

## まえがき

この規格は、工業標準化法第 14 条によって準用する第 12 条第 1 項の規定に基づき、社団法人電子情報技術産業協会(JEITA)/財団法人日本規格協会(JSA)から、工業標準原案を具して日本工業規格を改正すべきとの申出があり、日本工業標準調査会の審議を経て、経済産業大臣が改正した日本工業規格である。

これによって、**JIS X 0510:1999** は改正され、この規格に置き換えられる。

改正に当たっては、日本工業規格と国際規格との対比、国際規格に一致した日本工業規格の作成及び日本工業規格を基礎にした国際規格原案の提案を容易にするために、**ISO/IEC 18004:2000, Information technology—Automatic identification and data capture techniques—Bar code symbology—QR Code** を基礎として用いた。

この規格に従うことは、次に示す特許権の使用に該当するおそれがある。

発明の名称 二次元コード (特許 2938338 号)

設定登録日 平成 11 年 6 月 11 日

なお、この記載は、上記に示す特許権の効力範囲などに対して何ら影響を与えるものではない。

上記特許の権利者は、日本工業標準調査会に対して、非差別的、かつ、合理的な条件で、いかなる者に対しても当該特許権の実施を許諾する意志があることを保証している。

この規格の一部が、上記に示す以外の技術的性質をもつ特許権、出願公開後の特許出願、実用新案権、又は出願公開後の実用新案登録出願に抵触する可能性がある。経済産業大臣及び日本工業標準調査会は、このような技術的性質をもつ特許権、出願公開後の特許出願、実用新案権又は出願公開後の実用新案登録出願にかかわる確認について、責任はもたない。

**JIS X 0510** には、次に示す附属書がある。

附属書 A (規定) 誤り検出及び訂正の生成多項式

附属書 B (規定) 誤り訂正復号手順

附属書 C (規定) 形式情報

附属書 D (規定) 型番情報

附属書 E (規定) 位置合せパターンの位置

附属書 F (規定) データキャリア識別子

附属書 G (参考) シンボルの符号化例

附属書 H (参考) ビット列の長さの最適化

附属書 I (参考) QR コードシンボルの印刷及び読取りのための利用者手引き

附属書 J (参考) 自動識別能力

附属書 K (規定) マトリックスコードの印刷品質の手引き

附属書 L (参考) プロセス制御技術

附属書 M (規定) QR コードモデル 1 シンボルの特性

附属書 1 (規定) マイクロ QR コード

附属書 2 (参考) **JIS** と対応する国際規格との対比表

## 目 次

	ページ
序文	1
1. 適用範囲	1
2. 適合条件	1
3. 引用規格	1
4. 定義	2
5. 記号	3
6. 記法	3
7. QR コードの仕様	4
7.1 基本的特性	4
7.2 追加機能のまとめ	5
7.3 シンボルの構造	5
8. 要求事項	11
8.1 符号化手順の概要	11
8.2 データ分析	14
8.3 モード	14
8.4 データの符号化	15
8.5 誤り訂正	29
8.6 最終的なメッセージコード語列の構成	37
8.7 コード語の配置	38
8.8 マスク処理	42
8.9 形式情報	45
8.10 型番情報	47
9. 構造的接続	48
9.1 基本原理	48
9.2 シンボル列指示子	48
9.3 パリティデータ	49
10. シンボルの印刷及びマーキング	49
10.1 寸法	49
10.2 文字による表記	49
10.3 マーキングの手引き	49
11. シンボルの品質	49
11.1 試験画像の取得	49
11.2 シンボル品質のパラメタ	50
11.3 シンボル等級の総合評価	50
11.4 プロセス制御の測定	50

12. 復号手順の概要	51
13. QR コードの参照復号アルゴリズム	52
14. 自動識別能力	56
15. 送信データ	57
15.1 データキャリア識別子	57
15.2 拡張チャンネル解釈	57
15.3 FNC1	58
附属書 A (規定) 誤り検出及び訂正の生成多項式	59
附属書 B (規定) 誤り訂正復号手順	64
附属書 C (規定) 形式情報	65
附属書 D (規定) 型番情報	67
附属書 E (規定) 位置合せパターンの位置	70
附属書 F (規定) データキャリア識別子	72
附属書 G (参考) シンボルの符号化例	73
附属書 H (参考) ビット列の長さの最適化	75
附属書 I (参考) QR コードシンボルの印刷及び読取りのための利用者手引き	77
附属書 J (参考) 自動識別能力	79
附属書 K (規定) マトリックスコードの印刷品質の手引き	80
附属書 L (参考) プロセス制御技術	84
附属書 M (規定) QR コードモデル 1 シンボルの特性	86
附属書 1 (規定) マイクロ QR コード	102
附属書 2 (参考) JIS と対応する国際規格との対比表	118

X 0510 : 2004

白 紙

# 二次元コードシンボル—QR コード— 基本仕様

## Two dimensional symbol—QR Code—Basic specification

**序文** この規格は、2000年に第1版として発行された **ISO/IEC 18004**, Information technology—Automatic identification and data capture techniques—Bar code symbology—QR Code を元に、対応する部分については対応国際規格を翻訳し、技術的内容を変更することなく作成した日本工業規格であるが、対応国際規格には規定されていない規定項目を日本工業規格として追加している。

なお、この規格で側線又は点線の下線を施してある箇所は、原国際規格にない事項である。変更の一覧表をその説明を付けて、**附属書 2 (参考)** に示す。

**1. 適用範囲** この規格は、QR コードとして知られる二次元シンボル体系の技術的要件を規定する。ここでは、QR コードのモデル 2 コードシンボル体系並びにマイクロ QR コードシンボル体系の特徴、データキャラクタの符号化、シンボルの形式、寸法特性、誤り訂正規則、参照復号アルゴリズム及びデータキャリア識別子を規定する。また、**附属書 M** に、QR コードのモデル 2 コードシンボル体系と異なる QR コードのモデル 1 コードシンボル体系の特徴を示す。

**備考** この規格の対応国際規格を、次に示す。

なお、対応の程度を表す記号は、**ISO/IEC Guide 21** に基づき、IDT (一致している)、MOD (修正している)、NEQ (同等でない) とする。

**ISO/IEC 18004:2000**, Information technology—Automatic identification and data capture techniques—Bar code symbology—QR Code (MOD)

**2. 適合条件** QR コードシンボル (及び、その生成・読取り用に設計された機器) は、QR コードシンボルのモデル 2, QR コードシンボルのモデル 1 又はマイクロ QR コードシンボルのいずれかに対して規定された要求事項を満足すれば、この規格に合致しているものとみなす。しかし、新規用途又はオープンシステム用途にあっては、QR コードモデル 2 シンボル又はマイクロ QR コードシンボルが推奨シンボル形式であることに注意する。

**3. 引用規格** 次に掲げる規格は、この規格に引用されることによって、この規格の規定の一部を構成する。これらの引用規格は、その最新版 (追補を含む。) を適用する。

**JIS X 0201** 7ビット及び8ビットの情報交換用符号化文字集合

**備考** **ISO/IEC 646** Information technology—ISO 7-bit coded character set for information interchