размером 14 × 14

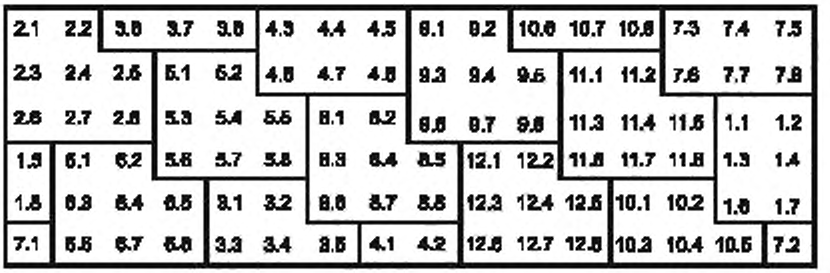


Рисунок F.13 — Размещение кодовых слов в прямоугольной координатной матрице размером 6 × 16

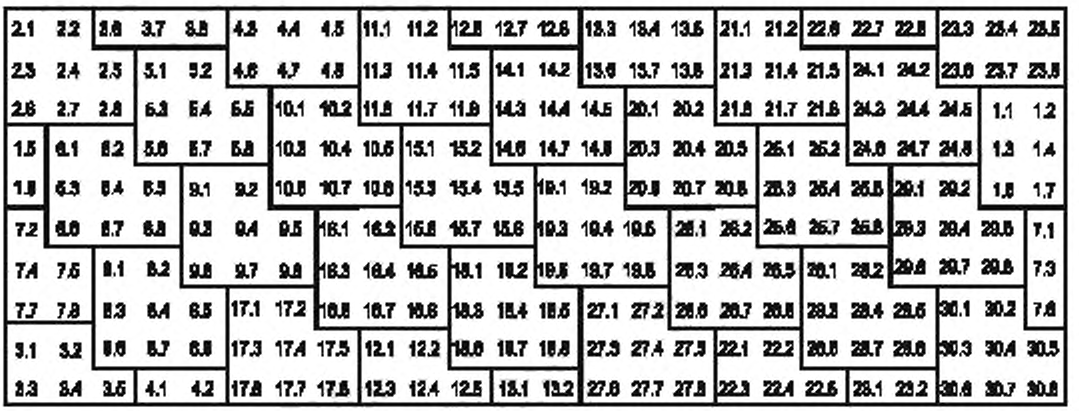


Рисунок F.14 — Размещение кодовых слов в прямоугольной координатной матрице размером 10 × 24



Рисунок F.15 — Размещение кодовых слов в прямоугольной координатной матрице размером 6 × 28

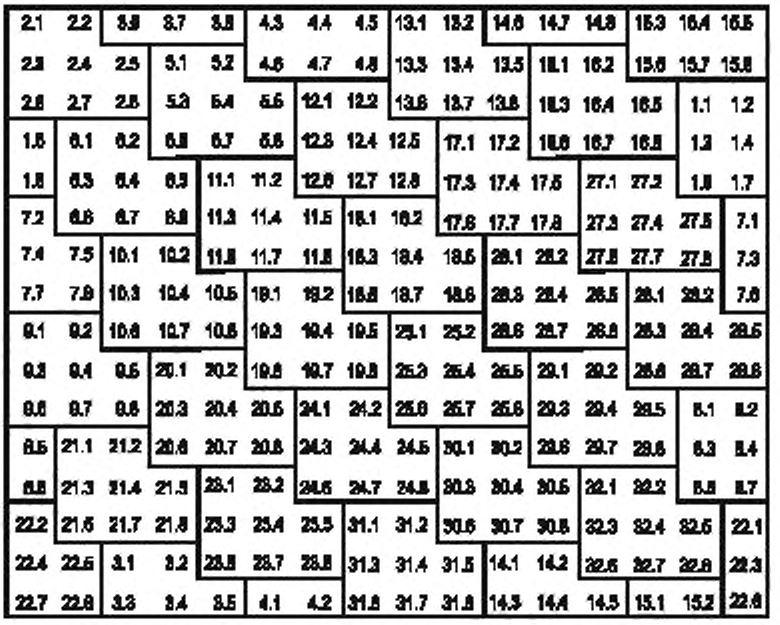


Рисунок F.16 — Размещение кодовых слов в квадратной координатной матрице размером 16 × 16

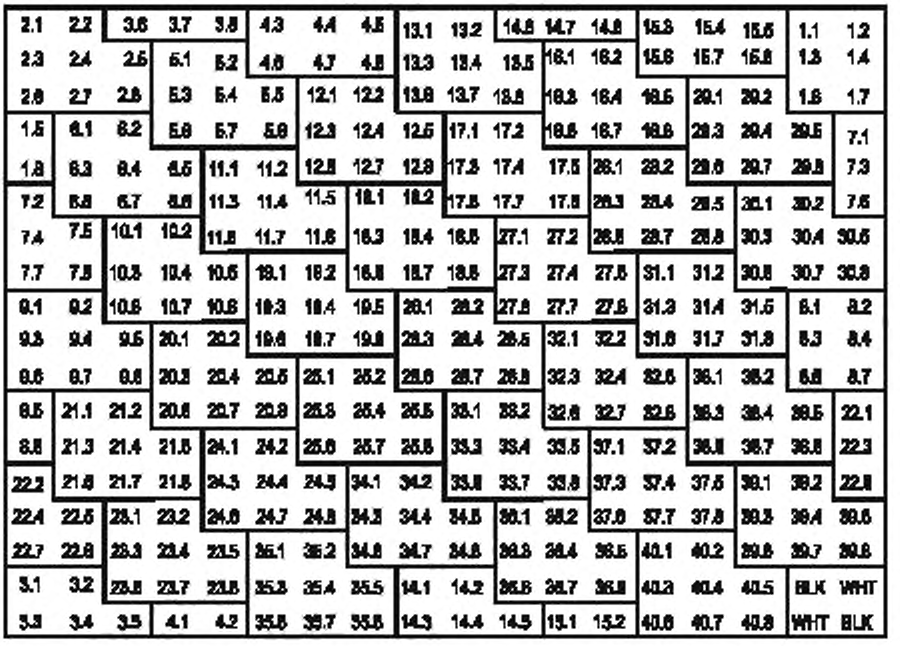


Рисунок F.17 — Размещение кодовых слов в квадратной координатной матрице размером 18 × 18

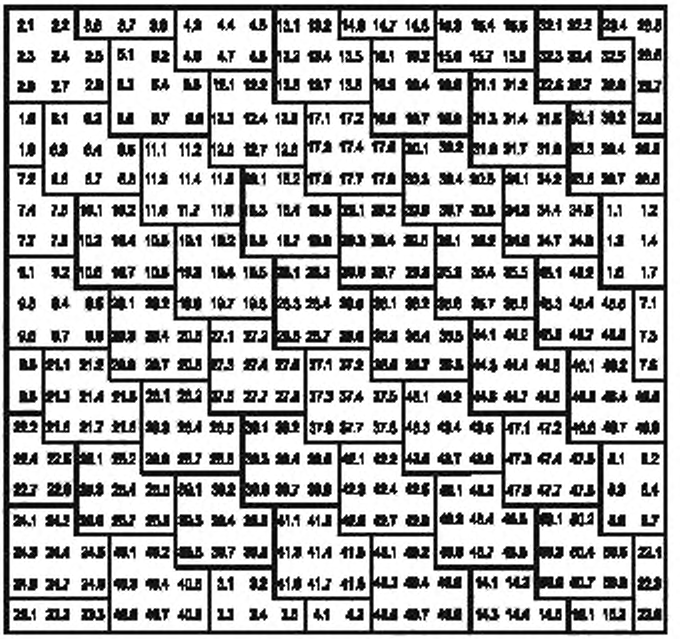


Рисунок F.18 — Размещение кодовых слов в квадратной координатной матрице размером 20 × 20

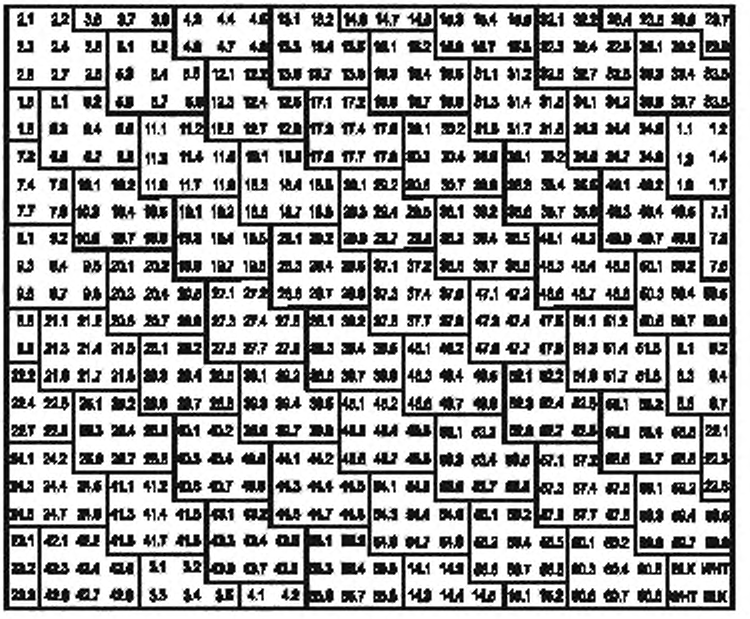


Рисунок F.19 — Размещение кодовых слов в квадратной координатной матрице размером 22 × 22

Приложение G
(обязательное)

Параметры символов версии ECC 000-140

Таблица G.1 — Параметры символа уровня ECC 000

Размер символа[a]		Размер области данных		Емкость символа			Содержание кодовых слов исправления ошибок, %	Возможность исправления %
Число строк	Число столбцов	Число строк	Число столбцов	в цифрах	в буквах и цифрах	в восьми разрядных байтах		
9	9	7	7	3	2	1	0,0	0,0
11	11	9	9	12	8	5	0,0	0,0
13	13	11	11	24	16	10	0,0	0,0
15	15	13	13	37	25	16	0,0	0,0
17	17	15	15	53	35	23	0,0	0,0
19	19	17	17	72	48	31	0,0	0,0
21	21	19	19	92	61	40	0,0	0,0
23	23	21	21	115	76	50	0,0	0,0
25	25	23	23	140	93	61	0,0	0,0
27	27	25	25	168	112	73	0,0	0,0
29	29	27	27	197	131	86	0,0	0,0
31	31	29	29	229	153	100	0,0	0,0
33	33	31	31	264	176	115	0,0	0,0
35	35	33	33	300	200	131	0,0	0,0
37	37	35	35	339	226	148	0,0	0,0
39	39	37	37	380	253	166	0,0	0,0
41	41	39	39	424	282	185	0,0	0,0
43	43	41	41	469	313	205	0,0	0,0
45	45	43	43	500	345	226	0,0	0,0
47	47	45	45	560	378	248	0,0	0,0
49	49	47	47	596	413	271	0,0	0,0

[a] Исключая свободные зоны.

Таблица G.2 — Параметры символа уровня ECC 050

Размер символа[a]		Размер области данных		Емкость символа			Содержание кодовых слов исправления ошибок, %	Возможность исправления, %
Число строк	Число столбцов	Число строк	Число столбцов	в цифрах	в буквах и цифрах	в восьми разрядных байтах		
11	11	9	9	1	1	0[b]	25,0	2,8
13	13	11	11	10	6	4	25,0	2,8
15	15	13	13	20	13	9	25,0	2,8
17	17	15	15	32	21	14	25,0	2,8
19	19	17	17	46	30	20	25,0	2,8
21	21	19	19	61	41	27	25,0	2,8
23	23	21	21	78	52	34	25,0	2,8
25	25	23	23	97	65	42	25,0	2,8
27	27	25	25	118	78	51	25,0	2,8
29	29	27	27	140	93	61	25,0	2,8
31	31	29	29	164	109	72	25,0	2,8
33	33	31	31	190	126	83	25,0	2,8
35	35	33	33	217	145	95	25,0	2,8
37	37	35	35	246	164	108	25,0	2,8
39	39	37	37	277	185	121	25,0	2,8
41	41	39	39	310	206	135	25,0	2,8
43	43	41	41	344	229	150	25,0	2,8
45	45	43	43	380	253	166	25,0	2,8
47	47	45	45	418	278	183	25,0	2,8
49	49	47	47	457	305	200	25,0	2,8

[a] Исключая свободные зоны.
[b] При данной комбинации строк и столбцов кодирование невозможно.

Таблица G.3 — Параметры символа уровня ECC 080

Размер символа[a]		Размер области данных		Емкость символа			Содержание кодовых слов исправления ошибок, %	Возможность исправления, %
Число строк	Число столбцов	Число строк	Число столбцов	в цифрах	в буквах и цифрах	в восьми разрядных байтах		
13	13	11	11	4	3	2	33,3	5,5
15	15	13	13	13	9	6	33,3	5,5
17	17	15	15	24	16	10	33,3	5,5
19	19	17	17	36	24	16	33,3	5,5
21	21	19	19	50	33	22	33,3	5,5
23	23	21	21	65	43	28	33,3	5,5
25	25	23	23	82	54	36	33,3	5,5
27	27	25	25	100	67	44	33,3	5,5
29	29	27	27	120	80	52	33,3	5,5
31	31	29	29	141	94	62	33,3	5,5
33	33	31	31	164	109	72	33,3	5,5
35	35	33	33	188	125	82	33,3	5,5
37	37	35	35	214	143	94	33,3	5,5
39	39	37	37	242	161	106	33,3	5,5
41	41	39	39	270	180	118	33,3	5,5
43	43	41	41	301	201	132	33,3	5,5
45	45	43	43	333	222	146	33,3	5,5
47	47	45	45	366	244	160	33,3	5,5
49	49	47	47	402	268	176	33,3	5,5

[a] Исключая свободные зоны.

Таблица G.4 — Параметры символа уровня ECC 100

Размер символа[a]		Размер области данных		Емкость символа			Содержание кодовых слов исправления ошибок, %	Возможность исправления, %
Число строк	Число столбцов	Число строк	Число столбцов	в цифрах	в буквах и цифрах	в восьми разрядных байтах		
13	13	11	11	1	1	0[b]	50,0	12,6
15	15	13	13	8	5	3	50,0	12,6
17	17	15	15	16	11	7	50,0	12,6
19	19	17	17	25	17	11	50,0	12,6
21	21	19	19	36	24	15	50,0	12,6
23	23	21	21	47	31	20	50,0	12,6
25	25	23	23	60	40	26	50,0	12,6
27	27	25	25	73	49	32	50,0	12,6
29	29	27	27	88	59	38	50,0	12,6
31	31	29	29	104	69	45	50,0	12,6
33	33	31	31	121	81	53	50,0	12,6
35	35	33	33	140	93	61	50,0	12,6
37	37	35	35	159	106	69	50,0	12,6
39	39	37	37	180	120	78	50,0	12,6
41	41	39	39	201	134	88	50,0	12,6
43	43	41	41	224	149	98	50,0	12,6
45	45	43	43	248	165	108	50,0	12,6
47	47	45	45	273	182	119	50,0	12,6
49	49	47	47	300	200	131	50,0	12,6

[a] Исключая свободные зоны.
[b] При данной комбинации строк и столбцов кодирование невозможно.

Таблица G.5 — Параметры символа уровня ECC-140

Размер символа[a]		Размер области данных		Емкость символа			Содержание кодовых слов исправления ошибок, %	Возможность исправления, %
Число строк	Число столбцов	Число строк	Число столбцов	в цифрах	в буквах и цифрах	в восьми разрядных байтах		
17	17	15	15	2	1	1	75,0	25,0
19	19	17	17	6	4	3	75,0	25,0
21	21	19	19	12	8	5	75,0	25,0
23	23	21	21	17	11	7	75,0	25,0
25	25	23	23	24	16	10	75,0	25,0
27	27	25	25	30	20	13	75,0	25,0
29	29	27	27	38	25	16	75,0	25,0
31	31	29	29	46	30	20	75,0	25,0
33	33	31	31	54	36	24	75,0	25,0
35	35	33	33	64	42	28	75,0	25,0
37	37	35	35	73	49	32	75,0	25,0
39	39	37	37	84	56	36	75,0	25,0
41	41	39	39	94	63	41	75,0	25,0
43	43	41	41	106	70	46	75,0	25,0
45	45	43	43	118	78	51	75,0	25,0
47	47	45	45	130	87	57	75,0	25,0
49	49	47	47	144	96	63	75,0	25,0

[a] Исключая свободные зоны.

Приложение Н
(обязательное)

Сетки размещения модулей данных для символов версии ECC 000-140[1), 2)]

Таблица Н.1 — Сетка размещения модулей данных для символа размером 7 × 7

2	45	10	38	24	21	1
12	40	26	5	33	19	47
22	31	29	15	43	8	36
34	20	48	13	41	27	6
44	9	37	23	17	30	16
39	25	4	32	18	46	11
0	28	14	42	7	35	3

Таблица Н.2 — Сетка размещения модулей данных для символа размером 9 × 9

2	19	55	10	46	28	64	73	1
62	17	53	35	71	8	80	44	26
49	31	67	4	76	40	22	58	13
69	6	78	42	24	60	15	51	33
74	38	20	56	11	47	29	65	37
25	61	16	52	34	70	7	79	43
12	48	30	66	63	75	39	21	57
32	68	5	77	41	23	59	14	50
0	72	36	18	54	9	45	27	3

Таблица Н.3 — Сетка размещения модулей данных для символа размером 11 × 11

2	26	114	70	15	103	59	37	81	4	1
117	73	18	106	62	40	84	7	95	51	29
12	100	56	34	78	92	89	45	23	111	67
65	43	87	10	98	54	32	120	76	21	109
82	5	93	49	27	115	71	16	104	60	38
96	52	30	118	74	19	107	63	41	85	8
24	112	68	13	101	57	35	79	48	90	46
75	20	108	64	42	86	9	97	53	31	119
102	58	36	80	77	91	47	25	113	69	14
39	83	6	94	50	28	116	72	17	105	61
0	88	44	22	110	66	11	99	55	33	3

Таблица Н.4 — Сетка размещения модулей данных для символа размером 13 × 13

2	159	29	133	81	16	120	68	42	146	94	91	1
37	141	89	24	128	76	50	154	102	11	115	63	167
83	18	122	70	44	148	96	5	109	57	161	31	135
125	73	47	151	99	8	112	60	164	34	138	86	21
40	144	92	107	105	53	157	27	131	79	14	118	66
103	12	116	64	168	38	142	90	25	129	77	51	155
110	58	162	32	136	84	19	123	71	45	149	97	6
165	35	139	87	22	126	74	48	152	100	9	113	61
132	80	15	119	67	41	145	93	55	106	54	158	28
23	127	75	49	153	101	10	114	52	166	36	140	88
69	43	147	95	4	108	56	160	30	134	82	17	121
150	98	7	111	59	163	33	137	85	20	124	72	46
0	104	52	156	26	130	78	13	117	65	39	143	3

[1)] *В сетках приведены порядковые номера модулей рандомизированного двоичного потока данных.*

[2)] Таблицы Н13 — Н21 настоящего приложения приведены на вкладках к данному стандарту в связи с большим объемом информации.

Таблица Н.5 — Сетка размещения модулей данных для символа размером 15 × 15

2	187	37	157	97	217	22	142	82	202	52	172	112	7	1
41	161	101	221	26	146	86	206	56	176	116	11	131	71	191
93	213	18	138	78	198	48	168	108	105	123	63	183	33	153
28	148	88	208	58	178	118	13	133	73	193	43	163	103	223
80	200	50	170	110	5	125	65	185	35	155	95	215	20	140
54	174	114	9	129	69	189	39	159	99	219	24	144	84	204
106	127	121	61	181	31	151	91	211	16	136	76	196	46	166
134	74	194	44	164	104	224	29	149	89	209	59	179	119	14
186	36	156	96	216	21	141	81	201	51	171	111	6	126	66
160	100	220	25	145	85	205	55	175	115	10	130	70	190	40
212	17	137	77	197	47	167	107	67	122	62	182	32	152	92
147	87	207	57	177	117	12	132	72	192	42	162	102	222	27
199	49	169	109	4	124	64	184	34	154	94	214	19	139	79
173	113	8	128	68	188	38	158	98	218	23	143	83	203	53
0	120	60	180	30	150	90	210	15	135	75	195	45	165	3

Таблица Н.6 — Сетка размещения модулей данных для символа размером 17 × 17

2	69	205	35	171	103	239	18	154	86	222	52	188	120	256	273	1
220	50	186	118	254	33	169	101	237	67	203	135	271	16	288	152	84
178	110	246	25	161	93	229	59	195	127	263	8	280	144	76	212	42
250	29	165	97	233	63	199	131	267	12	284	148	80	216	46	182	114
157	89	225	55	191	123	259	4	276	140	72	208	38	174	106	242	21
235	65	201	133	269	14	286	150	82	218	48	184	116	252	31	167	99
193	125	261	6	278	142	74	210	40	176	108	244	23	159	91	227	57
265	10	282	146	78	214	44	180	112	248	27	163	95	231	61	197	129
274	138	70	206	36	172	104	240	19	155	87	223	53	189	121	257	137
83	219	49	185	117	253	32	168	100	236	66	202	134	270	15	287	151
41	177	109	245	24	160	92	228	58	194	126	262	7	279	143	75	211
113	249	28	164	96	232	62	198	130	266	11	283	147	79	215	45	181
20	156	88	224	54	190	122	258	255	275	139	71	207	37	173	105	241
98	234	64	200	132	268	13	285	149	81	217	47	183	115	251	30	166
56	192	124	260	5	277	141	73	209	39	175	107	243	22	158	90	226
128	264	9	281	145	77	213	43	179	111	247	26	162	94	230	60	196
0	272	136	68	204	34	170	102	238	17	153	85	221	51	187	119	3

Таблица Н.7 — Сетка размещения модулей данных для символа размером 19 × 19

2	82	234	44	348	196	120	272	25	329	177	101	253	63	215	139	291	6	1
239	49	353	201	125	277	30	334	182	106	258	68	220	144	296	11	315	163	87
343	191	115	267	20	324	172	96	248	58	210	134	286	310	305	153	77	229	39
132	284	37	341	189	113	265	75	227	151	303	18	322	170	94	246	56	360	208
28	332	180	104	256	66	218	142	294	9	313	161	85	237	47	351	199	123	275
185	109	261	71	223	147	299	14	318	166	90	242	52	356	204	128	280	33	337
251	61	213	137	289	4	308	156	80	232	42	346	194	118	270	23	327	175	99
225	149	301	16	320	168	92	244	54	358	206	130	282	35	339	187	111	263	73
292	7	311	159	83	235	45	349	197	121	273	26	330	178	102	254	64	216	140
316	164	88	240	50	354	202	126	278	31	335	183	107	259	69	221	145	297	12
78	230	40	344	192	116	268	21	325	173	97	249	59	211	135	287	158	306	154
55	359	207	131	283	36	340	188	112	264	74	226	150	302	17	321	169	93	245
198	122	274	27	331	179	103	255	65	217	141	293	8	312	160	84	236	46	350
279	32	336	184	108	260	70	222	146	298	13	317	165	89	241	51	355	203	127
326	174	98	250	60	212	136	288	285	307	155	79	231	41	345	193	117	269	22
110	262	72	224	148	300	15	319	167	91	243	53	357	205	129	281	34	338	186
62	214	138	290	5	309	157	81	233	43	347	195	119	271	24	328	176	100	252
143	295	10	314	162	86	238	48	352	200	124	276	29	333	181	105	257	67	219
0	304	152	76	228	38	342	190	114	266	19	323	171	95	247	57	209	133	3

Таблица Н.8 — Сетка размещения модулей данных для символа размером 21 × 21

2	88	424	256	46	382	214	130	298	25	361	193	109	277	67	403	235	151	319	4	1
437	269	59	395	227	143	311	38	374	206	122	290	80	416	248	164	332	17	353	185	101
49	385	217	133	301	28	364	196	112	280	70	406	238	154	322	7	343	175	91	427	259
222	138	306	33	369	201	117	285	75	411	243	159	327	12	348	180	96	432	264	54	390
295	22	358	190	106	274	64	400	232	148	316	340	337	169	85	421	253	43	379	211	127
377	209	125	293	83	419	251	167	335	20	356	188	104	440	272	62	398	230	146	314	41
115	283	73	409	241	157	325	10	346	178	94	430	262	52	388	220	136	304	31	367	199
78	414	246	162	330	15	351	183	99	435	267	57	393	225	141	309	36	372	204	120	288
236	152	320	5	341	173	89	425	257	47	383	215	131	299	26	362	194	110	278	68	404
333	18	354	186	102	438	270	60	396	228	144	312	39	375	207	123	291	81	417	249	165
344	176	92	428	260	50	386	218	134	302	29	365	197	113	281	71	407	239	155	323	8
97	433	265	55	391	223	139	307	34	370	202	118	286	76	412	244	160	328	13	349	181
254	44	380	212	128	296	23	359	191	107	275	65	401	233	149	317	172	338	170	86	422
397	229	145	313	40	376	208	124	292	82	418	250	166	334	19	355	187	103	439	271	61
135	303	30	366	198	114	282	72	408	240	156	324	9	345	177	93	429	261	51	387	219
35	371	203	119	287	77	413	245	161	329	14	350	182	98	434	266	56	392	224	140	308
192	108	276	66	402	234	150	318	315	339	171	87	423	255	45	381	213	129	297	24	360
289	79	415	247	163	331	16	352	184	100	436	268	58	394	226	142	310	37	373	205	121
405	237	153	321	6	342	174	90	426	258	48	384	216	132	300	27	363	195	111	279	69
158	326	11	347	179	95	431	263	53	389	221	137	305	32	368	200	116	284	74	410	242
0	336	168	84	420	252	42	378	210	126	294	21	357	189	105	273	63	399	231	147	3

Таблица Н.9 — Сетка размещения модулей данных для символа размером 23 × 23

2	102	470	286	56	424	240	148	516	332	33	401	217	125	493	309	79	447	263	171	355	10	1
476	292	62	430	246	154	522	338	39	407	223	131	499	315	85	453	269	177	361	16	384	200	108
50	418	234	142	510	326	27	395	211	119	487	303	73	441	257	165	349	4	372	188	96	464	280
249	157	525	341	42	410	226	134	502	318	88	456	272	180	364	19	387	203	111	479	295	65	433
513	329	30	398	214	122	490	306	76	444	260	168	352	7	375	191	99	467	283	53	421	237	145
36	404	220	128	496	312	82	450	266	174	358	13	381	197	105	473	289	59	427	243	151	519	335
208	116	484	300	70	438	254	162	346	378	369	185	93	461	277	47	415	231	139	507	323	24	392
505	321	91	459	275	183	367	22	390	206	114	482	298	68	436	252	160	528	344	45	413	229	137
80	448	264	172	356	11	379	195	103	471	287	57	425	241	149	517	333	34	402	218	126	494	310
270	178	362	17	385	201	109	477	293	63	431	247	155	523	339	40	408	224	132	500	316	86	454
350	5	373	189	97	465	281	51	419	235	143	511	327	28	396	212	120	488	304	74	442	258	166
388	204	112	480	296	66	434	250	158	526	342	43	411	227	135	503	319	89	457	273	181	365	20
100	468	284	54	422	238	146	514	330	31	399	215	123	491	307	77	445	261	169	353	8	376	192
290	60	428	244	152	520	336	37	405	221	129	497	313	83	451	267	175	359	14	382	198	106	474
416	232	140	508	324	25	393	209	117	485	301	71	439	255	163	347	194	370	186	94	462	278	48
159	527	343	44	412	228	136	504	320	90	458	274	182	366	21	389	205	113	481	297	67	435	251
331	32	400	216	124	492	308	78	446	262	170	354	9	377	193	101	469	285	55	423	239	147	515
406	222	130	498	314	84	452	268	176	360	15	383	199	107	475	291	61	429	245	153	521	337	38
118	486	302	72	440	256	164	348	345	371	187	95	463	279	49	417	233	141	509	325	26	394	210
317	87	455	271	179	363	18	386	202	110	478	294	64	432	248	156	524	340	41	409	225	133	501
443	259	167	351	6	374	190	98	466	282	52	420	236	144	512	328	29	397	213	121	489	305	75
173	357	12	380	196	104	472	288	58	426	242	150	518	334	35	403	219	127	495	311	81	449	265
0	368	184	92	460	276	46	414	230	138	506	322	23	391	207	115	483	299	69	437	253	161	3

Таблица Н.10 — Сетка размещения модулей данных для символа размером 25 × 25

2	603	103	503	303	53	453	253	153	553	353	28	428	228	128	528	328	78	478	278	178	578	378	375	1
123	523	323	73	473	273	173	573	373	48	448	248	148	548	348	98	498	298	198	598	398	23	423	223	623
311	61	461	261	161	561	361	36	436	236	136	536	336	86	486	286	186	586	386	11	411	211	611	111	511
467	267	167	567	367	42	442	242	142	542	342	92	492	292	192	592	392	17	417	217	617	117	517	317	67
155	555	355	30	430	230	130	530	330	80	480	280	180	580	380	5	405	205	605	105	505	305	55	455	255
370	45	445	245	145	545	345	95	495	295	195	595	395	20	420	220	620	120	520	320	70	470	270	170	570
433	233	133	533	333	83	483	283	183	583	383	8	408	208	608	108	508	308	58	458	258	158	558	358	33
139	539	339	89	489	289	189	589	389	14	414	214	614	114	514	314	64	464	264	164	564	364	39	439	239
326	76	476	276	176	576	376	403	401	201	601	101	501	301	51	451	251	151	551	351	26	426	226	126	526
499	299	199	599	399	24	424	224	624	124	524	324	74	474	274	174	574	374	49	449	249	149	549	349	99
187	587	387	12	412	212	612	112	512	312	62	462	262	162	562	362	37	437	237	137	537	337	87	487	287
393	18	418	218	618	118	518	318	68	468	268	168	568	368	43	443	243	143	543	343	93	493	293	193	593
406	206	606	106	506	306	56	456	256	156	556	356	31	431	231	131	531	331	81	481	281	181	581	381	6
621	121	521	321	71	471	271	171	571	371	46	446	246	146	546	346	96	496	296	196	596	396	21	421	221
509	309	59	459	259	159	559	359	34	434	234	134	534	334	84	484	284	184	584	384	9	409	209	609	109
65	465	265	165	565	365	40	440	240	140	540	340	90	490	290	190	590	390	15	415	215	615	115	515	315
252	152	552	352	27	427	227	127	527	327	77	477	277	177	577	377	203	402	202	602	102	502	302	52	452
572	372	47	447	247	147	547	347	97	497	297	197	597	397	22	422	222	622	122	522	322	72	472	272	172
35	435	235	135	535	335	85	485	285	185	585	385	10	410	210	610	110	510	310	60	460	260	160	560	360
241	141	541	341	91	491	291	191	591	391	16	416	216	616	116	516	316	66	466	266	166	566	366	41	441
529	329	79	479	279	179	579	379	4	404	204	604	104	504	304	54	454	254	154	554	354	29	429	229	129
94	494	294	194	594	394	19	419	219	619	119	519	319	69	469	269	169	569	369	44	444	244	144	544	344
282	182	582	382	7	407	207	607	107	507	307	57	457	257	157	557	357	32	432	232	132	532	332	82	482
588	388	13	413	213	613	113	513	313	63	463	263	163	563	363	38	438	238	138	538	338	88	488	288	188
0	400	200	600	100	500	300	50	450	250	150	550	350	25	425	225	125	525	325	75	475	275	175	575	3

Таблица Н.11 — Сетка размещения модулей данных для символа размером 27 × 27

2	658	118	550	334	64	496	280	712	172	604	388	37	469	253	685	145	577	361	91	523	307	199	631	415	10	1
125	557	341	71	503	287	719	179	611	395	44	476	260	692	152	584	368	98	530	314	206	638	422	17	449	233	665
327	57	489	273	705	165	597	381	30	462	246	678	138	570	354	84	516	300	192	624	408	405	435	219	651	111	543
511	295	727	187	619	403	52	484	268	700	160	592	376	106	538	322	214	646	430	25	457	241	673	133	565	349	79
714	174	606	390	39	471	255	687	147	579	363	93	525	309	201	633	417	12	444	228	660	120	552	336	66	498	282
613	397	46	478	262	694	154	586	370	100	532	316	208	640	424	19	451	235	667	127	559	343	73	505	289	721	181
32	464	248	680	140	572	356	86	518	302	194	626	410	5	437	221	653	113	545	329	59	491	275	707	167	599	383
265	697	157	589	373	103	535	319	211	643	427	22	454	238	670	130	562	346	76	508	292	724	184	616	400	49	481
143	575	359	89	521	305	197	629	413	8	440	224	656	116	548	332	62	494	278	710	170	602	386	35	467	251	683
366	96	528	312	204	636	420	15	447	231	663	123	555	339	69	501	285	717	177	609	393	42	474	258	690	150	582
514	298	190	622	406	442	433	217	649	109	541	325	55	487	271	703	163	595	379	28	460	244	676	136	568	352	82
215	647	431	26	458	242	674	134	566	350	80	512	296	728	188	620	404	53	485	269	701	161	593	377	107	539	323
418	13	445	229	661	121	553	337	67	499	283	715	175	607	391	40	472	256	688	148	580	364	94	526	310	202	634
452	236	668	128	560	344	74	506	290	722	182	614	398	47	479	263	695	155	587	371	101	533	317	209	641	425	20
654	114	546	330	60	492	276	708	168	600	384	33	465	249	681	141	573	357	87	519	303	195	627	411	6	438	222
563	347	77	509	293	725	185	617	401	50	482	266	698	158	590	374	104	536	320	212	644	428	23	455	239	671	131
63	495	279	711	171	603	387	36	468	252	684	144	576	360	90	522	306	198	630	414	9	441	225	657	117	549	333
286	718	178	610	394	43	475	259	691	151	583	367	97	529	313	205	637	421	16	448	232	664	124	556	340	70	502
164	596	380	29	461	245	677	137	569	353	83	515	299	191	623	407	226	434	218	650	110	542	326	56	488	272	704
402	51	483	267	699	159	591	375	105	537	321	213	645	429	24	456	240	672	132	564	348	78	510	294	726	186	618
470	254	686	146	578	362	92	524	308	200	632	416	11	443	227	659	119	551	335	65	497	281	713	173	605	389	38
693	153	585	369	99	531	315	207	639	423	18	450	234	666	126	558	342	72	504	288	720	180	612	396	45	477	261
571	355	85	517	301	193	625	409	4	436	220	652	112	544	328	58	490	274	706	166	598	382	31	463	247	679	139
102	534	318	210	642	426	21	453	237	669	129	561	345	75	507	291	723	183	615	399	48	480	264	696	156	588	372
304	196	628	412	7	439	223	655	115	547	331	61	493	277	709	169	601	385	34	466	250	682	142	574	358	88	520
635	419	14	446	230	662	122	554	338	68	500	284	716	176	608	392	41	473	257	689	149	581	365	95	527	311	203
0	432	216	648	108	540	324	54	486	270	702	162	594	378	27	459	243	675	135	567	351	81	513	297	189	621	3

Таблица Н.12 — Сетка размещения модулей данных для символа размером 29 × 29

2	703	123	587	355	819	65	529	297	761	181	645	413	36	500	268	732	152	616	384	94	558	326	790	210	674	442	7	1
141	605	373	837	83	547	315	779	199	663	431	54	518	286	750	170	634	402	112	576	344	808	228	692	460	25	489	257	721
359	823	69	533	301	765	185	649	417	40	504	272	736	156	620	388	98	562	330	794	214	678	446	11	475	243	707	127	591
76	540	308	772	192	656	424	47	511	279	743	163	627	395	105	569	337	801	221	685	453	18	482	250	714	134	598	366	830
293	757	177	641	409	32	496	264	728	148	612	380	90	554	322	786	206	670	438	435	467	235	699	119	583	351	815	61	525
201	665	433	56	520	288	752	172	636	404	114	578	346	810	230	694	462	27	491	259	723	143	607	375	839	85	549	317	781
419	42	506	274	738	158	622	390	100	564	332	796	216	680	448	13	477	245	709	129	593	361	825	71	535	303	767	187	651
513	281	745	165	629	397	107	571	339	803	223	687	455	20	484	252	716	136	600	368	832	78	542	310	774	194	658	426	49
730	150	614	382	92	556	324	788	208	672	440	5	469	237	701	121	585	353	817	63	527	295	759	179	643	411	34	498	266
632	400	110	574	342	806	226	690	458	23	487	255	719	139	603	371	835	81	545	313	777	197	661	429	52	516	284	748	168
96	560	328	792	212	676	444	9	473	241	705	125	589	357	821	67	531	299	763	183	647	415	38	502	270	734	154	618	386
335	799	219	683	451	16	480	248	712	132	596	364	828	74	538	306	770	190	654	422	45	509	277	741	161	625	393	103	567
204	668	436	471	465	233	697	117	581	349	813	59	523	291	755	175	639	407	30	494	262	726	146	610	378	88	552	320	784
463	28	492	260	724	144	608	376	840	86	550	318	782	202	666	434	57	521	289	753	173	637	405	115	579	347	811	231	695
478	246	710	130	594	362	826	72	536	304	768	188	652	420	43	507	275	739	159	623	391	101	565	333	797	217	681	449	14
717	137	601	369	833	79	543	311	775	195	659	427	50	514	282	746	166	630	398	108	572	340	804	224	688	456	21	485	253
586	354	818	64	528	296	760	180	644	412	35	499	267	731	151	615	383	93	557	325	789	209	673	441	6	470	238	702	122
836	82	546	314	778	198	662	430	53	517	285	749	169	633	401	111	575	343	807	227	691	459	24	488	256	720	140	604	372
532	300	764	184	648	416	39	503	271	735	155	619	387	97	561	329	793	213	677	445	10	474	242	706	126	590	358	822	68
771	191	655	423	46	510	278	742	162	626	394	104	568	336	800	220	684	452	17	481	249	713	133	597	365	829	75	539	307
640	408	31	495	263	727	147	611	379	89	553	321	785	205	669	437	239	466	234	698	118	582	350	814	60	524	292	756	176
55	519	287	751	171	635	403	113	577	345	809	229	693	461	26	490	258	722	142	606	374	838	84	548	316	780	200	664	432
273	737	157	621	389	99	563	331	795	215	679	447	12	476	244	708	128	592	360	824	70	534	302	766	186	650	418	41	505
164	628	396	106	570	338	802	222	686	454	19	483	251	715	135	599	367	831	77	541	309	773	193	657	425	48	512	280	744
381	91	555	323	787	207	671	439	4	468	236	700	120	584	352	816	62	526	294	758	178	642	410	33	497	265	729	149	613
573	341	805	225	689	457	22	486	254	718	138	602	370	834	80	544	312	776	196	660	428	51	515	283	747	167	631	399	109
791	211	675	443	8	472	240	704	124	588	356	820	66	530	298	762	182	646	414	37	501	269	733	153	617	385	95	559	327
682	450	15	479	247	711	131	595	363	827	73	537	305	769	189	653	421	44	508	276	740	160	624	392	102	566	334	798	218
0	464	232	696	116	580	348	812	58	522	290	754	174	638	406	29	493	261	725	145	609	377	87	551	319	783	203	667	3

Приложение I
(обязательное)

Схемы кодирования знаков символики версии ECC 000-140

В настоящем приложении приведены описание кодируемого набора знаков ASCII (КОИ-7) (национальная версия США — ИСО/МЭК 646), применяемого в одной из схем кодирования, используемой в символике версии ECC 000-140, а также соответствие десятичных значений знаков указанной схемы кодирования знакам данных четырех других схем кодирования.

Таблица I.1 — Соответствие значений знаков данных для различных схем кодирования

Знак данных схемы кодирования ASCII (КОИ-7)		Значение знака в схеме кодирования			
Знак	Десятичное значение	по основанию 11	по основанию 27	по основанию 37	по основанию 41
NUL	0	—	—	—	—
SOH	1	—	—	—	—
STX	2	—	—	—	—
ETX	3	—	—	—	—
EOT	4	—	—	—	—
ENQ	5	—	—	—	—
ACK	6	—	—	—	—
BEL	7	—	—	—	—
BS	8	—	—	—	—
HT	9	—	—	—	—
LF	10	—	—	—	—
VT	11	—	—	—	—
FF	12	—	—	—	—
CR	13	—	—	—	—
SO	14	—	—	—	—
SI	15	—	—	—	—
DLE	16	—	—	—	—
DC1	17	—	—	—	—
DC2	18	—	—	—	—
DC3	19	—	—	—	—
DC4	20	—	—	—	—
NAK	21	—	—	—	—
SYN	22	—	—	—	—
ETB	23	—	—	—	—
CAN	24	—	—	—	—
EM	25	—	—	—	—
SUB	26	—	—	—	—
ESC	27	—	—	—	—

Продолжение таблицы I.1

Знак данных схемы кодирования ASCII (КОИ-7)		Значение знака в схеме кодирования			
Знак	Десятичное значение	по основанию 11	по основанию 27	по основанию 37	по основанию 41
FS	28	—	—	—	—
GS	29	—	—	—	—
RS	30	—	—	—	—
US	31	—	—	—	—
ПРОБЕЛ (Space)[1)]	32	0	0	0	0
!	33	—	—	—	—
"	34	—	—	—	—
#	35	—	—	—	—
$	36	—	—	—	—
%	37	—	—	—	—
&	38	—	—	—	—
'	39	—	—	—	—
(	40	—	—	—	—
)	41	—	—	—	—
*	42	—	—	—	—
+	43	—	—	—	—
,	44	—	—	—	38
-	45	—	—	—	39
.	46	—	—	—	37
/	47	—	—	—	40
0	48	1	—	27	27
1	49	2	—	28	28
2	50	3	—	29	29
3	51	4	—	30	30
4	52	5	—	31	31
5	53	6	—	32	32
6	54	7	—	33	33
7	55	8	—	34	34
8	56	9	—	35	35
9	57	10	—	36	36
:	58	—	—	—	—
;	59	—	—	—	—
<	60	—	—	—	—

[1)] *Приведено наименование знака, используемое в настоящем стандарте, в скобках указано обозначение, применяемое в ИСО/МЭК 16022.*

Продолжение таблицы I.1

Знак данных схемы кодирования ASCII (КОИ-7)		Значение знака в схеме кодирования			
Знак	Десятичное значение	по основанию 11	по основанию 27	по основанию 37	по основанию 41
=	61	—	—	—	—
>	62	—	—	—	—
?	63	—	—	—	—
@	64	—	—	—	—
A	65	—	1	1	1
B	66	—	2	2	2
C	67	—	3	3	3
D	68	—	4	4	4
E	69	—	5	5	5
F	70	—	6	6	6
G	71	—	7	7	7
H	72	—	8	8	8
I	73	—	9	9	9
J	74	—	10	10	10
K	75	—	11	11	11
L	76	—	12	12	12
M	77	—	13	13	13
N	78	—	14	14	14
O	79	—	15	15	15
P	80	—	16	16	16
Q	81	—	17	17	17
R	82	—	18	18	18
S	83	—	19	19	19
T	84	—	20	20	20
U	85	—	21	21	21
V	86	—	22	22	22
W	87	—	23	23	23
X	88	—	24	24	24
Y	89	—	25	25	25
Z	90	—	26	26	26
[	91	—	—	—	—
\	92	—	—	—	—
]	93	—	—	—	—
^	94	—	—	—	—
-	95	—	—	—	—
`	96	—	—	—	—

Продолжение таблицы I.1

Знак данных схемы кодирования ASCII (КОИ-7)		Значение знака в схеме кодирования			
Знак	Десятичное значение	по основанию 11	по основанию 27	по основанию 37	по основанию 41
a	97	—	—	—	—
b	98	—	—	—	—
c	99	—	—	—	—
d	100	—	—	—	—
e	101	—	—	—	—
f	102	—	—	—	—
g	103	—	—	—	—
h	104	—	—	—	—
i	105	—	—	—	—
j	106	—	—	—	—
k	107	—	—	—	—
l	108	—	—	—	—
m	109	—	—	—	—
n	110	—	—	—	—
o	111	—	—	—	—
p	112	—	—	—	—
q	113	—	—	—	—
r	114	—	—	—	—
s	115	—	—	—	—
t	116	—	—	—	—
u	117	—	—	—	—
v	118	—	—	—	—
w	119	—	—	—	—
x	120	—	—	—	—
y	121	—	—	—	—
z	122	—	—	—	—
{	123	—	—	—	—
|	124	—	—	—	—
}	125	—	—	—	—
~	126	—	—	—	—
DEL	127	—	—	—	—

I.1 Схема кодирования по основанию 11

I.1.1 Процедура первого этапа

Исходные знаки данных должны быть преобразованы в соответствующие значения схемы кодирования по основанию 11, используя таблицу I.1 как переходную таблицу.

I.1.2 Процедура второго этапа

Уплотнение последовательности значений знаков схемы кодирования по основанию 11 в двоичную строку проводят следующим образом:

a) разбивают слева направо последовательность значений знаков схемы кодирования по основанию 11 на группы по шесть значений в каждой. Если в группе менее шести значений, переходят к этапу, указанному в перечислении e)[1];

b) присваивают обозначения шести значениям знаков по основанию 11 в группе как $C_1,...,C_6$, где C_1 — значение первого знака;

c) проводят преобразование от основания 11 к основанию 2, получая последовательность, состоящую из 21 бита с помощью уравнения для числа знаков 6 в таблице I.2;

d) повторяют действия, начиная с этапа, указанного в перечислении a) (при необходимости);

e) если число значений знаков в группе менее шести, проводят преобразование от основания 11 к основанию 2, используя уравнение из таблицы I.2 соответствующее числу оставшихся значений знаков схемы кодирования по основанию 11.

Таблица I.2 — Уравнения кодирования знаков схемы кодирования по основанию 11 (цифр)

Число знаков данных	Уравнение кодирования	Длина, биты
1	C_1	4
2	$C_1 + C_2 \times 11$	7
3	$C_1 + C_2 \times 11 + C_3 \times 11^2$	11
4	$C_1 + C_2 \times 11 + C_3 \times 11^2 + C_4 \times 11^3$	14
5	$C_1 + C_2 \times 11 + C_3 \times 11^2 + C_4 \times 11^3 + C_5 \times 11^4$	18
6	$C_1 + C_2 \times 11 + C_3 \times 11^2 + C_4 \times 11^3 + C_5 \times 11^4 + C_6 \times 11^5$	21

I.1.3 Пример

Процесс кодирования по основанию 11 строки знаков 123<ПРОБЕЛ>45678 приведен на рисунке I.1.

Данные	**1**	**2**	**3**	**ПРОБЕЛ**	**4**	**5**	**6**	**7**	**8**
Значение в схеме кодирования по основанию 11	2	3	4	0	5	6	7	8	9
Позиция знака	C_1	C_2	C_3	C_4	C_5	C_6	C_1	C_2	C_3
Весовой коэффициент	1	11	121	1331	14641	161051	1	11	121
Результат произведения	2	33	484	0	73205	966306	7	88	1089
Десятичное значение	1040030						1184		
Двоичная строка	01111110111010011110						10010100000		

Рисунок I.1 — Пример кодирования по основанию 11

I.2 Схема кодирования по основанию 27

I.2.1 Процедура первого этапа

Исходные знаки данных должны быть преобразованы в соответствующие значения схемы кодирования по основанию 27, используя таблицу I.1 как переходную таблицу.

I.2.2 Процедура второго этапа

Уплотнение последовательности значений знаков схемы кодирования по основанию 27 в двоичную строку проводят следующим образом:

a) разбивают слева направо последовательность значений знаков схемы кодирования по основанию 27 на группы по пять значений в каждой. Если в группе менее пяти значений, то переходят к этапу, указанному в перечислении e)[1];

---

[1] *В оригинале ИСО/МЭК 16022 ошибочно указано: «...переходят к этапу 5».*

b) присваивают обозначения пяти значениям знаков в группе как $C_1,...,C_5$, где C_1 — значение первого знака;

c) проводят преобразование от основания 27 к основанию 2, получая последовательность, состоящую из 24 битов с помощью уравнения для числа знаков 5 в таблице I.3;

d) повторяют действия, начиная с этапа, указанного в перечислении a) (при необходимости);

e) если число значений знаков в группе менее пяти, проводят преобразование от основания 27 к основанию 2, используя уравнение таблицы I.3, соответствующее числу оставшихся значений знаков схемы кодирования по основанию 27.

Таблица I.3 — Уравнения кодирования знаков схемы кодирования по основанию 27

Число знаков данных	Уравнение кодирования	Длина, биты
1	C_1	5
2	$C_1 + C_2 \times 27$	10
3	$C_1 + C_2 \times 27 + C_3 \times 27^2$	15
4	$C_1 + C_2 \times 27 + C_3 \times 27^2 + C_4 \times 27^3$	20
5	$C_1 + C_2 \times 27 + C_3 \times 27^2 + C_4 \times 27^3 + C_5 \times 27^4$	24

I.2.3 Пример

Процесс кодирования по основанию 27 строки знаков DATA<ПРОБЕЛ>MATRIX приведен на рисунке I.2.

Данные	**D**	**A**	**T**	**A**	**ПРОБЕЛ**	**M**	**A**	**T**	**R**	**I**	**X**
Значение в схеме кодирования по основанию 27	4	1	20	1	0	13	1	20	18	9	24
Позиция знака	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5	C_1
Весовой коэффициент	1	27	729	19683	531441	1	27	729	19683	531441	1
Результат произведения	4	27	14580	19683	0	13	27	14580	354294	4782969	24
Десятичное значение	34294					5151883					24
Двоичная строка	000000001000010111110110					010011101001110010001011					11000

Рисунок I.2 — Пример кодирования по основанию 27

I.3 Схема кодирования по основанию 37

I.3.1 Процедура первого этапа

Исходные знаки данных должны быть преобразованы в соответствующие значения схемы кодирования по основанию 37, используя таблицу I.1 как переходную таблицу.

I.3.2 Процедура второго этапа

Уплотнение последовательности значений знаков схемы кодирования по основанию 37 в двоичную строку проводят следующим образом:

a) разбивают слева направо последовательность значений знаков схемы кодирования по основанию 37 на группы по четыре значения в каждой. Если в группе менее четырех значений, переходят к этапу, указанному в перечислении e)[1)];

b) присваивают обозначения четырем значениям знаков в группе как $C_1,...,C_4$, где C_1 — значение первого знака;

[1)] *В оригинале ИСО/МЭК 16022 ошибочно указано: «...переходят к этапу 5».*

c) проводят преобразование от основания 37 к основанию 2, получая последовательность, состоящую из 21 бита с помощью уравнения для числа знаков 4 по таблице I.4;

d) повторяют действия, начиная с этапа, указанного в перечислении a) (при необходимости);

e) если число значений знаков в группе менее четырех, проводят преобразование от основания 37 к основанию 2, используя уравнение (с 1 по 3) из таблицы I.4 соответствующее числу оставшихся значений знаков схемы кодирования по основанию 37.

Таблица I.4 — Уравнения кодирования схемы кодирования по основанию 37 (прописные латинские буквы и цифры)

Число знаков данных	Уравнение кодирования	Длина, биты
1	C_1	6
2	$C_1 + C_2 \times 37$	11
3	$C_1 + C_2 \times 37 + C_3 \times 37^2$	16
4	$C_1 + C_2 \times 37 + C_3 \times 37^2 + C_4 \times 37^3$	21

I.3.3 Пример

Процесс кодирования строки знаков 123ABCD89 по основанию 37 приведен на рисунке I.3.

Данные	**1**	**2**	**3**	**A**	**B**	**C**	**D**	**8**	**9**
Значение в схеме кодирования по основанию 37	28	29	30	1	2	3	4	35	36
Позиция знака	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1
Весовой коэффициент	1	37	1369	50653	1	37	1369	50653	1
Результат произведения	28	1073	41070	50653	2	111	5476	1772855	36
Десятичное значение	92824				1778444				36
Двоичная строка	00001011010101001100 0				11011001000110000110 0				100100

Рисунок I.3 — Пример кодирования по основанию 37

I.4 Схема кодирования по основанию 41

I.4.1 Процедура первого этапа

Исходные знаки данных должны быть преобразованы в соответствующие значения схемы кодирования по основанию 41, используя таблицу I.1 как переходную таблицу.

I.4.2 Процедура второго этапа

Уплотнение последовательности значений знаков схемы кодирования по основанию 41 в двоичную строку проводят следующим образом:

a) разбивают слева направо последовательность значений знаков схемы кодирования по основанию 37 на группы по четыре значения в каждой. Если в группе менее четырех значений, то переходят к этапу, указанному в перечислении e)[1];

b) присваивают обозначения четырем значениям в группе как $C_1,...,C_4$, где C_1 — значение первого знака;

c) проводят преобразование от основания 41 к основанию 2, получая последовательность, состоящую из 22 битов с помощью уравнения для числа знаков 4 в таблице I.5;

d) повторяют действия, начиная с этапа, указанного в перечислении a) (при необходимости);

e) если число значений знаков в группе менее четырех, проводят преобразование от основания 41 к основанию 2, используя уравнение таблицы I.5, соответствующее числу оставшихся значений знаков схемы кодирования по основанию 41.

[1] *В оригинале ИСО/МЭК 16022 ошибочно указано: «...переходят к этапу 5».*

Таблица I.5 — Уравнения кодирования схемы кодирования по основанию 41 (цифры, специальные графические знаки и прописные латинские буквы)

Число знаков данных	Уравнение кодирования	Длина, биты
1	C_1	6
2	$C_1 + C_2 \times 41$	11
3	$C_1 + C_2 \times 41 + C_3 \times 41^2$	17
4	$C_1 + C_2 \times 41 + C_3 \times 41^2 + C_4 \times 41^3$	22

I.4.3 Пример

Процесс кодирования по основанию 41 строки знаков AB/C123-X приведен на рисунке I.4.

Данные	**A**	**B**	**/**	**C**	**1**	**2**	**3**	**-**	**X**
Значение в схеме кодирования по основанию 41	1	2	40	3	28	29	30	39	24
Позиция знака	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1
Весовой коэффициент	1	41	1681	68921	1	41	1681	68921	1
Результат произведения	1	82	67240	206763	28	1189	50430	2687919	24
Десятичное значение	274086				2739566				24
Двоичная строка	000100001011101010100110				101001110011010110110				011000

Рисунок I.4 — Пример кодирования по основанию 41

Приложение J
(обязательное)

Алгоритм расчета CRC для ECC 000-140

В данном приложении приведены два способа представления значений контроля циклической избыточности (CRC).

J.1 Конечный автомат CRC

Контроль циклической избыточности (CRC) представлен в виде схемы на рисунке J.1. После того как исходный двоичный поток будет пропущен через конечный автомат, итоговое значение CRC (последовательность битов) считывается из 16 регистров памяти (m) в соответствии со схемой на рисунке J.1 (крайний левый регистр соответствует старшему значащему разряду битов).

J.2 Полином CRC

Алгоритмом расчета CRC должен быть типовой полиномом CCITT[1]:

$$X^{16} + X^{12} + X^5 + 1.$$

Если X равно 2, полином представляет собой 17-битовое значение $10001000000100001_{\text{основание 2}}$.

CRC представляет собой остаток от деления исходной строки данных на это значение.

J.3 Двухбайтовая головная метка CRC

Головные метки для расчета CRC, определенные в соответствии с таблицей J.1, используют в операциях CRC как префикс к значениям 8-битовых байтов знаков данных. Двухбайтовая головная метка CRC поступает на вход конечного автомата перед началом расчета CRC.

Таблица J.1 — Головная метка для расчета CRC

Идентификатор формата	Схема кодирования	Головная метка для расчета CRC		
		Байт старшего порядка	Байт младшего порядка	Шестнадцатиричное значение
1	По основанию 11	00000001	00000000	01 00
2	По основанию 27	00000010	00000000	02 00
3	По основанию 41	00000011	00000000	03 00
4	По основанию 37	00000100	00000000	04 00
5	КОИ-7 (ASCII)	00000101	00000000	05 00
6	Для 8-битовых байтов	00000110	00000000	06 00

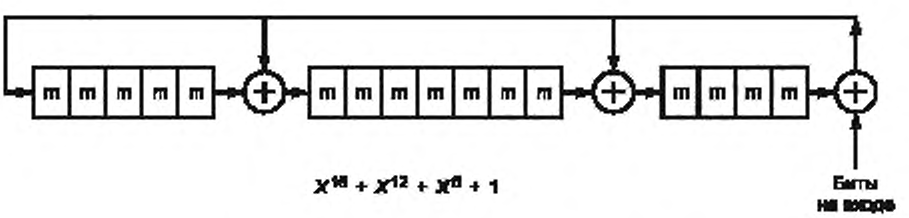


Рисунок J.1 — Схема алгоритма расчета CRC

[1] *Consultative Commitce for International Telegraph and Telephone (CCITT) - прежнее наименование Международного консультатиеного комитета по телеграфии и телефонии (МККТТ). С марта 1993 г. принято новое наименование — ITU-R).*

Приложение К
(обязательное)

Алгоритмы обнаружения и исправления ошибок для символов версии ECC 000-140

К.1 Символы уровня ECC 000

В указанных символах исправление ошибок не предусмотрено.

К.2 Символы уровня ECC 050

Двоичный поток с исправлением ошибок 'v' для символов уровня ECC 050 должен быть создан путем обработки незащищенного двоичного потока 'u', через конечный автомат, соответствующий сверточному коду структуры 4-3-3 (рисунок К.1).

К.3 Символы уровня ECC 080

Двоичный поток с исправлением ошибок 'v' для символов уровня ECC 080 должен быть создан путем обработки незащищенного двоичного потока 'u', через конечный автомат, соответствующий сверточному коду структуры 3-2-11 (рисунок К.2).

К.4 Символы уровня ECC 100

Двоичный поток с коррекцией ошибок 'v' для символов уровня ECC 100 должен быть создан путем обработки незащищенного двоичного потока 'u', через конечный автомат, соответствующий сверточному коду структуры 2-1-15 (рисунок К.3).

К.5 Символы уровня ECC 140

Двоичный поток с коррекцией ошибок 'v' для символов уровня ECC 140 должен быть создан путем обработки незащищенного двоичного потока 'u', через конечный автомат, соответствующий сверточному коду структуры 4-1-13 (рисунок К.4).

К.6 Обработка сверточных кодов

В схемах конечных автоматов применяют следующие обозначения:



— однобитовый регистр памяти;



— однобитовый сумматор с выводом на выходе младших битов, что эквивалентно генератору проверки четности;



— соединение линий;


— пересечение не соединяющихся линий.

Конечный автомат действует следующим образом:

a) регистры памяти (m) заполняют нулевыми значениями перед началом процесса;

b) выполняют входной цикл, состоящий в прохождении входного бита данного пользователя через входной переключатель в регистр памяти (m), для каждого возможного положения ключа-переключателя, т.е. для *k* битов;

c) после ввода всех *k* входных битов выполняют выходной цикл. Выходной цикл заключается в считывании выходных битов исправления ошибок при каждом возможном положении выходного ключа-переключателя, т.е. для *n* битов. При каждом положении ключа-переключателя выходной бит вычисляют путем выполнения операции «исключающего ИЛИ» (XOR) над подключенными к сумматору разрядами регистра сдвига;

d) после одного цикла ввода и вывода выполняют операцию сдвига, состоящую в переносе всех значений регистров сдвига вправо на одну позицию;

e) повторяют действия, указанные в перечислениях b) — d) — до тех пор, пока не будут введены все входные биты:

1) возможно потребуется добавить несколько нулевых битов к окончанию последнего сегмента входных битов, чтобы обеспечить ввод *k* битов;

2) добавляют на вход соответствующее количество дополнительных нулевых битов до тех пор, пока все m регистры сдвига не вернутся к нулевым значениям. Выходные данные этапов, указанные в перечислениях e) 1) и 2), представляют собой часть закодированных данных. Процесс завершают после того, как все истинные входные биты данных прошли через последний крайний правый регистр памяти.

К.7 Рекомендуемый алгоритм декодирования на основе сверточных кодов

Для декодирования данных, защищенных с помощью сверточных кодов, может использоваться алгоритм Фано, исправляющий ошибки. Описание действия алгоритма Фано приведено в [1].

Построение декодера сверточного кодирования рекомендуется проводить следующим образом.

Начальные значения переменных величин должны быть следующими:

- обратная метрика (движение назад) — максимальное отрицательное число;
- текущая метрика — 0;
- прямая метрика (движение вперед) — 0;
- порог — 0.

Метрику вычисляют путем определения числа отличающихся битов между обрабатываемым поврежденным блоком и подобранным блоком-кандидатом:

Metric = (1*correct bits) – (penalty*incorrect)

Метрика = (1 × число совпадающих битов) — (коэффициент потерь × число неверных (несовпадающих) битов).

Значения коэффициентов потерь на один бит и параметра Delta алгоритма Фано, которые должны использоваться при декодировании символа любой версии ECC, приведены в таблице К.1

Таблица К.1 — Коэффициенты алгоритма Фано

Уровень символа ECC	Коэффициент потерь на один бит	Параметр Delta
050	31	20
080	16	11
100	8	6
140	4	1

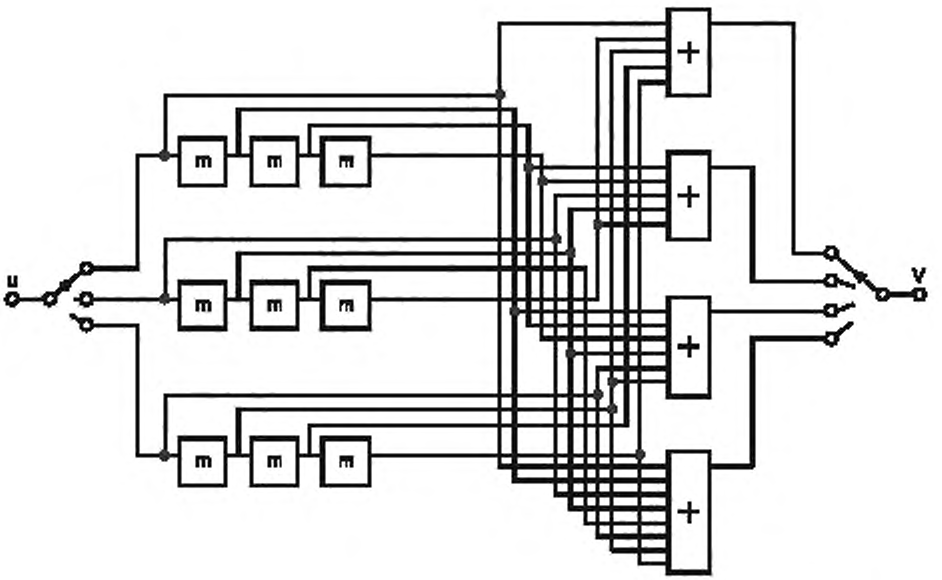


Рисунок К.1 — Схема кодирования со структурой 4-3-3 для символа уровня ECC 050

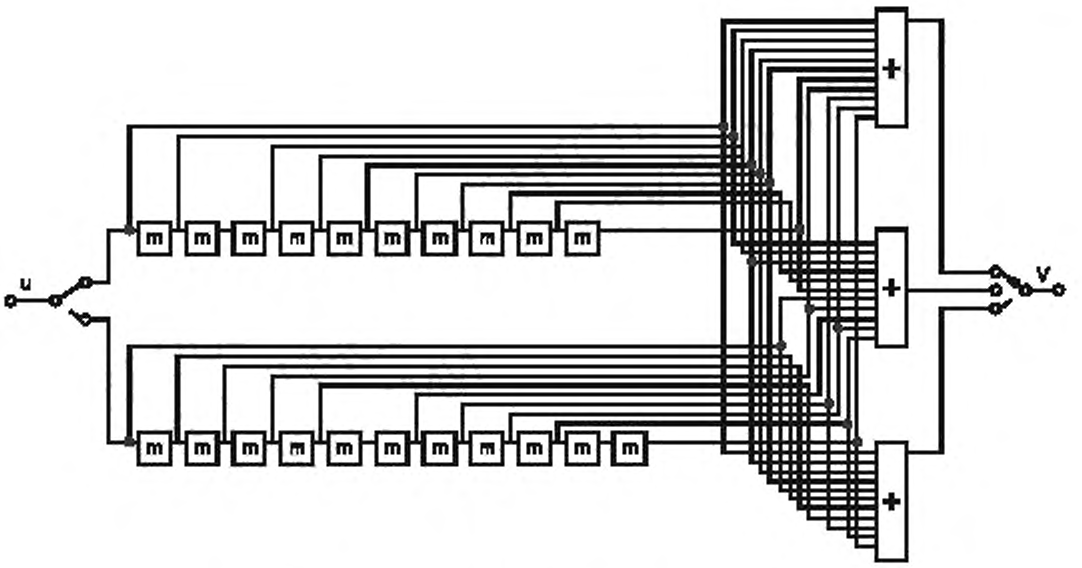


Рисунок К.2 — Схема кодирования со структурой 3-2-11 для символа уровня ECC 080



Рисунок К.3 — Схема кодирования со структурой 2-1-15 для символа уровня ECC 100

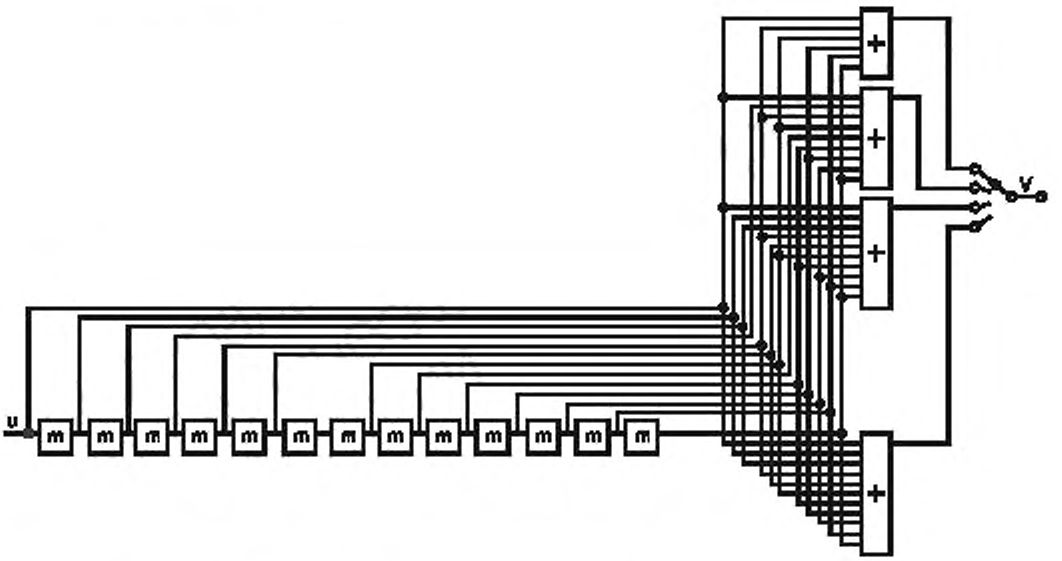


Рисунок К.4 — Схема кодирования со структурой 4-1-13 для символа уровня ECC 140

Приложение L
(обязательное)

Образец рандомизации двоичного потока в символах версии ECC 000-140 (шестнадцатеричные значения)

(Бит старшего порядка)

05	ff	c7	31	88	a8	83	9c	64	87	9f	64	b3	e0	4d	9c	80	29	3a	90
b3	8b	9e	90	45	bf	f5	68	4b	08	cf	44	b8	d4	4c	5b	a0	ab	72	52
1c	e4	d2	74	a4	da	8a	08	fa	a7	c7	dd	00	30	a9	e6	64	ab	d5	8b
ed	9c	79	f8	08	d1	8b	c6	22	64	0b	33	43	d0	80	d4	44	95	2e	6f
5e	13	8d	47	62	06	eb	80	82	c9	41	d5	73	8a	30	23	24	e3	7f	b2
a8	0b	ed	38	42	4c	d7	b0	ce	98	bd	e1	d5	e4	c3	1d	15	4a	cf	d1
1f	39	26	18	93	fc	19	b2	2d	ab	f2	6e	a1	9f	af	d0	8a	2b	a0	56
b0	41	6d	43	a4	63	f3	aa	7d	af	35	57	c2	94	4a	65	0b	41	de	b8
e2	30	12	27	9b	66	2b	34	5b	b8	99	e8	28	71	d0	95	6b	07	4d	3c
7a	b3	e5	29	b3	ba	8c	cc	2d	e0	c9	c0	22	ec	4c	de	f8	58	07	fc
19	f2	64	e2	c3	e2	d8	b9	fd	67	a0	bc	f5	2e	c9	49	75	62	82	27
10	f4	19	6f	49	f7	b3	84	14	ea	eb	e1	2a	31	ab	47	7d	08	29	ac
bb	72	fa	fa	62	b8	c8	d3	86	89	95	fd	df	cc	9c	ad	f1	d4	6c	64
23	24	2a	56	1f	36	eb	b7	d6	ff	da	57	f4	50	79	08	0	(Бит младшего порядка)		

Приложение М
(обязательное)

Качество печати символов Data Matrix. Аспекты, связанные с особенностями символики

Из-за различий в структурах, установленных символикой и рекомендуемых алгоритмах декодирования, влияние некоторых параметров на эффективность считывания символов может варьироваться для различных символик. ИСО/МЭК 15415 обеспечивает для спецификаций символик задание классификации признаков, связанных с особенностями символики. В настоящем приложении установлена методика классификации повреждения фиксированного шаблона (Fixed Pattern Damage), которую следует использовать при применении ИСО/МЭК 15415 к символике Data Matrix.

М.1 Повреждение фиксированного шаблона символики Data Matrix

М.1.1 Параметры, подлежащие оценке

Параметры фиксированного шаблона, подлежащего оценке, находятся внутри периметра символа шириной в один модуль и окружающей символ свободной зоны шириной не менее одного модуля (или более, в соответствии с установленным значением в стандарте по применению). В символах большой емкости, содержащих направляющий шаблон (символы квадратной формы размером в 32 × 32 модуля и более или символы прямоугольной формы размером 8 × 32, 12 × 36 и более), направляющий шаблон также является частью фиксированного шаблона. Левая и нижняя стороны символа должны образовывать однородную темную фигуру в виде буквы «L» шириной один модуль, а правая и верхняя стороны должны состоять из чередующихся одиночных темных и светлых модулей (называемых дорожкой синхронизации). Направляющие штрихи и внутренние дорожки синхронизации направляющего шаблона должны представлять собой по всему символу сплошные темные полосы шириной один модуль и последовательности из чередующихся одиночных темных и светлых модулей соответственно. При классификации по параметру повреждение фиксированного шаблона следует учитывать кроме общего числа поврежденных модулей, также концентрацию повреждений.

М.1.2 Классификация внешней L-образной структуры фиксированного шаблона

Повреждение каждой стороны L-образной структуры следует классифицировать на основании модуляции отдельных модулей, которые составляют ее. Соответствующую оценку проводят по всей длине каждой из сторон L-образной структуры и соседних свободных зон.

На рисунке М.1 приведены четыре сегмента L1, L2, QZL1 и QZL2. Сегмент L1 является вертикальной частью L-образной структуры, продленной на один модуль в свободную зону, смежную с углом L-образной структуры. Сегмент L2 является горизонтальной частью L-образной структуры, продленной на один модуль в свободную зону, смежную с углом L-образной структуры. Сегменты QZL1 и QZL2 являются частями свободной зоны, смежными с сегментами L1 и L2 соответственно и продленные на один модуль за внешние (удаленные от угла) концы L1 и L2 соответственно как показано на затемненных участках рисунка М.1. Модуль, находящийся на пересечении сторон угла структуры L, входит в оба сегмента L1 и в L2, так же как и модуль на пересечении сегментов QZL1 и в QZL2.

Все сегменты поочередно должны быть подвергнуты следующей процедуре:

a) определяют класс модуляции для каждого модуля с использованием методики по ИСО/МЭК 15415. Так как заранее известно, какой модуль следует считать темным или светлым, любой модуль, который должен быть темным, но коэффициент отражения которого выше глобального порога, а также любой модуль, который должен быть светлым, но коэффициент отражения которого ниже глобального порога, приводит к классу модуляции ноль;

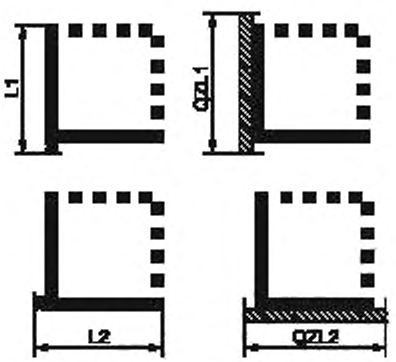


Рисунок М.1 — Сегменты внешней L-образной структуры и свободной зоны фиксированного шаблона

b) к каждому классу модуляции применяют метод классификации параметров, приведенный в ИСО/МЭК 15415:

1) для каждой стороны L-образной структуры (сегменты L1 и L2 на рисунке М.1) и каждой части свободной зоны (сегменты QZL1 и QZL2, прилегающие соответственно к сегментам L1 и L2 на рисунке М.1), считают, что все модули, класс модуляции которых меньше выбранного на этапе, указанном в перечислении b), являются поврежденными модулями и вычисляют условный класс повреждения на основе пороговых значений класса по таблице М.1. Сравнивают полученный условный класс модуляции со значением, полученным на этапе, указанном в перечислении b), и меньшее из этих значений считают классом модуляции;

2) классом любого сегмента следует считать наивысший полученный класс для всех уровней классов модуляции;

c) для символов и квадратной и прямоугольной форм, имеющий более одной области данных, повторяют этапы, указанные в перечислениях a) и b), за исключением того, что сегменты L1 и L2 начинают с модуля в свободной зоне и продолжают до модуля дорожки синхронизации того же поля данных включительно, а сегменты QZL1 и QZL2 включают в себя части свободной зоны, смежной с сегментами L1 и L2, то есть рассматривают левую нижнюю область данных так, будто это символ с одной областью данных. Если полученный на этом этапе класс ниже полученного для сегментов L1, L2, QZL1 и QZL2 на этапах, указанных в перечислениях a) и b), то заменяют классы на этапах, указанных в перечислениях a) и b), на указанный класс;

d) для сегментов L1 и L2 проводят проверку на наличие областей, включающих в себя более трех поврежденных модулей подряд, и поврежденных участков между блоками, состоящими более чем из четырех корректных модулей. Если во время испытаний обнаружены такие области, то класс модуляции, определенный на этапе, указанном в перечислении a), должен быть равен нулю;

Таблица М.1 — Пороговые значения классов для угловых повреждений

Повреждение модулей, %	Класс
0	4
≤ 9	3
≤ 13	2
≤ 17	1
> 17	0

e) класс параметра повреждения фиксированного шаблона для сегмента должен соответствовать наибольшему из полученных значений модуляции.

М.1.3 Оценка сегментов дорожки синхронизации и смежных однородных областей

В настоящем пункте установлена методика измерения повреждений внутренних направляющих шаблонов (при их наличии), а также внешних дорожек синхронизации на правой и верхней сторонах символа и примыкающих свободных зон. Данные измерения проводят отдельно для каждого сегмента внутренних направляющих шаблонов, дорожек синхронизации и соответствующих свободных зон, которые служат границами области данных символа, или отдельных областей данных для больших символов. Каждый сегмент состоит из частей дорожек синхронизации и однородных областей (которые являются либо частью свободной зоны, либо частью темной полосы внутреннего направляющего шаблона).

Часть дорожек синхронизации начинают с темного модуля на стороне L-образной структуры или на внутренней темной полосе направляющего шаблона, перпендикулярной к дорожке, и продолжают до светлого модуля, предшествующего свободной зоне или до темной полосы следующего направляющего шаблона.

Часть однородной области со штрихом направляющего шаблона, не примыкающую к свободной зоне, начинают с модуля, смежного с первым модулем связанной части дорожки синхронизации, и продолжают до модуля, следующего за последним модулем связанной части дорожки синхронизации. На рисунке М.4 (a) представлена структура этих сегментов. Однородные сегменты, которые соответствуют частям внешней свободной зоны, определяют в соответствии с рисунком М.2

Часть однородной области со штрихом направляющего шаблона, примыкающую к свободной зоне, начинают с модуля, смежного с первым модулем связанной части дорожки синхронизации, и продолжают до модуля, смежного с последним модулем связанной части дорожки синхронизации. На рисунке М.4 (b) изображена структура этих сегментов.

Методика измерений состоит из следующих этапов:

a) для каждого сегмента внешней дорожки синхронизации или сегмента внутреннего направляющего шаблона (для символов с несколькими сегментами) повреждения вычисляют с помощью следующей процедуры;

b) оценка по коэффициенту переходов.

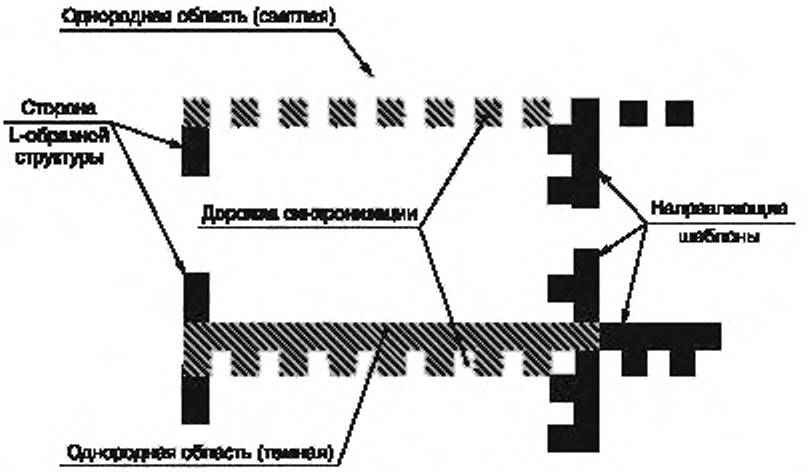


Рисунок М.2 — Структура сегмента внешней дорожки синхронизации и сегмента внутреннего направляющего шаблона

Примечание — В символах без внутренних направляющих шаблонов, сегменты внешних дорожек синхронизации продолжают на полную длину или высоту символа.

Примечание — Внутренний сегмент направляющего шаблона, завершающийся у другого внутреннего сегмента направляющего шаблона того же цвета, приведен на рисунке М.2

Для всех сегментов дорожки синхронизации в черно-белом (двоичном) изображении как внешних (смежных со свободной зоной), так и внутренних (смежных с темной полосой внутреннего направляющего шаблона) подсчитывают число переходов *Tc* на стороне дорожки синхронизации и число переходов *Ts* на стороне однородной области и вычисляют класс коэффициента переходов (transition ratio) *TR* по формулам:

$$Ts' = \text{Max}\,(0,\ Ts - 1);$$

$$TR = Ts'/Tc.$$

Таблица М.2 — Класс коэффициента переходов

Значение *TR*	Класс
$TR < 0{,}06$	4
$0{,}06 \le TR < 0{,}08$	3
$0{,}08 \le TR < 0{,}10$	2
$0{,}10 \le TR < 0{,}12$	1
$TR \ge 0{,}12$	0

Примечание — Конечными точками линий, по которым проводят подсчет числа переходов, являются пересечения линий сетки, нанесенные согласно рекомендуемому алгоритму декодирования, в первом и последнем модуле дорожки синхронизации (рисунок М.3);

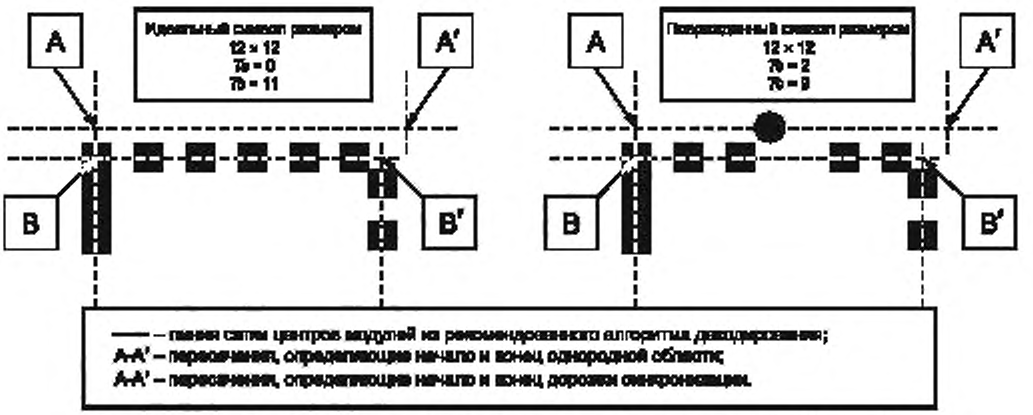


Рисунок М.3 — Переходы в идеальном символе (слева) и в поврежденном (справа)

c) условный класс повреждения.

Вычисляют класс модуляции для каждого модуля с помощью методики по ИСО/МЭК 15415. Так как заранее известно, является данный модуль темным или светлым, любой модуль, предполагаемый темным, коэффициент отражения которого выше глобального порога, а так же любой модуль, предполагаемый светлым, коэффициент отражения которого ниже глобального порога, получает класс модуляции, равный нулю;

d) для каждого уровня класса модуляции считают все модули:

имеющие класс модуляции меньше класса, выбранного выше, «поврежденными модулями», и вычисляют условный класс повреждения с помощью трех контрольных оценок, указанных в перечислениях e) - g);

e) оценка регулярности дорожки синхронизации.

Для каждого сегмента дорожки синхронизации берут группы, состоящие из пяти смежных модулей, и, двигаясь вдоль сегмента с шагом в один модуль, проверяют, чтобы в любой группе из пяти смежных модулей было не более двух поврежденных модулей. Если это условие выполняется, класс регулярности дорожки синхронизации должен быть равен 4, в противном случае он должен быть равен нулю.

f) оценка наличия повреждений дорожки синхронизации.

Для каждого сегмента подсчитывают число некорректных модулей дорожки синхронизации. Определение доли повреждений P — отношения числа некорректных модулей к общему числу модулей в сегменте (в процентах) в результате даст классы повреждения дорожки синхронизации (таблица М.3);

g) оценка фиксированного шаблона однородной области.

Для каждого сегмента подсчитывают число некорректных модулей в однородных областях (темные линии внутреннего направляющего шаблона или внешней свободной зоны), смежных с дорожкой синхронизации. Определение доли повреждений P — отношения числа некорректных модулей к общему числу модулей в сегменте (в процентах) в результате даст классы повреждения (таблица М.3);

Таблица М.3 — Оценка повреждений сегментов дорожек синхронизации и сегментов однородных областей

Доля повреждений P, %	Класс
$P < 10$	4
$10 \le R < 15$	3
$15 \le R < 20$	2
$20 \le R < 25$	1
$P \ge 25$	0

h) для каждого уровня класса выбирают наименьший уровень класса модуляции, класса регулярности дорожки синхронизации, класса доли повреждений дорожки синхронизации и класса доли повреждений фиксированного шаблона однородной области в процентах;

i) класс условного повреждения для сегмента должен соответствовать наибольшему значению класса модуляции, из всех полученных на этапе, указанном в перечислении h);

j) класс повреждения фиксированного шаблона для сегмента должен соответствовать наименьшему из значений класса оценки по коэффициенту переходов и класса условных повреждений;

k) полный класс повреждения фиксированного шаблона для сегментов дорожки синхронизации и смежной однородной области соответствует наименьшему значению из классов, полученных для каждого отдельного сегмента.

Сегмент внутреннего направляющего шаблона, включающий в себя часть дорожки синхронизации и часть однородной области, для которых проводят оценки коэффициента переходов на регулярность и наличие повреждений фиксированных шаблонов однородных областей, представлен в качестве примера на рисунке М.4 и выделен затенением.

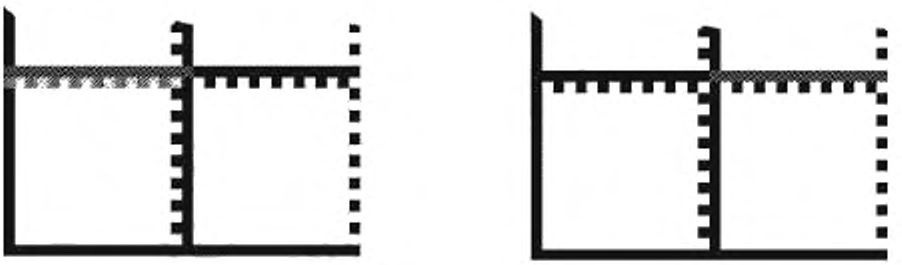


Рисунок М.4 — Сегмент внутреннего направляющего шаблона, завершающийся у внешней свободной зоны

Сегмент внешней дорожки синхронизации и смежной свободной зоны, для которых производят проверки коэффициента переходов на регулярность и наличие повреждений фиксированных шаблонов однородных областей, представлен в качестве примера на рисунке М.5 и выделен затенением.

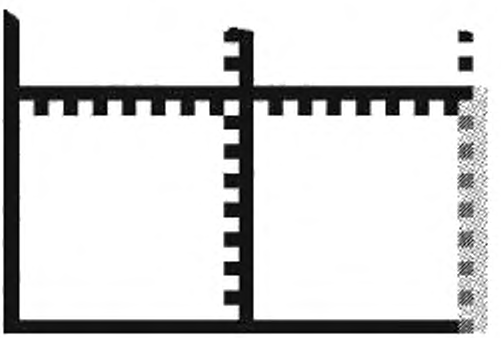


Рисунок М.5 — Сегмент внешней дорожки синхронизации



Рисунок М.6 — Пример, представляющий 37 модулей, подлежащих оценке, на L-образной стороне символа размером 36 × 36 модулей[1)]

Пример — Пример оценки класса сегмента L1 символа размером 36 × 36, в котором *SC* = 89 % и *GT* = 51 % приведен на рисунке М.6. Значения коэффициентов отражения и модуляции, а также классы модуляции приведены в таблице М.4 для всех 36 модулей этого сегмента.

Значения коэффициентов отражения и модуляции, а также классы модуляции приведены в таблице М.4 для модулей данного сегмента от 0 до 36. Дополнительный модуль свободной зоны, смежный со структурой L, обозначен как модуль 0.

[1)] *Пример приведен для демонстрации эффектов модуляции.*

Таблица М.4 — Пример оценки модуляции для сегмента из 36 модулей

Номер модуля	0	1	2	3	4	5	6	7	8	9
Коэффициент отражения, %	84	15	13	13	13	9	11	84	11	10
MOD (значение модуляции)	74	80	86	86	86	94	90	(74)	90	92
Класс модуляции	4	4	4	4	4	4	4	0	4	4
Номер модуля		10	11	12	13	14	15	16	17	18
Коэффициент отражения, %		9	11	70	13	12	15	11	11	11
MOD (значение модуляции)		94	90	(42)	86	88	80	90	90	90
Класс модуляции		4	4	0	4	4	4	4	4	4
Номер модуля		19	20	21	22	23	24	25	26	27
Коэффициент отражения, %		27	11	14	10	12	50	12	11	14
MOD (значение модуляции)		54	90	83	92	88	2	88	90	83
Класс модуляции		4	4	4	4	4	0	4	4	4
Номер модуля		28	29	30	31	32	33	34	35	36
Коэффициент отражения, %		13	12	37	13	12	13	11	13	12
MOD (значение модуляции)		86	88	31	86	88	86	90	86	88
Класс модуляции		4	4	2	4	4	4	4	4	4

Примечание — Следует обратить внимание на то, что модули 7 и 12 — светлые, а модуль 24 и, в меньшей степени, модуль 30 имеют низкую модуляцию.

На основе этих значений оценивают класс сегмента (таблица М.5).

Таблица М.5 — Пример оценки класса сегмента

Класс *MOD*	Число модулей	Общее число модулей	Оставшиеся поврежденные модули	Доля поврежденных модулей, %	Условный класс повреждений	Наименьшее значение класса
4	32	32	4	10,8	2	2
3	0	32	4	10,8	2	2
2	1	33	3	8,1	3	2
1	0	33	3	8,1	3	1
0	3	37	0	0	4	0
Окончательный класс сегмента соответствует наибольшему значению, указанному в графе «Наименьшее значение класса»						2

М.1.4 Вычисление и оценка усредненного класса

Дополнительно к оценке отдельных сегментов следует также вычислить средний класс (AG), чтобы учесть совокупный эффект повреждений, которые имеют относительно незначительное влияние в отдельных сегментах, но затрагивают несколько сегментов. В основу данного вычисления положено определение среднего значе-

ния классов сегментов L1, L2, QZL1, QZL2 и полного класса сегмента дорожки синхронизации и смежной однородной области.

Если определены классы всех сегментов, вычисляют средний класс (*AG*) по формуле

$$AG = (\text{Сумма классов сегментов}) / 5.$$

Назначают класс *AG* в соответствии с таблицей М.6.

Класс повреждений фиксированного шаблона для символа должен быть меньшим из пяти классов сегментов и класса AG.

Таблица М.6 — Оценка для класса *AG*

Среднее значение классов пяти сегментов	Класс *AG*
4	4
≥ 3,5	3
≥ 3,0	2
≥ 2,5	1
< 2,5	0

Пример 1 — Если четыре из пяти сегментов имеют класс 4 и один сегмент имеет класс 1, в этом случае

$$(4 \times 4) + (1 \times 1) = 17.$$

Следовательно

$$AG = 17 / 5 = 3{,}4.$$

По таблице М.6 среднему значению 3,4 соответствует класс 2. Наименьшим из этих шести классов[1)] является класс 1, и класс повреждений фиксированного шаблона соответственно принимает значение 1.

Пример 2 — Если три из пяти сегментов имеют класс 4, один сегмент - класс 3 и один сегмент имеет класс 1, в этом случае

$$(3 \times 4) + (1 \times 3) + (1 \times 1) = 16.$$

Следовательно

$$AG = 16 / 5 = 3{,}2.$$

По таблице М.6 среднему значению 3,2 соответствует класс 2. Наименьшим из этих шести классов[1)] является класс 1, и класс повреждений фиксированного шаблона соответственно принимает значение 1.

Пример 3 — Если все пять сегментов имеют класс 3, в этом случае

$$5 \times 3 = 15.$$

Следовательно

$$AG = 15 / 5 = 3{,}0.$$

По таблице М.6 среднему значению 3,0 соответствует класс 2. Наименьшим из этих шести классов[1)] является 2, и класс повреждений фиксированного шаблона соответственно принимает значение 2.

М.2 Класс сканирования

Класс сканирования — это наименьший из классов типовых параметров, оцениваемых по ИСО/МЭК 15415, вместе с классом повреждений фиксированного шаблона, определяемого в соответствии с требованиями настоящего приложения.

[1)] *В данном случае в расчет принимают пять классов рассматриваемых сегментов и один класс, вычисленный как усредненное значение 5 классов сегментов. Таким образом рассматривают шесть классов.*

Приложение N
(обязательное)

Идентификатор символики

ИСО/МЭК 15424 обеспечивает единую методологию для сообщения о типе считанной символики, наборе вариантов обработки устройством считывания и других особенностях, встречающихся в символике.

Идентификатором символики Data Matrix является:

]dm

где

] — знак флага идентификатора символики (знак ASCII (КОИ-7) с десятичным значением 93);

d — знак кода для символики Data Matrix (знак ASCII (КОИ-7) с десятичным значением 100);

m — модификатор с одним из значений, установленных в таблице N.1.

Таблица N.1 — Значения вариантов обработки идентификатора символики для символа Data Matrix

Значение m	Вариант обработки
0	Символ версии ECC 000-140
1	Символ версии ECC 200
2	Символ версии ECC 200 с знаком FNC1 в 1-й или 5-й позиции
3	Символ версии ECC 200 с знаком FNC1 во 2-й или 6-й позиции
4	Символ версии ECC 200 с поддержкой протокола ECI
5	Символ версии ECC 200 с знаком FNC1 в 1-й или 5-й позиции и поддержкой протокола ECI
6	Символ версии ECC 200 с знаком FNC1 во 2-й или 6-й позиции и поддержкой протокола ECI
Примечание — Допустимые значения m: 0, 1, 2, 3, 4, 5 и 6.	

Приложение О
(справочное)

Пример кодирования символа версии ECC 200

В этом примере данными пользователя, подлежащими кодированию, является строка «123456» длиной 6 знаков.

Этап 1. Кодирование данных

Представление в 7-битовых знаках ASCII (КОИ-7):

Знаки данных:	'1'	'2'	'3'	'4'	'5'	'6'
Десятичные значения знаков:	49	50	51	52	53	54

В схеме кодирования ASCII (КОИ-7) осуществляется преобразование шести вышеуказанных знаков в три байта с использованием следующей формулы для пар цифр:

Значение кодового слова = (численное значение пары цифр) + 130.

Для данного примера вычисляют:

"12" = 12 + 130 = 142;

"34" = 34 + 130 = 164;

"56" = 56 + 130 = 186.

Поток данных после кодирования:

Десятичные значения: 142 164 186.

В соответствии с таблицей 7 размещают три кодовых слова данных в символе размером 10 × 10, который необходимо дополнить пятью кодовыми словами исправления ошибок. Если кодируемые данные не полностью заполняют область данных, то должны быть закодированы дополнительные знаки-заполнители.

Этап 2. Проверка и исправление ошибок

Кодовые слова исправления ошибок генерируют с использованием алгоритма Рида-Соломона и добавляют к потоку кодированных данных:

Номер кодового слова:	1	2	3	4	5	6	7	8
Десятичное значение кодового слова:	142	164	186	114	25	5	88	102
Шестнадцатеричное значение кодового слова:	8E	A4	BA	72	19	05	58	66
	Данные			Проверка				

Процесс генерации кодовых слов исправления ошибок для символа версии ECC 200 описан в приложении Е. Пример процедуры расчета кодовых слов исправления ошибок приведен в разделе Е.3.

Этап 3. Размещение модулей в матрице

Сформированные кодовые слова после этапа 2 размещают в двоичной матрице как знаки символа согласно алгоритму, указанному в 5.8.1 (также см. рисунок F.1):

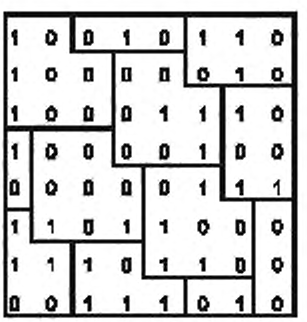


Рисунок О.1 — Позиционирование модулей в матрице

Этап 4. Действительный символ

Окончательный вид символа Data Matrix формируют путем добавления модулей шаблона поиска и преобразования двоичных единиц в темные модули и двоичных нулей — в светлые.

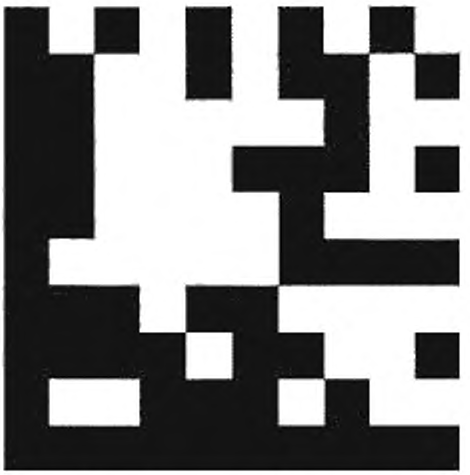


Рисунок О.2 — Окончательный вид символа Data Matrix, кодирующего строку «123456»

Приложение Р
(справочное)

Кодирование данных в символе ECC 200 с использованием минимального числа знаков символа

Одни и те же данные могут быть представлены различными символами Data Matrix путем использования различных схем кодирования.

Следующий алгоритм обычно порождает самый короткий поток кодовых слов:

a) кодирование начинают по схеме кодирования ASCII (КОИ-7/КОИ-8);

b) при кодировании по схеме кодирования ASCII (КОИ-7):

1) если следующая последовательность данных состоит, по крайней мере, из двух последовательных цифр, то кодируют следующие две цифры как сдвоенный разряд в схеме кодирования ASCII (КОИ-7/КОИ-8);

2) если проверка с предварительным просмотром (начинающимся на этапе, указанном в перечислении j), указывает иную схему кодирования, то следует перейти на эту схему;

3) если указана схема кодирования по основанию 256, кодируют знак фиксации этой схемы кодирования со следующим за ним байтом длины, значение которого пока неизвестно. На этапе, указанном в перечислении g или i, поле длины будет заполнено (может потребоваться второй дополнительный байт длины);

4) если следующий знак данных относится к расширенному набору ASCII (КОИ-8) и его десятичное значение превосходит 127, то его кодируют в схеме кодирования ASCII (КОИ-7/КОИ-8) с использованием в качестве первого знака управляющего знака верхнего регистра (Upper Shift) с десятичным значением 235;

5) в противном случае следующий знак данных кодируют по схеме кодирования ASCII (КОИ-7/КОИ-8);

c) при кодировании по схеме кодирования C40:

1) если схема кодирования C40 начинает обработку нового двойного знака символа, и если проверка с предварительным просмотром (начинающаяся на этапе, приведенном в перечислении j), указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак данных обрабатывают по схеме кодирования C40;

d) При кодировании по схеме кодирования Text:

1) если схема кодирования Text начинает обработку нового двойного знака символа, и если проверка с предварительным просмотром (начинающаяся на этапе, приведенном в перечислении j), указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак данных обрабатывают по схеме кодирования Text;

e) при кодировании по схеме кодирования ANSI X12:

1) если схема кодирования ANSI X12 начинает обработку нового двойного знака символа, и если проверка с предварительным просмотром (начинающаяся на этапе, приведенном в перечислении j) указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак данных обрабатывают по схеме кодирования ANSI X12;

f) при кодировании по схеме кодирования EDIFACT (EDF):

1) если схема кодирования EDIFACT начинает обработку нового тройного знака символа, и если проверка с предварительным просмотром (начинающаяся на этапе, приведенном в перечислении j) указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак данных обрабатывают по схеме кодирования EDIFACT;

g) при кодировании по схеме кодирования по основанию 256 (B256):

1) если проверка с предварительным просмотром (начинающаяся на этапе, приведенном в перечислении j) указывает иную схему кодирования, следует перейти на использование этой схемы;

2) в противном случае следующий знак данных обрабатывают по схеме кодирования по основанию 256;

h) повторяют этап, указанный в перечислении b), до конца данных;

i) после просмотра данных, если действует схема кодирования по основанию 256, устанавливают поле длины в ноль (ноль указывает, что символ завершает схему кодирования по основанию 256).

Проверка с предварительным просмотром (этапы, указанные в перечислениях с j по s):

При проверке с предварительным просмотром изучают кодируемые данные для определения наилучшей схемы кодирования;

j) задают начальное значение счетчика знаков символа для каждой схемы:

1) если текущей является схема кодирования ASCII (КОИ-7), устанавливают следующее:

счетчик схемы кодирования ASCII (КОИ-7/КОИ-8) равен 0;

счетчик схемы кодирования C40 равен 1;

счетчик схемы кодирования Text равен 1;

счетчик схемы кодирования ANSI X12 равен 1;

счетчик схемы кодирования EDIFACT равен 1;
счетчик схемы кодирования по основанию 256 равен 1,25.

В противном случае устанавливают:

счетчик схемы кодирования ASCII (КОИ-7/КОИ-8) равен 1;
счетчик схемы кодирования C40 равен 2;
счетчик схемы кодирования Text равен 2;
счетчик схемы кодирования ANSI X12 равен 2;
счетчик схемы кодирования EDIFACT равен 2;
счетчик схемы кодирования по основанию 256 равен 2,25.

2) если текущей является схема кодирования C40, то счетчик этой схемы равен 0;

3) если текущей является схема кодирования Text, то счетчик этой схемы равен 0;

4) если текущей является схема кодирования ANSI X12, то счетчик этой схемы равен 0;

5) если текущей является схема кодирования EDIFACT, то счетчик этой схемы равен 0;

6) если текущей является схема кодирования по основанию 256, то счетчик этой схемы равен 0;

k) после просмотра данных:

1) округляют в большую сторону все счетчики до целых чисел;

2) если значение счетчика схемы кодирования ASCII (КОИ-7/КОИ-8) не больше значений остальных счетчиков, завершают проверку с указанием схемы кодирования ASCII (КОИ-7/КОИ-8);

3) если значение счетчика схемы кодирования по основанию 256 меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования по основанию 256;

4) если значение счетчика схемы кодирования EDIFACT меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования EDIFACT;

5) если значение счетчика схемы кодирования Text меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования Text;

6) если значение счетчика схемы кодирования ANSI X12 меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования ANSI X12;

7) заканчивают проверку с указанием схемы кодирования C40;

l) обработка счетчика схемы кодирования ASCII (КОИ-7/КОИ-8):

1) если знак данных — цифра, добавляют 1/2 к значению счетчика схемы кодирования ASCII (КОИ-7/КОИ-8);

2) если знак данных — знак расширенного набора ASCII (КОИ-8) (знак с десятичным значением более 127), округляют значение счетчика схемы кодирования ASCII (КОИ-7/КОИ-8) в большую сторону до целого значения и добавляют к этому значению 2;

3) в противном случае округляют значение счетчика схемы кодирования ASCII (КОИ-7/КОИ-8) в большую сторону и добавляют к этому значению 1;

m) обработка счетчика схемы кодирования C40:

1) если знак данных входит в основной набор знаков C40, добавляют 2/3 к значению счетчика схемы кодирования C40;

2) если знак данных — знак расширенного набора ASCII (КОИ-8) (знак с десятичным значением более 127), добавляют 8/3 к значению счетчика схемы кодирования C40;

3) в противном случае добавляют 4/3 к значению счетчика схемы кодирования C40;

n) обработка счетчика схемы кодирования Text:

1) если знак данных входит в основной набор знаков схемы кодирования Text, добавляют 2/3 к значению счетчика схемы кодирования Text;

2) если знак данных — знак расширенного набора ASCII (КОИ-8) (знак с десятичным значением более 127), добавляют 8/3 к значению счетчика схемы кодирования Text;

3) в противном случае добавляют 4/3 к значению счетчика схемы кодирования Text;

o) обработка счетчика схемы кодирования ANSI X12:

1) если знак данных входит в число знаков по ANSI X12, добавляют 2/3 к значению счетчика схемы кодирования ANSI X12;

2) если знак данных — знак расширенного набора ASCII (КОИ-8) (знак с десятичным значением более 127), добавляют 13/3 к значению счетчика схемы кодирования ANSI X12;

3) в противном случае добавляют 10/3 к значению счетчика схемы кодирования ANSI X12;

p) обработка счетчика схемы кодирования EDIFACT:

1) если знак данных входит в число знаков по EDIFACT, добавляют 3/4 к значению счетчика схемы кодирования EDIFACT[1)];

2) если знак данных — знак расширенного набора ASCII (КОИ-8) (знак с десятичным значением более 127), добавляют 17/4 к значению счетчика схемы кодирования EDIFACT[1)];

3) в противном случае добавляют 13/4 к значению счетчика схемы кодирования EDIFACT[1)];

[1)] *В ИСО/МЭК 16022 указано значение счетчика схемы кодирования по ANSI X12.*

q) обработка счетчика схемы кодирования по основанию 256:

1) если знак является управляющим знаком (FNC 1, структурированного соединения (Structured Append), программирования устройства считывания (Reader Program) или кодовой страницы (Code Page)), добавляют 4 к значению счетчика схемы кодирования по основанию 256;

2) в противном случае добавляют 1 к значению счетчика схемы кодирования по основанию 256;

r) если, по крайней мере, четыре знака данных были обработаны в цикле проверки:

1) если значение счетчика схемы кодирования ASCII (КОИ-7/КОИ-8), к которому прибавлена 1, не больше значений остальных счетчиков, завершают проверку с указанием схемы кодирования ASCII (КОИ-7/КОИ-8);

2) если значение счетчика схемы кодирования по основанию 256, к которому прибавлена 1, не больше значения счетчика схемы кодирования ASCII (КОИ-7/КОИ-8) или меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования по основанию 256;

3) если значение счетчика схемы кодирования EDIFACT, к которому прибавлена 1, меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования EDIFACT;

4) если значение счетчика схемы кодирования Text, к которому прибавлена 1, меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования Text;

5) если значение счетчика схемы кодирования ANSI X12, к которому прибавлена 1, меньше значений остальных счетчиков, завершают проверку с указанием схемы кодирования ANSI X12;

6) если значение счетчика схемы кодирования C40, к которому прибавлена 1, меньше значений счетчиков схем кодирования ASCII (КОИ-7/КОИ-8), по основанию 256, EDIFACT, а также Text, то :

i) если значение счетчика схемы кодирования C40 меньше значения счетчика схемы кодирования ANSI X12, завершают проверку с указанием схемы кодирования C40;

ii) если значение счетчика схемы кодирования C40 равно значению счетчика схемы кодирования ANSI X12, то:

I) если один из трех знаков: знак-ограничитель и знаки-разделители по ANSI X12 впервые встречается в еще не обработанных данных перед знаком, не представленным в ANSI X12, завершают проверку с указанием схемы кодирования ANSI X12;

II) в противном случае проводят возврат к схеме кодирования C40;

s) повторяют этап, указанный в перечислении k), до встречи с условием возврата.

Приложение Q
(справочное)

Пример кодирования данных в символах версии ECC 000-140 с использованием кода исправления ошибок уровня ECC 050

Q.1 Пример кодирования

Данными пользователя, подлежащими кодированию, является последовательность "AB12-X". Эти данные кодируют по основанию 41 (идентификатор формата равен 3)

Этап 1. Кодирование данных:

	Первая последовательность знаков	Вторая последовательность знаков
a) подразделение данных на четырехзнаковые последовательности:	A B 1 2	-X
b) преобразование к кодовым значениям по основанию 41:	1 2 28 29	39 24
c) использование уравнений преобразования:	2045860	1023
d) преобразование в двоичный поток битов:	01111100111011110100100	01111111111
e) преобразование каждой последовательности для создания окончательного потока кодированных битов:	00100101111011001111110	11111111110

Этап 2. Построение префикса данных:

a) поле идентификатора формата для основания 41 выбирают по таблице 11 (раздел 6.5.1[1]):

00010;

b) поле контроля циклической избыточности CRC вычисляют как указано в Q.2[2] , а затем меняют местами старшие и младшие биты:

1001 1010 1010 1110;

c) поле длины должно быть равно шести в двоичном виде с перестановкой старших и младших разрядов (MSB/LSB):

011000000;

d) окончательный незащищенный двоичный поток приведен на рисунке Q.1.

Этап 3. Обнаружение и исправление ошибок:

Незащищенный двоичный поток подразделяют на блоки по три бита при подготовке к вводу в конечный автомат уровня ECC 050. К перечню входных блоков добавляют три дополнительных заполненных нулями блока, что дает в целом 24 входных блока (рисунок Q.1). Число дополнительных блоков (с нулями) равно наибольшей длине регистра сдвига для используемого конечного автомата ECC; для уровня ECC 050 добавляют три блока. Основными операциями всех конечных автоматов уровней ECC 050-140 являются следующие:

a) обнуляют регистры конечного автомата;

b) вводят очередной входной блок (MSB находится в позиции 1);

c) проводят вычисления на выходах всех вентилей «исключающее ИЛИ»;

d) производят запись в выходной поток (MSB находится в позиции 1).

---

[1] *В ИСО/МЭК 16022 ошибочно указана ссылка на 5.4.1.*

[2] *Результат вычислений - 0111 0101 0101 1001.*

В таблице Q.1 указаны значения всех регистров конечного автомата в ходе сверточного кодирования 24 входных блоков.

Окончательный защищенный двоичный поток длиной 96 битов представляет собой:

0000 1010 1011 1111 1010 1010 1010 0000 0100 0011 0110 1000 0101 0001 1000 0000 1110 1010 1001 1010 1001 1000 0100 1010.

Незащищенный двоичный поток (этап 2):

00010 1001101010101110 011000000 0010010111101100111110 11111111110

Формат 3 CRC-16 Длина Закодированные данные

Незащищенный двоичный поток, подразделенный на блоки по три блока с дополнительными входными блоками (этап 3):

000 101 001 101 010 101 110 011 000 000 001 001 011 110 110 011 111 011 111 111 110 000 000 000

Рисунок Q.1 — Незащищенный двоичный поток при передаче от этапа 2 к этапу 3

Таблица Q.1 — Значения всех регистров конечного автомата в ходе сверточного кодирования

Цикл конечного автомата	Ввод 1 2 3	Регистр 1A 1B 1C 2A 2B 2C 3A 3B 3C	Вывод 1 2 3 4	Цикл конечного автомата	Ввод 1 2 3	Регистр 1A 1B 1C 2A 2B 2C 3A 3B 3C	Вывод 1 2 3 4
1	0	000	0		0	101	1
	0	000	0				0
	0	000	0	8	0	110	0
			0		1	101	0
2	1	000	1		1	010	0
	0	000	0				0
	1	000	1	9	0	011	0
			0		0	110	1
3	0	100	1		0	101	0
	0	000	0				0
	1	100	1	10	0	001	0
			1		0	011	0
4	1	010	1		0	010	1
	0	000	1				1
	1	110	1	11	0	000	0
			1		0	001	1
5	0	101	1		1	001	1
	1	000	0				0
	0	111	1	12	0	000	1
			0		0	000	0
6	1	010	1		1	100	0
	0	100	0				0
	1	011	1	13	0	000	0
			0		1	000	1
7	1	101	1		1	110	0
	1	010	0				1

Окончание таблицы Q.1

Цикл конечного автомата	Ввод 1 2 3	Регистр 1A 1B 1C 2A 2B 2C 3A 3B 3C	Вывод 1 2 3 4	Цикл конечного автомата	Ввод 1 2 3	Регистр 1A 1B 1C 2A 2B 2C 3A 3B 3C	Вывод 1 2 3 4
14	1	000	0		1	111	0
	1	100	0				1
	0	111	0	20	1	101	1
			1		1	111	0
15	1	100	1		1	111	1
	1	110	0				0
	0	011	0	21	1	110	1
			0		1	111	0
16	0	110	0		0	111	0
	1	111	0				1
	1	001	0	22	0	111	1
			0		0	111	0
17	1	011	1		0	011	0
	1	111	1				0
	1	100	1	23	0	011	0
			0		0	011	1
18	0	101	1		0	001	0
	1	111	0				0
	1	110	1	24	0	001	1
			0		0	001	0
19	1	010	1		0	000	1
	1	111	0				0

Этап 4. Построение головной и конечной меток:

Головная метка содержит двоичное поле кода исправления ошибок (ECC) для уровня 050 по таблице 12 (раздел 6.6.1) с перестановкой старших и младших битов (MSB/LSB):

0111000000000111000 (длиной 19 битов).

Конечная метка содержит достаточное число битов-заполнителей для того, чтобы нерандомизированный двоичный поток точно соответствовал квадратной матрице наименьших размеров. 96 битов в защищенном двоичном потоке и 19 битов в головной метке вместе составляют 115 битов.

Матрица данных размером 13 × 13 включает в себя 11 × 11 информационных битов (121 бит); это матрица наименьшего размера, способная включать в себя 115 битов. Добавляют шесть нулевых битов (121 — 115), поэтому конечная метка имеет вид:

000000.

Окончательный нерандомизированный двоичный поток приведен на рисунке Q.2.

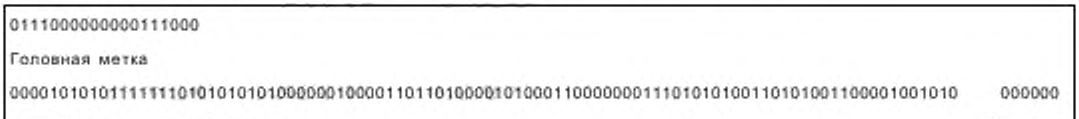


Рисунок Q.2 — Нерандомизированный двоичный поток

Этап 5. Шаблонная рандомизация:

Нерандомизированный двоичный поток разбивают на 4-битовые полубайты для облегчения выполнения операции исключающего ИЛИ (XOR):

0111 0000 0000 0111 0000 0001 0101 0111 1111 0101 0101 0100 0000 1000 0110 1101 0000 1010 0011 0000 0001 1101 0101 0011 0101 0011 0000 1001 0100 0000 0.

Получение требуемого числа (121) рандомизированных битов с помощью образца рандомизированного двоичного потока (приложение L):

(05, FF, C7, 31, 88, A8, 83, 9C, 64, 87, 9F, 64, B3, E0, 4D первый бит из 9C) =

=0000 0101 1111 1111 1100 0111 0011 0001 1000 1000 1010 1000 1000 0011 1001 1100 0110 0100 1000 0111 1001 1111 0110 0100 1011 0011 1110 0000 0100 1101 1.

Построение рандомизированного двоичного потока с применением операции исключающего ИЛИ (XOR) ко входному потоку со случайными битами:

0111 0101 1111 1000 1100 0110 0110 0110 0111 1101 1111 1100 1000 1011 1111 0001 0110 1110 1011 0111 1000 0010 0011 0111 1110 0000 1110 1001 0000 1101 1.

Этап 6. Размещение модулей в матрице:

Используя сетку размещения модулей данных для матрицы указанного размера, помещают модули данных в области данных двоичной матрицы:

11010011001
10010101101
10111001010
11011101010
01100001100
11101001101
00100111110
10101111001
01111101010
10010011110
00110110111.

После дополнения модулями шаблона поиска формируют окончательную двоичную матрицу:

1010101010101
1110100110010
1100101011011
1101110010100
1110111010101
1011000011000
1111010011011
1001001111100
1101011110011
1011111010100
1100100111101
1001101101110
1111111111111.

Q.2 Пример вычисления алгоритма CRC

Формируют поток битов для ввода в CRC алгоритм, состоящий из 2-байтовой головной метки CRC и следующими за ней исходными данными пользователя. 2-байтовая головная метка CRC согласно приложению J (таблица J.1) для формата 3 имеет вид:

0000 0011 0000 0000.

Первоначальные данные пользователя:

A B 1 2 — X;

0100 0001, 0100 0010, 0011 0001, 0011 0010, 0010 1101, 0101 1000.

Полный поток битов для ввода в алгоритм CRC перед изменением порядка байтов на обратный:

0000 0011, 0000 0000, 0100 0001, 0100 0010, 0011 0001, 0011 0010, 0010 1101, 0101 1000.

Полный поток битов для ввода в алгоритм CRC после изменением порядка байтов на обратный (64 бита):

1100 0000, 0000 0000, 1000 0010, 0100 0010, 1000 1100, 0100 1100, 1011 0100, 0001 1010.

Указанный поток битов вводят в конечный автомат CRC в соответствии с таблицей Q.2. Старший разряд CRC находится в крайнем левом регистре сдвига так, что окончательным вычисленным значением CRC является 0111010101011001 при чтении непосредственно из конечного автомата. Подразделение на 4-битовые полубайты дает 0111, 0101, 0101, 1001, что является значением поля CRC, используемого в приложении Q на этапе 2b.

Таблица Q.2 — Значения регистров при вычислении CRC

Цикл автомата	1 — 5 биты регистра XOR	Выход вентиля 3	6 — 12 биты регистра XOR	Выход вентиля 2	13 — 16 биты регистра XOR	Входной бит	Выход вентиля 1
Начало	00000	1	0000000	1	0000	1	1
1	10000	1	1000000	1	1000	1	1
2	11000	0	1100000	0	1100	0	0
3	01100	0	0110000	0	0110	0	0
4	00110	1	0011000	1	0011	0	1
5	10011	0	1001100	1	1001	0	1
6	11001	1	0100110	0	1100	0	0
7	01100	0	1010011	1	0110	0	0
8	00110	1	0101001	0	1011	0	1
9	10011	0	1010100	1	0101	0	1
10	11001	1	0101010	0	1010	0	0
11	01100	1	1010101	0	0101	0	1
12	10110	0	1101010	0	0010	0	0
13	01011	0	0110101	0	0001	0	1
14	10101	1	0011010	0	0000	0	0
15	01010	0	1001101	1	0000	0	0
16	00101	0	0100110	1	1000	1	1
17	10010	0	0010011	1	1100	0	0
18	01001	1	0001001	1	1110	0	0
19	00100	1	1000100	1	1111	0	1
20	10010	1	1100010	1	1111	0	1
21	11001	0	1110001	0	1111	0	1
22	11100	0	0111000	0	0111	1	0
23	01110	1	0011100	1	0011	0	1
24	10111	0	1001110	1	1001	0	1
25	11011	0	0100111	0	1100	1	1
26	11101	1	0010011	1	0110	0	0
27	01110	1	1001001	0	1011	0	1
28	10111	0	1100100	1	0101	0	1
29	11011	1	0110010	0	1010	0	0
30	01101	1	1011001	1	0101	1	0
31	00110	0	1101100	0	1010	0	0
32	00011	1	0110110	0	0101	1	0
33	00001	1	1011011	1	0010	0	0
34	00000	1	1101101	0	1001	0	1
35	10000	0	1110110	0	0100	0	0
36	01000	1	0111011	0	0010	1	1
37	10100	0	1011101	1	0001	1	0

Продолжение таблицы Q.2

Цикл автомата	1 — 5 биты регистра XOR	Выход вентиля 3	6 — 12 биты регистра XOR	Выход вентиля 2	13 — 16 биты регистра XOR	Входной бит	Выход вентиля 1
38	01010	0	0101110	0	1000	0	0
39	00101	1	0010111	1	0100	0	0
40	00010	0	1001011	1	1010	0	0
41	00001	1	0100101	1	1101	1	0
42	00000	0	1010010	0	1110	0	0
43	00000	1	0101001	0	0111	0	1
44	10000	0	1010100	0	0011	1	0
45	01000	0	0101010	0	0001	1	0
46	00100	0	0010101	1	0000	0	0
47	00010	0	0001010	0	1000	0	0
48	00001	0	0000101	0	0100	1	1
49	10000	0	0000010	0	0010	0	0
50	01000	0	0000001	1	0001	1	0
51	00100	1	0000000	1	1000	1	1
52	10010	0	1000000	0	1100	0	0
53	01001	0	0100000	1	0110	1	1
54	10100	1	0010000	1	1011	0	1
55	11010	1	1001000	1	1101	0	1
56	11101	1	1100100	0	1110	0	0
57	01110	1	1110010	1	0111	0	1
58	10111	0	1111001	0	1011	0	1
59	11011	1	0111100	0	0101	1	0
60	01101	0	1011110	1	0010	1	1
61	10110	1	0101111	0	1001	0	1
62	11011	0	1010111	0	0100	1	1
63	11101	1	0101011	1	0010	0	0
64	01110		1010101		1001		

Приложение R
(справочное)

Рекомендации по методам контроля процесса формирования символов

В настоящем приложении описаны средства и процедуры, рекомендуемые для мониторинга и контроля процесса формирования пригодных для сканирования символов Data Matrix. Эти методы не включают в себя проверку качества печати производимых символов (метод, описанный в разделе 8 и приложении М, необходим для оценки качества печати символа), но они по отдельности и все вместе предоставляют полезные рекомендации относительно того, создает ли данная технология печати символа пригодные для работы символы.

R.1 Контраст символа

Большинство верификаторов линейного штрихового кода используют или режим рефлектометра, или режим графического построения профилей отображения при сканировании, и/или формируют отчеты о контрасте символа (по ИСО/МЭК 15415 и ИСО/МЭК 19762) для недекодируемых сканирований. За исключением символов, требующих специальной конфигурации освещения, результаты считывания контраста символа с использованием апертуры размером 6 или 10 мил[1)] при оптическом излучении с длиной волны 660 нм (а также значение контраста символа и диапазон крайних значений в профиле отражения при сканировании) хорошо коррелируются со значениями контраста символа, полученными при обработке изображения. В частности, результаты считывания могут быть использованы для проверки того, что контраст символа остается выше минимального допустимого значения, определяемого классом символа.

R.2 Специальный рекомендуемый символ

Для целей контроля процесса формирования символа может быть проведена печать рекомендуемого символа версии ECC 200 размером 16 × 16 модулей, кодирующего данные "30Q324343430794<OQQ" (рисунок R.1). Как показано на этом рисунке, указанный рекомендуемый символ имеет область параллельных штрихов и пробелов, которые могут быть отсканированы как будто они представлены в линейном символе и затем оценены на приращение ширины штриха при печати, используя методики измерения края по ИСО/МЭК 15416.



Рисунок R.1 — Рекомендуемый символ, кодирующий "30Q324343430794<OQQ"

Многие верификаторы линейного штрихового кода могут быть запрограммированы с целью выдачи перечня значений ширины элементов, полученных по методике ИСО/МЭК 15416, даже при сканировании без декодирования. Левая часть любого линейного контрольного сканирования вдоль верхней половины рекомендуемого символа версии ECC 200 будет содержать четыре пары штрихов и пробелов, размеры которых по ширине обозначают от b_1 до b_4 и от s_1 до s_4.

Нормализованное показание горизонтального приращения при печати может быть рассчитано по формуле

$$(b_1 + b_2 + b_3 + b_4) / (b_1 + s_1 + b_2 + s_2 + b_3 + s_3 + b_4 + s_4).$$

Номинальная величина этого значения в символах Data Matrix должна составлять 50 % и оставаться в пределах от 35 % до 65 %.

Данное измерение не будет чувствительным к вариации печати, параллельной длинному размеру элементов в рекомендуемом символе. Если предпочтительна более полная оценка процесса печати, рекомендуемый символ Data Matrix должен быть напечатан и протестирован в двух ориентациях.

[1)] *1 мил = 0,0254 мм.*

R.3 Оценка осевой неоднородности

Для любого символа измеряют длину обеих сторон L-образного шаблона поиска. Длину каждой стороны делят на число модулей в пределах этого размера, например символ размером 12 × 36 должен иметь делители 12 и 36. Эти два нормализованных размера X_{AVG} и Y_{AVG} используют в формуле для оценки осевой неоднородности

$$AN = \text{abs}(X_{AVG} - Y_{AVG}) / ((X_{AVG} + Y_{AVG}) / 2).$$

Если значение *AN* больше 0,12, символ считают дефектным по методике ИСО/МЭК 15415. Значение вплоть до 0,06 соответствует классу 4 этого параметра.

R.4 Визуальная проверка искажений и дефектов символа

С помощью визуального осмотра шаблонов по периметру в образцах символов можно контролировать два важных аспекта процесса печати.

Во-первых, двумерные (2D) матричные символы восприимчивы к ошибкам, вызванным локальными искажениями матричной сетки. Любые такие искажения обнаруживают визуально в символе Data Matrix или как исправленные края L-образного шаблона поиска, или как неравные расстояния в пределах чередующихся шаблонов, обнаруженных вдоль двух других границ символа. Символы версии ECC 200 больших размеров также включают в себя чередующиеся шаблоны, прямолинейность и равномерность которых могут быть визуально проверены. Соответствующие символы с подобными дефектами могут быть быстро идентифицированы указанным способом.

Во-вторых, две части шаблона поиска и смежные свободные зоны должны всегда иметь коэффициенты отражения с противоположными значениями. Отказы в устройстве печати, которые могут приводить к дефектам в виде светлых или темных полос, проходящих через символ, явно заметны там, где они нарушают шаблон поиска или свободную зону. Такие систематические дефекты в процессе печати должны быть исправлены.

Приложение S
(справочное)

Возможность автоматического распознавания

Символы Data Matrix могут быть считаны с помощью надлежащим образом запрограммированных декодеров, которые предназначены для их автоматического распознавания среди символов иных символик. Набор символик, на распознавание которого запрограммирован декодер, должен быть ограничен той потребностью, которая определяется данным прикладным применением, что позволяет повысить надежность считывания до самого высокого уровня.

Приложение Т
(справочное)

Системный подход

Любое практическое применение Data Matrix следует рассматривать как законченное системное решение. Всеми компонентами, связанными с кодированием и декодированием символики (устройством нанесения символа на подложку или принтером, устройствами считывания, этикетками), образующими конкретную прикладную систему применения, необходимо управлять как целостной системой. Отказ в любом звене цепочки, или несогласованность между ними могут поставить под угрозу эффективность функционирования всей системы:

- хотя соответствие спецификациям является единственным ключом к построению и нормальной работе системы, рекомендуется принимать во внимание иные соображения, оказывающие влияние на эффективность функционирования системы. Следующие рекомендации учитывают ряд факторов, которые необходимо иметь в виду при разработке или внедрении систем штрихового кодирования;

- следует выбирать плотность печати с допусками, которые могут быть обеспечены используемой технологией маркировки или печати;

- следует подбирать устройство считывания с разрешающей способностью, наиболее подходящей для плотности и качества печати символа, которая обеспечивает данная технология печати;

- должна быть обеспечена совместимость оптических свойств напечатанного символа с длиной волны источника излучения и измерительным элементом сканера;

- следует проверять соответствие символа на законченной этикетке или окончательной конфигурации упаковки. Покрытия, прозрачная упаковка, а также изогнутые и неровные поверхности - все это может оказывать влияние на считывание символа.

Технологии маркировки, которые не способны последовательно формировать сплошные линии непрерывных модулей, например матричные или струйные принтеры, требуют проведения специальных мер, гарантирующих, что промежутки между номинально различимыми модулями не препятствуют декодированию символа при использовании размера апертуры, установленного в прикладном документе. Кроме того, относительное позиционирование модулей, горизонтальных и вертикальных осей должно соответствовать требованиям к осевой неоднородности, приведенным в ИСО/МЭК 15415. Спецификации по применению должны также учитывать требования ИСО/МЭК 15415 относительно определения размера апертуры, освещения и прочих параметров.

Сканирующие системы должны учитывать колебания в диффузном отражении между темными и светлыми областями. При использовании ряда углов сканирования, зеркальные составляющие отраженного излучения могут превышать требуемые составляющие диффузного компонента отражения, затрудняя эффективное считывание. В случаях, когда поверхности детали или материала могут быть изменены путем получения матовой поверхности или окончательной обработки, устраняющей влияние глянца, это может помочь минимизировать эффекты зеркального отражения. Если подобные мероприятия осуществить невозможно, необходимо принять меры по обеспечению специального освещения для считывания маркировки, чтобы оптимизировать контраст символа до приемлемого уровня.

Приложение U
(справочное)

Соответствие терминов на русском и английском языках

Таблица U.1 — Соответствие терминов на русском языке, использованных в настоящем стандарте, международным по ИСО/МЭК 16022.

Английский термин (словосочетание)	Русский термин-эквивалент
Структура символа	
Alinement Pattern	Направляющий шаблон
data region	Область данных
Finder Pattern	Шаблон поиска
Least significant bit (LSB)	Младший значащий разряд
Most significant bit (MSB)	Старший значащий разряд
Управляющие знаки символики	
Pad character	Знак-заполнитель
Shift character	Знак регистра
Shift 1 character	Знак регистра 1
Shift 2 character	Знак регистра 2
Shift 3 character	Знак регистра 3
Structured Append character	Знак структурированного соединения
Unlatch character	Знак отказа от фиксации
Upper Shift character	Знак верхнего регистра
Latch to C40 encodation	Знак фиксации схемы кодирования С40
Latch to Base 256 encodation	Знак фиксации схемы кодирования по основанию 256
Latch to ANSI X12 encodation	Знак фиксации схемы кодирования ANSI X12
Latch to Text encodation	Знак фиксации схемы кодирования TEXT
Latch to EDIFACT encodation	Знак фиксации схемы кодирования EDIFACT
Reader Programming	Знак программирования устройства считывания
05 Macro	Знак Макро 05
06 Macro	Знак Макро 06
Extended Channel Interpretation character (ECI)	Знак интерпретации в расширенном канале (ECI)
FNC1 character	Функциональный знак 1 (FNC1)
Кодирование данных	
ECI Assignment member	Номер назначения ECI
exclusive-or (XOR)	Операция «исключающего ИЛИ»
Extended Channel Interpretation (ECI)	Интерпретация в расширенном канале
Protected Bit Stream	Защищенный двоичный поток
Randomised Bit Stream	Рандомизированный двоичный поток
Text	Схема кодирования TEXT
Unprotected Data Bit Stream	Незащищенный двоичный поток
Unrandomised Bit Stream	Нерандомизированный двоичный поток
Encoded Data Bit Stream	Двоичный поток закодированных данных
Error Checking and Correction (ECC)	Алгоритм контроля и исправления ошибок

Приложение V
(справочное)

Набор знаков ASCII (версия КОИ-7) по ИСО/МЭК 646, графические знаки расширенного набора знаков ASCII (версии КОИ-8) по ИСО/МЭК 8859-1 и набор 8-битовых графических знаков (версия КОИ-8) по ИСО/МЭК 8859-5

V.1 Набор 7-битовых знаков ASCII (версия КОИ-7) по ИСО 646

В таблице V.1 приведен набор 7-битовых знаков ASCII (версия КОИ-7) по ИСО 646 и соответствие международных и русских наименований и обозначений знаков.

Таблица V.1 — Набор 7-битовых знаков ASCII (версия КОИ-7) по ИСО 646

Десятичное значение	Обозначения знака		Наименование знака	
	международное	русское	международное	русское
00	NUL	ПУС	NULL	ПУСТО
01	SOH	НЗ	START OF HEADING	НАЧАЛО ЗАГОЛОВКА
02	STX	НТ	START OF TEXT	НАЧАЛО ТЕКСТА
03	ETX	КТ	END OF TEXT	КОНЕЦ ТЕКСТА
04	EOT	КП	END OF TRANSMISSION	КОНЕЦ ПЕРЕДАЧИ
05	ENQ	КТМ	ENQUIRY	КТО ТАМ?
06	ACK	ДА	ACKNOWLEDGE	ПОДТВЕРЖДЕНИЕ
07	BEL	ЗВ	BELL	ЗВОНОК
08	BS	ВШ	BACKSPACE	ВОЗВРАТ НА ШАГ
09	HT	ГТ	HORIZONTAL TABULATION	ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ
10	LF	ПС	LINE FEED	ПЕРЕВОД СТРОКИ
11	VT	ВТ	VERTICAL TABULATION	ВЕРТИКАЛЬНАЯ ТАБУЛЯЦИЯ
12	FF	ПФ	FORM FEED	ПЕРЕВОД ФОРМАТА
13	CR	ВК	CARRIAGE RETURN	ВОЗВРАТ КАРЕТКИ
14	SO	ВЫХ	SHIFT-OUT	ВЫХОД
15	SI	ВХ	SHIFT-IN	ВХОД
16	DLE	АР1	DATA LINK ESCAPE	АВТОРЕГИСТР ОДИН
17	DC1	СУ1	DEVICE CONTROL ONE	СИМВОЛ УСТРОЙСТВА ОДИН
18	DC2	СУ2	DEVICE CONTROL TWO	СИМВОЛ УСТРОЙСТВА ДВА
19	DC3	СУ3	DEVICE CONTROL THREE	СИМВОЛ УСТРОЙСТВА ТРИ
20	DC4	СУ4	DEVICE CONTROL FOUR	СИМВОЛ УСТРОЙСТВА ЧЕТЫРЕ
21	NAK	НЕТ	NEGATIVE ACKNOWLEDGE	ОТРИЦАНИЕ
22	SYN	СИН	SYNCHRONOUS IDLE	СИНХРОНИЗАЦИЯ
23	ETB	КБ	END OF TRANSMISSION BLOCK	КОНЕЦ БЛОКА
24	CAN	АН	CANCEL	АННУЛИРОВАНИЕ
25	EM	КН	END OF MEDIUM	КОНЕЦ НОСИТЕЛЯ
26	SUB	ЗМ	SUBSTITUTE CHARACTER	ЗАМЕНА СИМВОЛА
27	ESC	АР2	ESCAPE	АВТОРЕГИСТР ДВА

Продолжение таблицы V.1

Десятичное значение	Обозначения знака		Наименование знака	
	международное	русское	международное	русское
28	FS	РФ	FILE SEPARATOR	РАЗДЕЛИТЕЛЬ ФАЙЛОВ
29	GS	РГ	GROUP SEPARATOR	РАЗДЕЛИТЕЛЬ ГРУПП
30	RS	РЗ	RECORD SEPARATOR	РАЗДЕЛИТЕЛЬ ЗАПИСЕЙ
31	US	РЭ	UNIT SEPARATOR	РАЗДЕЛИТЕЛЬ ЭЛЕМЕНТОВ
32	SP		SPACE	ПРОБЕЛ
33	!	!	EXCLAMATION MARK	ВОСКЛИЦАТЕЛЬНЫЙ ЗНАК
34	"	"	QUOTATION MARK	КАВЫЧКИ
35	#	#	NUMBER SIGN	НОМЕР
36[1)]	¤	¤	CURRENCY SIGH	ЗНАК ДЕНЕЖНОЙ ЕДИНИЦЫ
37	%	%	PERCENT SIGH	ПРОЦЕНТЫ
38	&	&	AMPERSAND	КОММЕРЧЕСКОЕ И (АМПЕРСАНД)
39	'	'	APOSTROPHE	АПОСТРОФ
40	(	(	LEFT PARENTHESIS	КРУГЛАЯ СКОБКА ЛЕВАЯ
41	)	)	RIGHT PARENTHESIS	КРУГЛАЯ СКОБКА ПРАВАЯ
42	*	*	ASTERISK	ЗВЕЗДОЧКА
43	+	+	PLUS SIGH	ПЛЮС
44	,	,	COMMA	ЗАПЯТАЯ
45	-	-	HYPHEN-MINUS	ДЕФИС, МИНУС
46	.	.	FULL STOP	ТОЧКА
47	/	/	SOLIDUS	ДРОБНАЯ ЧЕРТА
48	0	0	DIGIT ZERO	ЦИФРА НОЛЬ
49	1	1	DIGIT ONE	ЦИФРА ОДИН
50	2	2	DIGIT TWO	ЦИФРА ДВА
51	3	3	DIGIT THREE	ЦИФРА ТРИ
52	4	4	DIGIT FOUR	ЦИФРА ЧЕТЫРЕ
53	5	5	DIGIT FIVE	ЦИФРА ПЯТЬ
54	6	6	DIGIT SIX	ЦИФРА ШЕСТЬ
55	7	7	DIGIT SEVEN	ЦИФРА СЕМЬ
56	8	8	DIGIT EIGHT	ЦИФРА ВОСЕМЬ
57	9	9	DIGIT NINE	ЦИФРА ДЕВЯТЬ
58	:	:	COLON	ДВОЕТОЧИЕ
59	;	;	SEMICOLON	ТОЧКА С ЗАПЯТОЙ
60	<	<	LESS THAN SIGN	МЕНЬШЕ
61	=	=	EQUALS SIGN	РАВНО

[1)] *В ИСО/МЭК 16022 знаку с десятичным значением 36 соответствует знак $ — DOLLAR SIGH (ДЕНЕЖНЫЙ ЗНАК ДОЛЛАРА).*

Продолжение таблицы V.1

Десятичное значение	Обозначения знака		Наименование знака	
	международное	русское	международное	русское
62	>	>	GREATER THAN SIGN	БОЛЬШЕ
63	?	?	QUESTION MARK	ВОПРОСИТЕЛЬНЫЙ ЗНАК
64	@	@	COMMERCIAL AT	КОММЕРЧЕСКОЕ ЭТ
65	A	A	LATIN CAPITAL LETTER **A**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **A**
66	B	B	LATIN CAPITAL LETTER **B**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **B**
67	C	C	LATIN CAPITAL LETTER **C**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **C**
68	D	D	LATIN CAPITAL LETTER **D**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **D**
69	E	E	LATIN CAPITAL LETTER **E**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **E**
70	F	F	LATIN CAPITAL LETTER **F**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **F**
71	G	G	LATIN CAPITAL LETTER **G**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **G**
72	H	H	LATIN CAPITAL LETTER **H**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **H**
73	I	I	LATIN CAPITAL LETTER **I**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **I**
74	J	J	LATIN CAPITAL LETTER **J**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **J**
75	K	K	LATIN CAPITAL LETTER **K**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **K**
76	L	L	LATIN CAPITAL LETTER **L**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **L**
77	M	M	LATIN CAPITAL LETTER **M**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **M**
78	N	N	LATIN CAPITAL LETTER **N**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **N**
79	O	O	LATIN CAPITAL LETTER **O**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O**
80	P	P	LATIN CAPITAL LETTER **P**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **P**
81	Q	Q	LATIN CAPITAL LETTER **Q**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **Q**
82	R	R	LATIN CAPITAL LETTER **R**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **R**
83	S	S	LATIN CAPITAL LETTER **S**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **S**
84	T	T	LATIN CAPITAL LETTER **T**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **T**

Продолжение таблицы V.1

Десятичное значение	Обозначения знака		Наименование знака	
	международное	русское	международное	русское
85	U	U	LATIN CAPITAL LETTER **U**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **U**
86	V	V	LATIN CAPITAL LETTER **V**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **V**
87	W	W	LATIN CAPITAL LETTER **W**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **W**
88	X	X	LATIN CAPITAL LETTER **X**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **X**
89	Y	Y	LATIN CAPITAL LETTER **Y**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **Y**
90	Z	Z	LATIN CAPITAL LETTER **Z**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **Z**
91	[	[	LEFT SQUARE BRACKET	КВАДРАТНАЯ СКОБКА ЛЕВАЯ
92	\	\	REVERSE SOLIDUS	ОБРАТНАЯ ДРОБНАЯ ЧЕРТА
93	]	]	RIGHT SQUARE BRACKET	КВАДРАТНАЯ СКОБКА ПРАВАЯ
94	^	^	CIRCUMFLEX ACCENT	ЦИРКЮМФЛЕКС УДАРЕНИЕ
95	_	_	LOW LINE	ПОДЧЕРКИВАНИЕ
96	'	'	GRAVE ACCENT	СЛАБОЕ УДАРЕНИЕ
97	a	a	LATIN SMALL LETTER **A**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a**
98	b	b	LATIN SMALL LETTER **B**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **b**
99	c	c	LATIN SMALL LETTER **C**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **c**
100	d	d	LATIN SMALL LETTER **D**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **d**
101	e	e	LATIN SMALL LETTER **E**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **e**
102	f	f	LATIN SMALL LETTER **F**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **f**
103	g	g	LATIN SMALL LETTER **G**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **g**
104	h	h	LATIN SMALL LETTER **H**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **h**
105	i	i	LATIN SMALL LETTER **I**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **i**
106	j	j	LATIN SMALL LETTER **J**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **j**
107	k	k	LATIN SMALL LETTER **K**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **k**
108	l	l	LATIN SMALL LETTER **L**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **l**
109	m	m	LATIN SMALL LETTER **M**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **m**
110	n	n	LATIN SMALL LETTER **N**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **n**
111	o	o	LATIN SMALL LETTER **O**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o**
112	p	p	LATIN SMALL LETTER **P**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **p**
113	q	q	LATIN SMALL LETTER **Q**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **q**
114	r	r	LATIN SMALL LETTER **R**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **r**
115	s	s	LATIN SMALL LETTER **S**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **s**
116	t	t	LATIN SMALL LETTER **T**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **t**
117	u	u	LATIN SMALL LETTER **U**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **u**

Окончание таблицы V.1

Десятичное значение	Обозначения знака		Наименование знака	
	международное	русское	международное	русское
118	v	v	LATIN SMALL LETTER **V**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **v**
119	w	w	LATIN SMALL LETTER **W**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **w**
120	x	x	LATIN SMALL LETTER **X**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **x**
121	y	y	LATIN SMALL LETTER **Y**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **y**
122	z	z	LATIN SMALL LETTER **Z**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **z**
123	{	{	LEFT CURLY BRACKET	ФИГУРНАЯ СКОБКА ЛЕВАЯ
124	|	|	VERTICAL LINE	ВЕРТИКАЛЬНАЯ ЧЕРТА
125	}	}	RIGHT CURLY BRACKET	ФИГУРНАЯ СКОБКА ПРАВАЯ
126	~	~	TILDE	ТИЛЬДА
127	DEL	DEL	DELETE	ЗАБОЙ

Примечание — Набор 7-битовых знаков ASCII (версия КОИ-7) состоит из набора знаков G0 по ИСО/МЭК 646 и C0 по ИСО/МЭК 6429, в котором знаки с десятичными значениями от 28 до 31 соответствуют знакам FS, GS, RS и US национальной версии США[1)] соответственно.

V.2 Графические знаки расширенного набора знаков ASCII (версия КОИ-8) по ИСО/МЭК 8859-1

В таблице V.2 приведены графические знаки расширенного набора 8-битовых знаков ASCII (версия КОИ-8) по ИСО/МЭК 8859-1:1998 (латинский алфавит № 1) и соответствие международных и русских наименований и обозначений знаков. В связи с тем, что знаки указанного набора с десятичными значениями с 0 по 127 полностью совпадают с набором знаков ASCII (КОИ-7), в таблице V.2 приводятся 8-битовые графические знаки с десятичными значениями от 160 по 255.

Таблица V.2 — Графические знаки расширенного набора 8-битовых знаков ASCII (версия КОИ-8) по ИСО/МЭК 8859-1

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
160	NBSP (НПР)	NO-BREAK SPACE	НЕПРЕРЫВАЮЩИЙ ПРОБЕЛ
161	¡	INVERTED EXCLAMATION MARK	ПЕРЕВЕРНУТЫЙ ВОСКЛИЦАТЕЛЬНЫЙ ЗНАК
162	¢	CENT SIGN	ДЕНЕЖНЫЙ ЗНАК ЦЕНТА
163	£	POUND SIGN	ДЕНЕЖНЫЙ ЗНАК ФУНТА
164	¤	CURRENCY SIGN	ЗНАК ДЕНЕЖНОЙ ЕДИНИЦЫ
165	¥	YEN SIGN	ДЕНЕЖНЫЙ ЗНАК ЙЕНЫ
166	¦	BROKEN BAR	ВЕРТИКАЛЬНАЯ ЧЕРТА С РАЗРЫВОМ
167	§	SECTION SIGN	ПАРАГРАФ
168	¨	DIAERESIS	ДИЕРЕЗ
169	©	COPYRIGHT SIGN	ЗНАК АВТОРСКОГО ПРАВА

[1)] *Набор знаков по ANSI INCITS 4-1986 (R2007) Information Systems - Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII) (Информационные системы — Кодированные наборы знаков — 7-битовый американский национальный стандартный код для обмена информацией (7- битовый ASCII).*

Продолжение таблицы V.2

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
170	ª	FEMININE ORDINAL INDICATOR	ОКОНЧАНИЕ ЧИСЛИТЕЛЬНОГО ЖЕНСКОГО РОДА
171	«	LEFT POINTING DOUBLE ANGLE QUOTATION MARK	ЗНАК ЛЕВОНАПРАВЛЕННОЙ ДВОЙНОЙ УГЛОВОЙ КАВЫЧКИ
172	¬	NOT SIGN	ЗНАК НЕТ
173		SOFT HYPHEN	ГИБКИЙ ДЕФИС
174	®	REGISTERED SIGN	ЗНАК РЕГИСТРАЦИИ
175	¯	MACRON	ЧЕРТА СВЕРХУ
176	°	DEGREE SIGN	ЗНАК ГРАДУСА
177	±	PLUS-MINUS SIGN	ЗНАК ПЛЮС-МИНУС
178	²	SUPERSCRIPT TWO	ВЕРХНИЙ ИНДЕКС ДВА
179	³	SUPERSCRIPT THREE	ВЕРХНИЙ ИНДЕКС ТРИ
180	´	ACUTE ACCENT	СИЛЬНОЕ УДАРЕНИЕ
181	µ	MICRO SIGN	ЗНАК МИКРО
182	¶	PILCROW SIGN	ЗНАК ПИ
183	·	MIDDLE DOT	СРЕДНЯЯ ТОЧКА
184	¸	CEDILLA	СЕДИЛЬ
185	¹	SUPERSCRIPT ONE	ВЕРХНИЙ ИНДЕКС ОДИН
186	º	MASCULINE INDICATOR ORDINAL	ОКОНЧАНИЕ ЧИСЛИТЕЛЬНОГО МУЖСКОГО РОДА
187	»	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK	ЗНАК ПРАВОНАПРАВЛЕННОЙ ДВОЙНОЙ УГЛОВОЙ КАВЫЧКИ
188	¼	VULGAR FRACTION ONE QUARTER	ПРОСТАЯ ДРОБЬ ОДНА ЧЕТВЕРТАЯ
189	½	VULGAR FRACTION ONE HALF	ПРОСТАЯ ДРОБЬ ОДНА ВТОРАЯ
190	¾	VULGAR FRACTION THREE QUARTERS	ПРОСТАЯ ДРОБЬ ТРИ ЧЕТВЕРТЫХ
191	¿	INVERTED QUESTION MARK	ПЕРЕВЕРНУТЫЙ ВОПРОСИТЕЛЬНЫЙ ЗНАК
192	À	LATIN CAPITAL LETTER **A** WITH GRAVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **А** СО СЛАБЫМ УДАРЕНИЕМ
193	Á	LATIN CAPITAL LETTER **A** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **А** С СИЛЬНЫМ УДАРЕНИЕМ
194	Â	LATIN CAPITAL LETTER **A** WITH CIRCUMFLEX	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **А** С ЦИРКУМФЛЕКСОМ
194	Ã	LATIN CAPITAL LETTER **A** WITH TILDE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **А** С ТИЛЬДОЙ
196	Ä	LATIN CAPITAL LETTER **A** WITH DIAERESIS	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **А** С ДИЕРЕЗОМ

Продолжение таблицы V.2

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
197	Å	LATIN CAPITAL LETTER **A** WITH RING ABOVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **A** С КРУЖКОМ СВЕРХУ
198	Æ	LATIN CAPITAL LETTER **AE**	ПРОПИСНАЯ ЛАТИНСКАЯ ЛИГАТУРА **Æ**
199	Ç	LATIN CAPITAL LETTER **C** WITH CEDILLA	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **C** С СЕДИЛЕМ
200	È	LATIN CAPITAL LETTER **E** WITH GRAVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **E** СО СЛАБЫМ УДАРЕНИЕМ
201	É	LATIN CAPITAL LETTER **E** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **E** С СИЛЬНЫМ УДАРЕНИЕМ
202	Ê	LATIN CAPITAL LETTER **E** WITH CIRCUMFLEX	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **E** С ЦИРКУМФЛЕКСОМ
203	Ë	LATIN CAPITAL LETTER **E** WITH DIAERESIS	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **E** С ДИЕРЕЗОМ
204	Ì	LATIN CAPITAL LETTER **I** WITH GRAVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **I** СО СЛАБЫМ УДАРЕНИЕМ
205	Í	LATIN CAPITAL LETTER **I** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **I** С СИЛЬНЫМ УДАРЕНИЕМ
206	Î	LATIN CAPITAL LETTER **I** WITH CIRCUMFLEX	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **I** С ЦИРКУМФЛЕКСОМ
207	Ï	LATIN CAPITAL LETTER **I** WITH DIAERESIS	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **I** С ДИЕРЕЗОМ
208	Ð	LATIN CAPITAL LETTER **ETH**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **D** СО ШТРИХОМ
209	Ñ	LATIN CAPITAL LETTER **N** WITH TILDE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **N** С ТИЛЬДОЙ
210	Ò	LATIN CAPITAL LETTER **O** WITH GRAVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** СО СЛАБЫМ УДАРЕНИЕМ
211	Ó	LATIN CAPITAL LETTER **O** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** С СИЛЬНЫМ УДАРЕНИЕМ
212	Ô	LATIN CAPITAL LETTER **O** WITH CIRCUMFLEX	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** С ЦИРКУМФЛЕКСОМ
213	Õ	LATIN CAPITAL LETTER **O** WITH TILDE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** С ТИЛЬДОЙ
214	Ö	LATIN CAPITAL LETTER **O** WITH DIAERESIS	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** С ДИЕРЕЗОМ
215	×	MULTIPLICATION SING	ЗНАК УМНОЖЕНИЯ
216	Ø	LATIN CAPITAL LETTER **O** WITH STROKE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **O** СО ШТРИХОМ
217	Ù	LATIN CAPITAL LETTER **U** WITH GRAVE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **U** СО СЛАБЫМ УДАРЕНИЕМ
218	Ú	LATIN CAPITAL LETTER **U** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **U** С СИЛЬНЫМ УДАРЕНИЕМ
219	Û	LATIN CAPITAL LETTER **U** WITH CIRCUMFLEX	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **U** С ЦИРКУМФЛЕКСОМ

Продолжение таблицы V.2

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
220	Ü	LATIN CAPITAL LETTER **U** WITH DIAERESIS	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **U** С ДИЕРЕЗОМ
221	Ý	LATIN CAPITAL LETTER **Y** WITH ACUTE	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **Y** С СИЛЬНЫМ УДАРЕНИЕМ
222	Þ	LATIN CAPITAL LETTER **THORN**	ПРОПИСНАЯ ЛАТИНСКАЯ БУКВА **þ** (ФОРН)
223	ß	LATIN SMALL LETTER SHARP **S**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **ß** (СДВОЕННАЯ s)
224	à	LATIN SMALL LETTER **A** WITH GRAVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** СО СЛАБЫМ УДАРЕНИЕМ
225	á	LATIN SMALL LETTER **A** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** С СИЛЬНЫМ УДАРЕНИЕМ
226	â	LATIN SMALL LETTER **A** WITH CIRCUMFLEX	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** С ЦИРКУМФЛЕКСОМ
227	ã	LATIN SMALL LETTER **A** WITH TILDE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** С ТИЛЬДОЙ
228	ä	LATIN SMALL LETTER **A** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** С ДИЕРЕЗОМ
229	å	LATIN SMALL LETTER **A** WITH RING ABOVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **a** С КРУЖКОМ СВЕРХУ
230	æ	LATIN SMALL LETTER **AE**	СТРОЧНАЯ ЛАТИНСКАЯ ЛИГАТУРА **æ**
231	ç	LATIN SMALL LETTER **C** WITH CEDILLA	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **c** С СЕДИЛЕМ
232	è	LATIN SMALL LETTER **E** WITH GRAVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **e** СО СЛАБЫМ УДАРЕНИЕМ
233	é	LATIN SMALL LETTER **E** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **e** С СИЛЬНЫМ УДАРЕНИЕМ
234	ê	LATIN SMALL LETTER **E** WITH CIRCUMFLEX	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **e** С ЦИРКУМФЛЕКСОМ
235	ë	LATIN SMALL LETTER **E** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **e** С ДИЕРЕЗОМ
236	ì	LATIN SMALL LETTER **I** WITH GRAVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **i** СО СЛАБЫМ УДАРЕНИЕМ
237	í	LATIN SMALL LETTER **I** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **i** С СИЛЬНЫМ УДАРЕНИЕМ
238	î	LATIN SMALL LETTER **I** WITH CIRCUMFLEX	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **i** С ЦИРКУМФЛЕКСОМ
239	ï	LATIN SMALL LETTER **I** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **i** С ДИЕРЕЗОМ
240	ð	LATIN SMALL LETTER **ETH**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **ð**
241	ñ	LATIN SMALL LETTER **N** WITH TILDE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **n** С ТИЛЬДОЙ
242	ò	LATIN SMALL LETTER **O** WITH GRAVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** СО СЛАБЫМ УДАРЕНИЕМ

Окончание таблицы V.2

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
243	ó	LATIN SMALL LETTER **O** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** С СИЛЬНЫМ УДАРЕНИЕМ
244	ô	LATIN SMALL LETTER **O** WITH CIRCUMFLEX	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** С ЦИРКУМФЛЕКСОМ
245	õ	LATIN SMALL LETTER **O** WITH TILDE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** С ТИЛЬДОЙ
246	ö	LATIN SMALL LETTER **O** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** С ДИЕРЕЗОМ
247	÷	DIVISION SING	ЗНАК ДЕЛЕНИЯ
248	ø	LATIN SMALL LETTER **O** WITH STROKE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **o** СО ШТРИХОМ
249	ù	LATIN SMALL LETTER **U** WITH GRAVE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **u** СО СЛАБЫМ УДАРЕНИЕМ
250	ú	LATIN SMALL LETTER **U** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **u** С СИЛЬНЫМ УДАРЕНИЕМ
251	û	LATIN SMALL LETTER **U** WITH CIRCUMFLEX	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **u** С ЦИРКУМФЛЕКСОМ
252	ü	LATIN SMALL LETTER **U** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **u** С ДИЕРЕЗОМ
253	ý	LATIN SMALL LETTER **Y** WITH ACUTE	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **y** С СИЛЬНЫМ УДАРЕНИЕМ
254	þ	LATIN SMALL LETTER **THORN**	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **þ** (ФОРН)
255	ÿ	LATIN SMALL LETTER **Y** WITH DIAERESIS	СТРОЧНАЯ ЛАТИНСКАЯ БУКВА **y** С ДИЕРЕЗОМ

V.3 Набор 8-битовых графических знаков (версия КОИ-8) по ИСО/МЭК 8859-5

В таблице V.3 приведен набор 8-битовых графических знаков (версия КОИ-8) по ИСО/МЭК 8859-5:1999 (латинский/кирилловский алфавит) и соответствие международных и русских наименований и обозначений знаков. В связи с тем, что знаки указанного набора с десятичными значениями с 0 по 127 полностью совпадают с набором 7-битовых знаков по ИСО 646, в таблице V.3 приводятся 8-битовые графические знаки с десятичными значениями от 160 по 255.

Таблица V.3 — Набор 8-битовых графических знаков (версия КОИ-8) по ИСО/МЭК 8859-5

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
160	NBSP (НПР)	NO-BREAK SPACE	НЕРАЗРЫВАЮЩИЙ ПРОБЕЛ
161	Ё	CYRILLIC CAPITAL LETTER **IO**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ё**
162	Ђ	CYRILLIC CAPITAL LETTER **DJE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ђ**
163	Ѓ	CYRILLIC CAPITAL LETTER **GJE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ѓ**
164	Є	CYRILLIC CAPITAL LETTER UKRAINIAN **IE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Є** (УКРАИНСКИЙ ЯЗЫК)
165	S	CYRILLIC CAPITAL LETTER **DZE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **S**

Продолжение таблицы V.3

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
166	І	CYRILLIC CAPITAL LETTER BYELORUSSIAN-UKRAINIAN **I**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **І** (БЕЛОРУССКИЙ И УКРАИНСКИЙ ЯЗЫКИ)
167	Ї	CYRILLIC CAPITAL LETTER **YI**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ї**
168	Ј	CYRILLIC CAPITAL LETTER **JE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ј**
169	Љ	CYRILLIC CAPITAL LETTER **LJE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Љ**
170	Њ	CYRILLIC CAPITAL LETTER **NJE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Њ**
171	Ћ	CYRILLIC CAPITAL LETTER **TSHE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ћ**
172	Ќ	CYRILLIC CAPITAL LETTER **KJE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ќ**
173	SHY (ГД)	SOFT HYPHEN	ГИБКИЙ ДЕФИС
174	Ў	CYRILLIC CAPITAL LETTER SHORT **U**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ў**
175	Џ	CYRILLIC CAPITAL LETTER **DZHE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Џ**
176	А	CYRILLIC CAPITAL LETTER **A**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **А**
177	Б	CYRILLIC CAPITAL LETTER **BE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Б**
178	В	CYRILLIC CAPITAL LETTER **VE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **В**
179	Г	CYRILLIC CAPITAL LETTER **GHE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Г**
180	Д	CYRILLIC CAPITAL LETTER **DE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Д**
181	Е	CYRILLIC CAPITAL LETTER **IE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Е**
182	Ж	CYRILLIC CAPITAL LETTER **ZHE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ж**
183	З	CYRILLIC CAPITAL LETTER **ZE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **З**
184	И	CYRILLIC CAPITAL LETTER **I**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **И**
185	Й	CYRILLIC CAPITAL LETTER SHORT **I**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Й**
186	К	CYRILLIC CAPITAL LETTER **KA**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **К**
187	Л	CYRILLIC CAPITAL LETTER **EL**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Л**
188	М	CYRILLIC CAPITAL LETTER **EM**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **М**
189	Н	CYRILLIC CAPITAL LETTER **EN**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Н**
190	О	CYRILLIC CAPITAL LETTER **O**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **О**
191	П	CYRILLIC CAPITAL LETTER **PE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **П**
192	Р	CYRILLIC CAPITAL LETTER **ER**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Р**
193	С	CYRILLIC CAPITAL LETTER **ES**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **С**
194	Т	CYRILLIC CAPITAL LETTER **TE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Т**
194	У	CYRILLIC CAPITAL LETTER **U**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **У**
196	Ф	CYRILLIC CAPITAL LETTER **EF**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ф**
197	Х	CYRILLIC CAPITAL LETTER **HA**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Х**
198	Ц	CYRILLIC CAPITAL LETTER **TSE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ц**

Продолжение таблицы V.3

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
199	Ч	CYRILLIC CAPITAL LETTER **CHE**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ч**
200	Ш	CYRILLIC CAPITAL LETTER **SHA**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ш**
201	Щ	CYRILLIC CAPITAL LETTER **SHCHA**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Щ**
202	Ъ	CYRILLIC CAPITAL LETTER **HARD SIGN**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ъ**
203	Ы	CYRILLIC CAPITAL LETTER **YERU**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ы**
204	Ь	CYRILLIC CAPITAL LETTER **SOFT SIGN**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ь**
205	Э	CYRILLIC CAPITAL LETTER **E**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Э**
206	Ю	CYRILLIC CAPITAL LETTER **YU**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Ю**
207	Я	CYRILLIC CAPITAL LETTER **YA**	ПРОПИСНАЯ КИРИЛЛОВСКАЯ БУКВА **Я**
208	а	CYRILLIC SMALL LETTER **A**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **а**
209	б	CYRILLIC SMALL LETTER **BE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **б**
210	в	CYRILLIC SMALL LETTER **VE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **в**
211	г	CYRILLIC SMALL LETTER **GHE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **г**
212	д	CYRILLIC SMALL LETTER **DE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **д**
213	е	CYRILLIC SMALL LETTER **IE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **е**
214	ж	CYRILLIC SMALL LETTER **ZHE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ж**
215	з	CYRILLIC SMALL LETTER **ZE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **з**
216	и	CYRILLIC SMALL LETTER **I**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **и**
217	й	CYRILLIC SMALL LETTER **SHORT I**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **й**
218	к	CYRILLIC SMALL LETTER **KA**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **к**
219	л	CYRILLIC SMALL LETTER **EL**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **л**
220	м	CYRILLIC SMALL LETTER **EM**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **м**
221	н	CYRILLIC SMALL LETTER **EN**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **н**
222	о	CYRILLIC SMALL LETTER **O**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **о**
223	п	CYRILLIC SMALL LETTER **PE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **п**
224	р	CYRILLIC SMALL LETTER **ER**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **р**
225	с	CYRILLIC SMALL LETTER **ES**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **с**
226	т	CYRILLIC SMALL LETTER **TE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **т**
227	у	CYRILLIC SMALL LETTER **U**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **у**
228	ф	CYRILLIC SMALL LETTER **EF**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ф**
229	х	CYRILLIC SMALL LETTER **HA**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **х**
230	ц	CYRILLIC SMALL LETTER **TSE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ц**
231	ч	CYRILLIC SMALL LETTER **CHE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ч**

Окончание таблицы V.3

Десятичное значение	Международное (русское) обозначение знака	Наименование знака	
		международное	русское
232	ш	CYRILLIC SMALL LETTER **SHA**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ш**
233	щ	CYRILLIC SMALL LETTER **SHCHA**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **щ**
234	ъ	CYRILLIC SMALL LETTER **HARD SIGN**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ъ**
235	ы	CYRILLIC SMALL LETTER **YERU**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ы**
236	ь	CYRILLIC SMALL LETTER **SOFT SIGN**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ь**
237	э	CYRILLIC SMALL LETTER **E**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **э**
238	ю	CYRILLIC SMALL LETTER **YU**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ю**
239	я	CYRILLIC SMALL LETTER **YA**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **я**
240	№	NUMERO SIGN	ЗНАК «НОМЕР»
241	ё	CYRILLIC SMALL LETTER **IO**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ё**
242	ђ	CYRILLIC SMALL LETTER **DJE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ђ**
243	ѓ	CYRILLIC SMALL LETTER **GJE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ѓ**
244	є	CYRILLIC SMALL LETTER UKRAINIAN **IE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **є** (УКРАИНСКИЙ ЯЗЫК)
245	ѕ	CYRILLIC SMALL LETTER **DZE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ѕ**
246	і	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN **I**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **і** (БЕЛОРУССКИЙ И УКРАИНСКИЙ ЯЗЫКИ)
247	ї	CYRILLIC SMALL LETTER **YI**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ї**
248	ј	CYRILLIC SMALL LETTER **JE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ј**
249	љ	CYRILLIC SMALL LETTER **LJE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **љ**
250	њ	CYRILLIC SMALL LETTER **NJE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **њ**
251	ћ	CYRILLIC SMALL LETTER **TSHE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ћ**
252	ќ	CYRILLIC SMALL LETTER **KJE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ќ**
253	§	SECTION SIGN	ПАРАГРАФ
254	ў	CYRILLIC SMALL LETTER **SHORT U**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **ў**
255	џ	CYRILLIC SMALL LETTER **DZHE**	СТРОЧНАЯ КИРИЛЛОВСКАЯ БУКВА **џ**

Приложение W
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам

Сведения о соответствии ссылочных международных стандартов национальным стандартам приведены в таблице W.1.

Таблица W.1 — Сведения о соответствии ссылочных международных стандартов национальным и межгосударственным стандартам

Обозначение ссылочного международного стандарта	Обозначение и наименование соответствующего национального и межгосударственного стандарта
ИСО/МЭК 15424	ГОСТ Р 51294.1 — 99 Автоматическая идентификация. Кодирование штриховое. Идентификаторы символик (NEQ)
ИСО/МЭК 19762-1	*
ИСО/МЭК 19762-2	ГОСТ 30721 — 2000/(ГОСТ Р 51294.3 — 99) Автоматическая идентификация. Кодирование штриховое. Термины и определения (NEQ)
ИСО/МЭК 15415	*
ИСО/МЭК 15416	ГОСТ 30832 — 2002 (ИСО/МЭК 15416—2006)/ГОСТ Р 51294.7 — 2001 Автоматическая идентификация. Кодирование штриховое. Линейные символы штрихового кода. Требования к испытаниям качества печати (MOD)
ИСО/МЭК 646:1991	ГОСТ 27463 — 87 Система обработки информации. 7-битные кодированные наборы символов (NEQ)
ИСО/МЭК 8859-1	*
ИСО/МЭК 8859-5:1999	*

* Соответствующий национальный стандарт отсутствует. Оригинал международного стандарта ИСО/МЭК находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице приняты следующие условные обозначения:
- MOD — модифицированный стандарт;
- NEQ — неэквивалентный стандарт.

Библиография

[1] Lin and Costello. Error Control Coding: Foudation and Application. — Prentice Hall, 1983.
[2] C. Britton Rorbaugh. Error Coding Cookbook. — McGrow Hill, 1996.
[3] AIM Inc. Data Matrix Developer's Diskette (AIM Inc., 125 Warrendale-Dayne Road, Suite 100, Warrendale, PA 15086, USA).

УДК 003.295.8:004.223:006.354 ОКС 35.040 П 85

Ключевые слова: автоматическая идентификация, кодирование штриховое, спецификация символики, Data Matrix, матричная символика

Редактор *Т. А. Леонова*
Технический редактор *В. Н. Прусакова*
Корректор *Н. И. Гаврищук*
Компьютерная верстка *З. И. Мартыновой*

Сдано в набор 18.03.2009. Подписано в печать 26.08.2009. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 14.88 + вкл. 2,09. Уч.-изд. л. 13,40 + вкл. 1,91. Тираж 203 экз. Зак. 522.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru
>@gostinfo. ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

Изменение № 1 ГОСТ Р ИСО/МЭК 16022—2008 Автоматическая идентификация. Кодирование штриховое. Спецификация символики Data Matrix

Утверждено и введено в действие Приказом Федерального агентства по техническому регулированию и метрологии от 26.09.2013 № 1112-ст

Дата введения — 2014—01—01

Раздел 9 изложить в новой редакции:

«9 Рекомендуемый алгоритм декодирования для символики Data Matrix

Данный рекомендуемый алгоритм декодирования[1)] позволяет находить на изображении символы Data Matrix и производить их декодирование следующим образом:

а) определяют значения параметров размеров и формируют цифровое изображение:

1) задают расстояние d_{min}, равное 7,5 диаметра апертуры, заданной в системе применения, которое считают минимальной длиной стороны L-образного шаблона поиска;

2) задают расстояние g_{max}, равное 7,5 диаметра апертуры. Это расстояние считают максимальным промежутком в L-образном шаблоне поиска, допускаемым алгоритмом поиска на этапе b);

3) задают расстояние m_{min}, равное 1,25 диаметра апертуры, которое считают номинальным наименьшим размером модуля, когда размер апертуры составляет 80 % размера X символа;

4) формируют черно-белое изображение, используя порог, определенный по методике, установленной в ИСО/МЭК 15415;

б) осуществляют поиск горизонтальной и вертикальной линий сканирования для двух внешних L-образных границ Data Matrix:

1) продлевают горизонтальную линию сканирования по обе стороны от центральной точки изображения и, следуя вдоль этой линии, находят все точки перехода черное/белое и белое/черное. Для каждой точки перехода, найденной на линии сканирования и приведенной к границе пикселя (называемой далее точкой старта), производят следующее:

i) следуют от точки старта вверх вдоль границы перехода черный — белый, пиксель за пикселем до точки, расположенной на расстоянии $3{,}5m_{min}$ от точки старта, или до той точки, в которой линия границы поворачивает вниз;

ii) следуют от точки старта вниз вдоль границы перехода черный — белый, пиксель за пикселем до точки, расположенной на расстоянии $3{,}5m_{min}$ от точки старта, или до той точки, в которой линия границы поворачивает вверх;

iii) если при движении вверх достигли точки, отстоящей на $3{,}5m_{min}$ от точки старта:

I) проводят линию A, соединяющую конечные точки вертикальной границы перехода;

II) проверяют, чтобы отклонение промежуточных точек границы от прямой линии A находилось в пределах $0{,}5m_{min}$. Если это условие выполняется, то продолжают выполнение с этапа, указанного в перечислении III). В противном случае переходят к выполнению этапа, указанного в перечислении I) iv), до достижения края границы перехода в противоположное направление;

III) продолжают следовать вверх вдоль границы перехода до расстояния $0{,}5m_{min}$ от линии A. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии, большем или равном m_{min} от последней точки границы перехода вдоль линии границы перехода и сохраняют ее как конечную точку границы перехода. Эту точку следует рассматривать как одно из предполагаемых положений границ внешнего края L-образных границ;

IV) продолжают следовать вниз вдоль границы перехода до расстояния $0{,}5m_{min}$ от линии A. Возвращаются в ближайшую точку границы перехода, находящуюся на расстоянии, большем или равным m_{min} от последней точки границы перехода вдоль линии границы перехода и сохраняют ее как конечную точку границы перехода. Эта точка должна быть расположена на предполагаемой границе перехода, и ее следует рассматривать как одно из предполагаемых положений границы внешнего края L-образных границ;

V) вычисляют новую откорректированную линию A1, которая является «наиболее приближенной» линией для границы перехода, определенной на двух предыдущих этапах. «Наиболее приближенную» линию вычисляют с использованием алгоритма линейной регрессии (используя конечные точки для выбора зависимой оси, то есть если они ближе к горизонтальной оси, зависимая ось — ось x) для каждой точки. На «наиболее приближенной» прямой линии отмечают отрезок, ограниченный точками p1 и p2, которые являются ближайшими к найденным выше конечным точкам границы перехода;

VI) сохраняют две конечные точки отрезка линии A1 — p1 и p2. Также сохраняют значение цвета левой стороны края границы перехода, видимое при движении от p1 к p2;

[1)] *Разработаны и иные алгоритмы декодирования с аналогичными функциями.*

iv) если этап, указанный в перечислении iii), закончился неудачей или невозможно продолжить движение вниз на $3{,}5m_{min}$ на этапе, указанном в перечислении iii) IV), проверяют, достигнута ли снизу граница перехода на расстоянии $3{,}5m_{min}$ от точки старта. Если да, повторяют операции этапа, указанные в перечислении iii), но не вверх, а вниз;

v) если этапы, указанные в перечислениях iii) и iv), закончились неудачей, проверяют, находятся ли верхняя и нижняя границы перехода на расстоянии не менее $2m_{min}$ от точки старта. Если достигнуты верхняя и нижняя границы перехода, то включают в формируемую границу перехода сегменты вверх и вниз на расстоянии $2m_{min}$ и повторяют операции этапа, указанные в перечислении iii), но с добавлением границы перехода;

vi) повторяют вышеуказанный процесс для следующей точки перехода на линии сканирования, начиная с этапа, указанного в перечислении i), до достижения края изображения;

2) проводят линию сканирования вертикально в обоих направлениях от центральной точки изображения. Находят отрезки линий с использованием той же логической процедуры, что и на этапе, указанном в пункте 1), одновременно следуя от каждой границы перехода символа влево, а затем вправо;

3) среди сохраненных отрезков линий A1 осуществляют поиск пар отрезков, удовлетворяющих следующим четырем условиям:

i) если два отрезка имеют одно и то же направление от p1 до p2, проверяют, что расстояние от точки p1 одного отрезка до точки p2 другого отрезка менее, чем g_{max}; если противоположное, то проверяют, что расстояние между точками p1 и p1 или между точками p2 и p2 разных отрезков менее, чем g_{max};

ii) два вышеуказанных отрезка должны быть параллельными с отклонением не более 5°;

iii) два вышеуказанных отрезка должны быть одного цвета, если отрезки имеют одинаковое направление от p1 к p2, или противоположного цвета, если направления этих отрезков противоположны;

iv) формируют две временные линии, продолжая каждый из двух рассматриваемых отрезков по достижении на их продолжениях точки, ближайшей к конечной точке отрезка другой линии. Проверяют, чтобы обе временные линии были отделены менее чем на $0{,}5m_{min}$ от любой иной точки каждой линии;

4) для каждой пары линий, соответствующих требованиям этапа, указанного в перечислении 3), заменяют эту пару отрезков линий на один удлиненный отрезок линии A1 путем выбора «наиболее приближенной» линии по четырем конечным точкам пары рассматриваемых коротких отрезков линий. Также запоминают значение цвета левой стороны границы перехода новой удлиненной линии, рассматриваемой от конечной точки p1 до конечной точки p2;

5) повторяют этапы, указанные в перечислениях 3) и 4), до тех пор, пока возможно комбинировать пары линии A1;

6) выбирают отрезки линии длиннее d_{min}. Помечают эти линии как предполагаемые L-образные стороны;

7) находят среди полученных пар предполагаемых L-образных сторон две линии, которые должны соответствовать следующим трем критериям:

i) ближайшие точки этих линий должны находиться друг от друга на расстоянии менее $1{,}5g_{max}$;

ii) эти две линии должны быть взаимно перпендикулярны с погрешностью до 5°;

iii) внутренняя сторона угла, образованного этими линиями, должна иметь один и тот же цвет.

Следует иметь ввиду, что если одна или обе линии простираются в обе стороны от точки их пересечения, то два или четыре образованных L-образных шаблона должны быть проверены на соответствие цвету и минимальной длине d_{min} для укороченной стороны или сторон, прежде чем они могут стать предполагаемыми L-образными границами;

8) для каждой пары отрезков — предполагаемых L-образных границ, найденных на этапе, указанном в перечислении 7), формируют предполагаемую L-образную структуру путем продления отрезков до точки их пересечения;

9) если предполагаемая L-образная структура была сформирована из отрезков линий белого цвета внутри угла L-образной структуры, формируют инвертированное по цвету изображение для декодирования. Предпринимают попытки декодировать символ, начиная с нормального или инвертированного изображения, выбирая в качестве начального этап, указанный в перечислении d), используя каждую предполагаемую L-образную структуру, определенную на этапе, указанном в перечислении 8), как L-образный шаблон поиска. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении e);

c) продолжают подбирать отрезки линий A1 и предполагаемые L-образные структуры аналогично предыдущим этапам, также продолжают поиски предполагаемых L-образных структур, используя горизонтальное и вертикальное смещение линий сканирования по отношению к предыдущим линиям сканирования:

1) используя новую горизонтальную линию сканирования, проведенную на расстоянии $3m_{min}$ выше от центральной горизонтальной линии, повторяют в том же порядке действия этапа, указанного в пункте b) 1), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислениях от b) 3) до b) 9). Если декодирование не удалось выполнить, переходят к следующему этапу;

2) используя новую вертикальную линию сканирования, проведенную на расстоянии $3m_{min}$ слева от центральной вертикальной линии сканирования, повторяют действия этапа, указанного в перечислении b) 2), исключая действия, при которых процесс начинается из центральной точки изображения, и этапы, указанные в перечислении от b) 3) до b) 9). Если декодирование не удалось выполнить, переходят к следующему этапу;

3) повторяют действия этапа, указанного в перечислении 1) выше, используя новую горизонтальную линию сканирования, расположенную на расстоянии $3m_{min}$ ниже центральной горизонтальной линии сканирования. Если декодирование не удалось выполнить, повторяют действия этапа, указанного в перечислении 2), но со сдвигом новой вертикальной линии сканирования на $3m_{min}$ вправо от центральной вертикальной линии сканирования. Если декодирование не удалось выполнить, переходят к этапу, указанному в перечислении 4);

4) продолжают производить горизонтальные и вертикальные линии сканирования, как это предусмотрено на этапах, указанных в перечислениях 1) — 3), на $3m_{min}$ вверх, затем влево, затем вниз, затем вправо от ранее произведенных линий сканирования до успешного декодирования символа или до достижения края изображения;

d) первоначально считают, что область-кандидат содержит символ квадратной формы. Если декодировать область как символ квадратной формы не удается, пытаются найти и декодировать символ прямоугольной формы, начиная с этапа, указанного в перечислении j). Для символа квадратной формы сначала формируют нормализованную схему переходов для равных сторон области-кандидата, чтобы найти шаблон поиска с чередующимися модулями:

1) проводят через область-кандидат линию, делящую пополам внутренний угол, образованный L-образными сторонами (рисунок 9). Определяют две равные области, образованные этой разделительной линией (биссектрисой), как левую и правую области со стороны угла L-образной структуры;

2) для каждой области формируют так называемую линию поиска, расположенную на расстоянии d_{min} от вершины угла L-образной структуры и параллельную другой ее стороне, и продолжают эту линию до биссектрисы согласно рисунку 9;

3) сдвигают каждую линию поиска от вершины угла L-образной структуры (рисунок 9), удлиняя каждую линию поиска, чтобы они всегда начинались от стороны угла L-образной структуры и заканчивались на биссектрисе, сохраняя линии поиска параллельными противоположным сторонам угла L-образной структуры. Каждый раз, когда линия поиска сдвигается на один пиксель изображения, подсчитывают число переходов от черного к белому и от белого к черному, начиная и заканчивая подсчет с перехода от цвета стороны L-образной структуры к противоположному цвету. Подсчет переходов следует делать только тогда, когда линия поиска имеет те же самые цвета, что и две линии непосредственно выше и ниже (левее и правее) текущей, и отличается по цвету от предыдущей линии поиска, для которой такой подсчет делался. Вычисляют величину T как число переходов, умноженное на длину наибольшей стороны L-образной структуры и разделенное на текущую длину линии поиска, измеренную между двумя граничными линиями:

T = (число переходов) × (максимальная длина стороны L-образной структуры)/(длина линии поиска).

Эта формула нормализует значение T, предупреждая его увеличение по мере увеличения длины линии поиска.

Продолжают вычислять значения T до тех пор, пока линия поиска не будет длиннее наибольшей оси предполагаемой области-кандидата символа на 50 %;

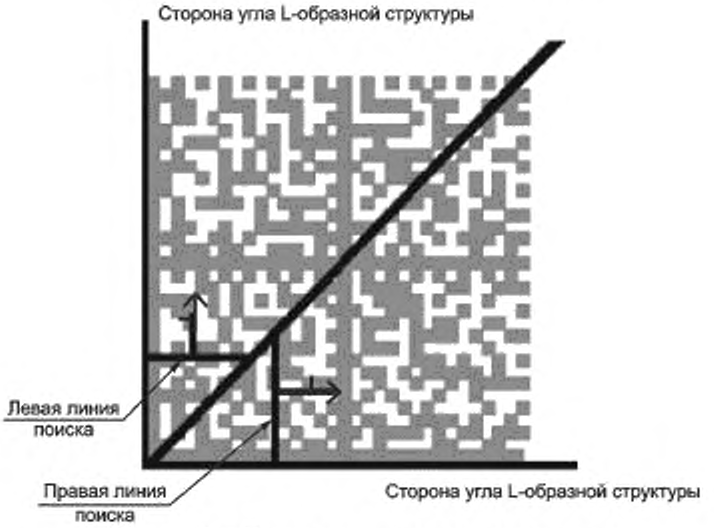


Рисунок 9 — Направления движения линий поиска

4) строят график зависимости T для каждой стороны (области), где на оси ординат Y указано значение T, а на оси абсцисс X — расстояние линии поиска от вершины угла L-образной структуры. Пример графика приведен на рисунке 10;

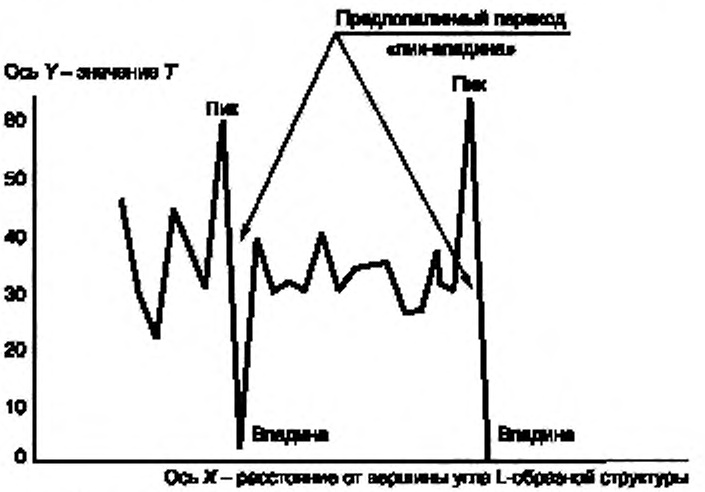


Рисунок 10 — Пример графика зависимости T от увеличения длины линии поиска

5) рассматривают график T для правой стороны (области), начиная с наименьших значений по оси X, с постепенным увеличением значений X по этой оси. Находят первое место резкого падения значения на графике T, где значение T_S (T_S — максимальное значение одной из двух величин — ноль и $T - 1$) менее 15 % значения T в местном локальном максимуме (при условии, что T более 1). Увеличивают это значение X, пока значение T не перестанет уменьшаться. Если в следующей точке значение T не увеличивается, то увеличивают значение X еще раз. Отмечают это значение X как соответствующее впадине. Увеличивают значение X для поиска локального максимума до тех пор, пока число переходов не станет уменьшаться. Отмечают это значение X как соответствующее пику. Значение X ровно посередине между X для впадины и X для пика называют X линии убывания. Линия поиска в точке пика может соответствовать стороне чередующегося шаблона поиска противоположной области. Линия поиска во впадине может соответствовать внутренней части однородной темной линии или светлой свободной зоне;

6) находят пик и впадину на графике для левой стороны (области), *X* линии убывания которых в наибольшей степени подходят по координате *X* линии убывания пика и впадины на графике для правой стороны (области). При возвращении к этому этапу от этапов, указанных ниже, рассматривают дополнительные пики и впадины левой области в порядке увеличения расстояния от пиков и впадин правой области. Однако должны быть просмотрены все пики и впадины левой области, чтобы гарантировать, что разница между значениями *X* для пиков правой и левой области менее чем на 15 % отличается от среднего значения *X* для двух пиков и что разница между значениями *X* для впадин правой и левой области менее чем на 15 % отличается от среднего значения *X* для двух впадин. Значение, равное 15 %, соответствует максимально разрешенному сокращению;

7) линия поиска, соответствующая впадине на графике для правой стороны, линия поиска, соответствующая впадине на графике для левой стороны, и две стороны угла L-образной структуры очерчивают возможную область данных символа. Проводят обработку этой области данных согласно этапу, указанному в перечислении e). Если декодирование выполнено неудачно, бракуют значения для пика и впадины на графике для левой области и продолжают поиск, начиная с этапа, указанного в пункте d) 6), для следующего пика и впадины. Если все пики и впадины левой области были забракованы, бракуют значения для пика и впадины на графике для правой области и продолжают поиск, начиная с этапа, указанного в пункте d) 5), для следующего пика и впадины;

e) для каждой из двух сторон чередующегося шаблона находят линию, проходящую через центры чередующихся темных и светлых модулей:

1) для каждой стороны формируют прямоугольную область, ограниченную линиями поиска для пика и впадины как двумя длинными сторонами прямоугольной области и стороной L-образной структуры и линией для впадины для другой стороны как короткими сторонами прямоугольной области (рисунок 11):

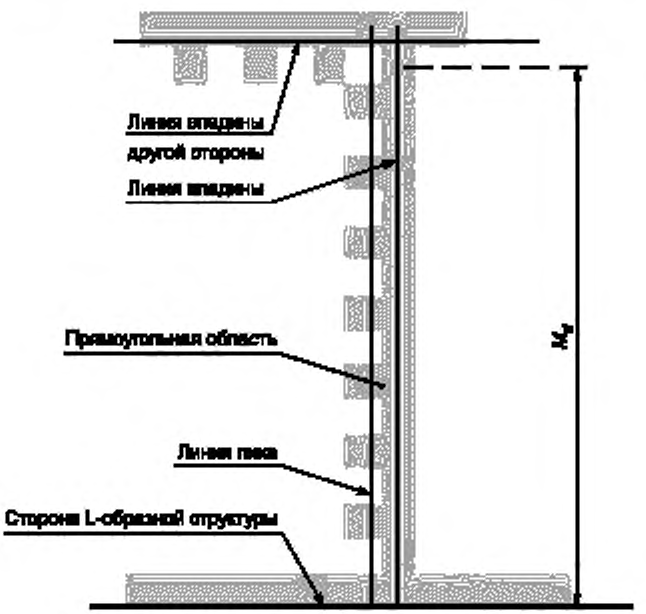


Рисунок 11 — Построение прямоугольной области

2) в пределах этой прямоугольной области находят пары границ между пикселями на стороне с «зубцами»:

i) проводят контрольные линии, параллельные линии впадины, первая из которых совпадает с этой линией, и определяют все переходы цвета в направлении, перпендикулярном к контрольным линиям. Выбирают только переходы от темного к светлому или от светлого к темному, где первый цвет соответствует преобладающему цвету на изображении вдоль линии впадины;

ii) если число найденных цветовых переходов менее, чем 15 % числа пикселей, составляющих линию впадины, и контрольная линия не является линией пика, сдвигают контрольную линию в сторону линии пика приблизительно на один пиксель и повторяют действия этапа, указанного в перечислении i), рассматривая новые переходы в дополнении к уже найденным. Если выполнено условие наличия 15 % числа цветовых переходов к числу пикселей или достигнута линия пика,

переходят к следующему этапу; иначе продолжают поиск, начиная с пункта d) 6) для следующих пиков и впадин левой области;

iii) вычисляют предварительную «наиболее приближенную линию» с помощью алгоритма линейной регрессии, используя точки на границах между выбранными парами пикселей;

iv) отбрасывают 25 % точек, наиболее отдаленных от предварительной «наиболее приближенной линии». Вычисляют окончательную «наиболее приближенную линию» с помощью алгоритма линейной регрессии с использованием оставшихся 75 % точек. Эта линия должна проходить вдоль внешней стороны чередующегося шаблона и указана на рисунке 12 как «наиболее приближенная» линия;

3) для каждой прямоугольной области строят линию, параллельную линии, определенной на этапе, указанном в перечислении e) 2), смещенную в сторону вершины угла L-образной структуры на длину «пиковой» линии поиска, разделенную на удвоенное число переходов на этой линии поиска:

Смещение (*Offset*) = длина линии пика/((число переходов) + 1) × 2).

Каждая из этих двух построенных линий должна соответствовать линии, проведенной через середины модулей внешнего или внутреннего чередующегося шаблона для этой стороны (рисунок 12).

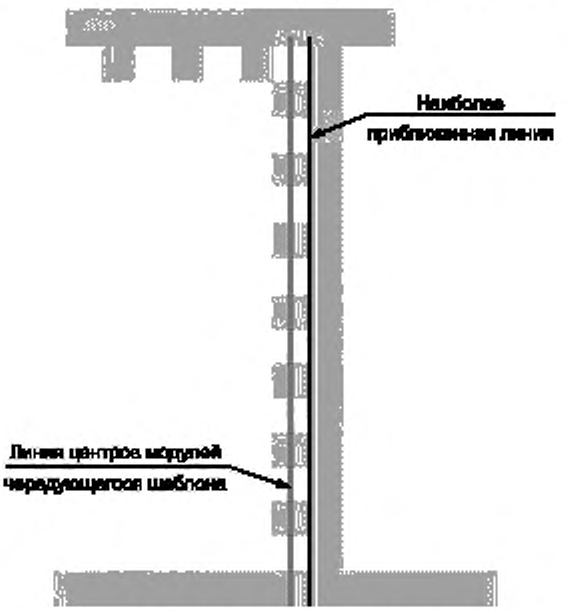


Рисунок 12 — Линия центров модулей чередующегося шаблона

f) для каждой стороны определяют расстояние от края до края в чередующемся шаблоне:

1) ограничивают линию, проходящую через центры модулей чередующегося шаблона, сформированную на этапе, указанном в перечислении e) 3), с одной стороны границей L-образной структуры, а с другой стороны — средней линией чередующегося шаблона, определенной на этапе, указанном в перечислении e) 3). Длину этой линии обозначают M_d (рисунок 11);

2) вдоль ограниченной средней линии измеряют расстояния от края до края между всеми подобными границами всех двухэлементных пар, то есть пар элементов темный/светлый и светлый/темный.

Измерения начинаются и заканчиваются с границы перехода от цвета стороны L-образной структуры к противоположному цвету;

3) вычисляют среднее арифметическое значение всех измерений расстояний от края до края и устанавливают текущую оценку расстояния от края до края *EE_Dist* как это среднее;

4) бракуют все пары элементов, у которых измеренные расстояния от края до края отличаются более чем на 25 % от *EE_Dist*;

g) для каждой стороны находят центральные точки модулей в чередующемся шаблоне:

1) используя измеренные расстояния пар элементов, не забракованные на этапе, указанном в пункте f) 4), вычисляют среднее приращение ширины темного элемента (штриха) при печати (по вертикали или горизонтали в зависимости от стороны сегмента) по формуле как среднее приращение ширины темного элемента (штриха) при печати для пар элементов (темный/светлый или штрих/про-

бел, в которой «штрих» — это ширина темного элемента, а «пробел» — ширина светлого элемента в оставшейся паре элементов):

$$ink_spread = Average\ ((bar - ((bar + space)/2)) / ((bar + space)/2))^{1)};$$

2) вычисляют центр темного элемента (штриха) в паре элементов, занимающих среднее положение, используя следующее смещение в сторону темного элемента (штриха) от внешнего края темного элемента (штриха) в паре элементов, занимающих среднее положение:

$$offset = (EE_Dist \times (1 + ink_spread)) / 4^{2)},$$

Если есть более одной пары элементов, занимающих среднее положение, выбирают единственную пару, используя следующий процесс:

i) просматривают список краев (пар элементов) в порядке увеличения расстояния от границы L-образного шаблона поиска. Количество краев должно быть нечетным числом, поскольку они начинаются и заканчиваются с перехода от темного к светлому, начиная с L-образного шаблона поиска;

ii) средний край в этом списке называют центральным краем;

iii) вычисляют (нечетное число) расстояния от края до края пар элементов и находят их медиану *EE_Dist;*

iv) выбирают одну или более пар элементов с длиной *EE_Dist;*

v) среди этих пар выбирают одну или две пары краев элементов, которые имеют края, самые близкие к центральному краю;

vi) если необходимо, выбирают пару элементов, которые имеют внешний темный край, самый близкий к центральному краю;

vii) если необходимо, выбирают пару элементов, которые имеют внутренний край, самый близкий к центральному краю;

3) начиная от центра темного элемента (штриха) пары элементов, занимающих среднее положение, из этапа, указанного в перечислении f) 3), и продолжая процесс в направлении светлого элемента (пробела) из пары элементов, до конца ограниченной средней линии, вычисляют центр каждого элемента, выделенного белыми точками на темном фоне (рисунок 13) с выполнением следующих действий:

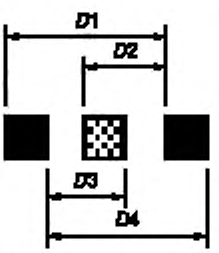


Рисунок 13 — Измерения расстояний от края до края для определения центра каждого элемента

---

[1] *Обозначения в формуле:*
ink_spread — среднее приращение ширины темного элемента (штриха) при печати;
Average — обозначение среднеарифметического значения;
bar — ширина темного элемента (штриха);
space — ширина светлого элемента (пробела).
[2] *Обозначения в формуле:*
offset — смещение;
EE_Dist — среднее арифметическое значение измеренных расстояний от края до края;
ink_spread — среднее приращение ширины темного элемента (штриха) при печати.

Примечание – На рисунке 13 показаны три темных (штриха) и два светлых элемента (пробела). Если элемент, центр которого вычисляют, светлый (пробел), то на схеме должны быть представлены три светлых элемента (пробела) вместо темных (штрихов) и два темных элемента (штриха) вместо светлых (пробелов). Для светлых элементов (пробелов), смежных с конечными элементами средней линии, измерения расстояний *D*1 или *D*4 не проводят, поскольку они оказываются за границами символа или измеряемого сегмента.

i) вычисляют точку p1, находящуюся на средней линии на расстоянии *EE_Dist/2* от предыдущего вычисленного центра элемента в направлении нового элемента;

ii) вычисляют значения $d_1 - d_4$:

$d_1 = D1/2$,

$d_2 = D2$,

$d_3 = D3$,

$d_4 = D4/2$.

iii) если одно из значений $d_1 - d_4$ находится в пределах 25 % от *EE_Dist*, выбирают одно из значений d_i, ближайшее к *EE_Dist*, и устанавливают новое значение *EE_Dist* как среднее между текущим значением *EE_Dist* и выбранным d_i из диапазона $d_1 - d_4$:

I) если выбрано значение d_1 или d_4, определяют соответствующий край *D*1 или *D*4, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние *(ink_spread/2)* × *(EE_Dist/2)* в соответствующем направлении (то есть, если приращение ширины темного элемента (штриха) *ink_spread* положительная величина, смещение края должно быть в сторону светлого элемента (пробела), заключенного в пределах значений *D*1 или *D*4, и, если отрицательная, смещение должно быть в противоположную сторону от светлого элемента). Вычисляют точку p2, находящуюся на средней линии на расстоянии 0,75 выбранного значения d_1 или d_4 от этого смещенного края в сторону элемента, центр которого должен быть вычислен;

II) если выбрано значение d_2 или d_3, определяют соответствующий край *D*2 или *D*3, ближайший к элементу, центр которого необходимо вычислить. Сдвигают этот край на расстояние *(ink_spread/2)* × *(EE_Dist/2)* в соответствующем направлении (то есть, если приращение ширины темного элемента (штриха) при печати является положительным значением, смещение края должно быть проведено в сторону светлого элемента (пробела), заключенного в пределах значений *D*2 или *D*3, и, если отрицательным, смещение должно быть в противоположную сторону от светлого элемента). Вычисляют точку p2 , находящуюся на средней линии на расстоянии 0,25 выбранного значения d_2 или d_3 от смещенного края в сторону элемента, центр которого следует вычислить;

III) считают, что центр элемента находится точно посередине между точками p1 и p2;

iv) в противном случае, если ни одно из значений d_1, d_2, d_3, d_4 не находится в пределах 25 % *EE_Dist*, оставляют текущее значение *EE_Dist*, используют p1 как центр нового элемента и переходят к определению следующего элемента;

4) начиная с темного элемента в паре элементов, занимающих среднее положение, и продолжая в противоположном направлении по отношению к определенному на этапе, указанном в пункте 3), вплоть до окончания ограниченной средней линии, вычисляют центры каждого элемента, используя порядок действий, установленный для этапа, указанного в перечислении 3);

h) если число модулей в каждой стороне не соответствует допустимой первой области, продолжают искать с этапа, указанного в перечислении d) 6), для следующего левого пика и впадины. Иначе составляют пробную сетку модулей данных в области данных, проводя линии из центров модулей шаблона чередующихся модулей:

1) для каждой стороны продолжают каждую линию, построенную на этапе, указанном в перечислении e) 3), и линию противоположной стороны L-образной структуры для формирования точек схода двух почти параллельных линий;

2) из каждой точки схода проводят лучи, проходящие через центры модулей, построенные на этапе, указанном в перечислении g), в направлении, близком к перпендикуляру к линии, полученной на этапе, указанном в перечислении e) 3);

3) точки пересечения этих двух направлений лучей, близких к перпендикулярным, должны соответствовать центрам модулей данных в области данных (рисунок 14);

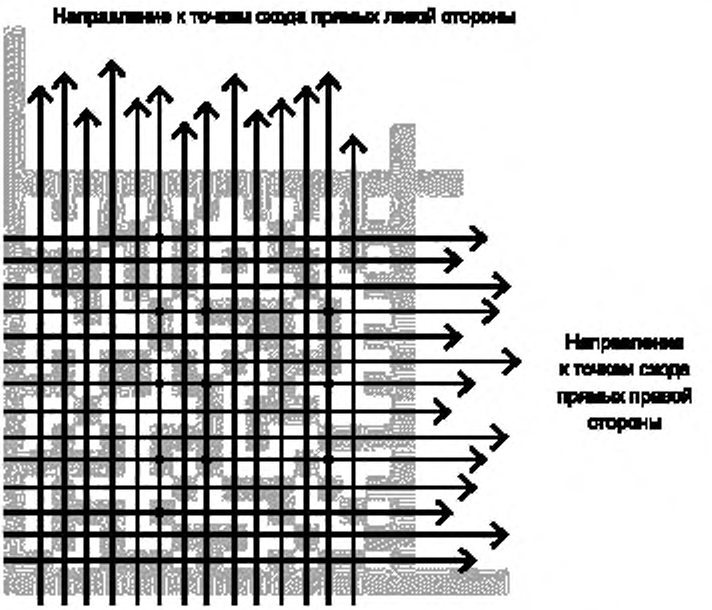


Рисунок 14 — Формирование реальной сетки модулей

i) продолжают заполнение остальных областей данных;

1) в процессе составления области данных формируют новую L-образную структуру части данных левее или выше, используя одну из двух следующих процедур:

i) если новая область данных по-прежнему ограничена с одной стороны исходной L-образной структурой, полученной на этапе, указанном в перечислении b), повторяют этап, указанный в перечислении c), устанавливая новую область данных и используя множество точек, выбранных на этапе, указанном в перечислении e) 2), и множество точек на стороне L-образной структуры из этапа, указанного в перечислении b) 2), которые находятся за пределами линии, полученной на этапе, указанном в перечислении e) 2).

ii) если новая область данных ограничена с двух сторон другими областями данных, повторяют порядок действий с этапа, указанного в перечислении c), для определения новой области данных с помощью множества точек, выбранных на этапе, указанном в перечислении e) 2), для каждой области данных, которая примыкает и ограничивает новую область данных с двух сторон;

2) если область данных не соответствует по числу модулей ранее полученным областям данных, символ корректируют путем его уменьшения до ближайшего большего числа областей, допускаемых стандартом для символа;

3) декодируют символ с одной или несколькими областями данных, начиная с последовательности действий, установленных на этапе, указанном в перечислении k);

4) если в текущей области данных закончили просмотр последнего пика и впадины, возвращаются в предыдущую область данных и продолжают поиск с этапа, указанного в перечислении d) 6) для следующего оставшегося пика и впадины в этой области данных;

j) находят области данных прямоугольного символа:

1) для каждой стороны L-образной структуры передвигают линию поиска, перпендикулярную этой стороне, от вершины угла L-образной структуры, осуществляя сканирование по длине другой стороны L-образной структуры, сохраняя линию поиска параллельной другой стороне L-образной структуры. Каждый раз, когда линия поиска сдвигается на один пиксель изображения, подсчитывают число переходов от черного к белому и от белого к черному, начиная и заканчивая подсчет с перехода от цвета стороны L-образной структуры к противоположному цвету. Подсчет переходов следует делать только тогда, когда линия поиска имеет те же самые цвета, что и две линии непосредственно выше и ниже (левее и правее) текущей, и отличается по цвету от предыдущей линии поиска, для которой такой подсчет делался. Строят график зависимости числа переходов T от расстояния X, на которое сдвигают линию поиска. Продолжают, пока линия не сдвинется на длину противоположной стороны L-образной структуры +10 %;

2) для каждого направления рассматривают графики T, начиная с наименьших значений по оси X, с постепенным увеличением значений X по этой оси. Находят первое место, где значение T_S (T_S — максимальное значение одной из двух величин — ноль и T — 1) станет меньше, чем 15 % предшествующего местного максимума значения T, при условии, что значение T больше единицы. Увеличивают значение X, пока значение T не перестанет уменьшаться. Если в следующей точке значение T не увеличивается, увеличивают значение X еще раз. Отмечают это значение X как соответствующее впадине. Увеличивают значение X для поиска локального максимума до тех пор, пока T не начнет уменьшаться, и отмечают это значение X как соответствующее пику. Точку X посередине между X пика и X впадины отмечают как X линии убывания. Линия впадины в этой точке может формировать сторону символа или его области данных;

3) ищут чередующийся шаблон поиска для каждой области данных, как описано на этапе, указанном в перечислении e);

4) составляют примерную сетку модулей области данных символа, как описано на этапах, указанных в перечислениях i), g) и h);

5) если область данных не является надлежащим прямоугольным символом, формируют новый регион данных, используя следующие пики и впадины;

6) строят все добавочные области данных, как описано на этапе, указанном в перечислении i);

7) если удалось обнаружить правильную область данных или две области, пытаются декодировать символ, как описано на этапах, указанных в перечислениях k) и l). Если область(и) не удается декодировать, исключают эту(и) область-кандидат;

k) если число модулей данных является четным числом или символ имеет надлежащую прямоугольную форму, выполняют его декодирование, используя алгоритм исправления ошибок Рида-Соломона:

1) определяют модули данных в предполагаемых центрах сетки. Темный модуль соответствует единице, светлый модуль — нулю;

2) преобразуют группы по восемь модулей по определенным шаблонам кодовых слов в 8-битовые значения знаков символа;

3) выполняют процедуру исправления ошибок Рида-Соломона с полученными значениями знаков символа;

4) декодируют знаки символа в знаки данных в соответствии с установленными схемами декодирования;

l) если число модулей данных является нечетным числом, то декодируют символ, используя алгоритм сверточного кода исправления ошибок:

1) определяют модули данных в предполагаемых центрах сетки. Темный модуль соответствует единице, светлый модуль — нулю;

2) применяют черно-белую выравнивающую маску;

3) используя соответствующую таблицу расположения битов, преобразуют данные в двоичный поток;

4) затем применяют алгоритм сверточного кода исправления ошибок;

5) преобразуют битовый поток в знаки данных, используя соответствующую схему кодирования;

6) выполняют проверку правильности контрольной суммы CRC.

(ИУС № 1 2014 г.)

gost.e-ecolog.ru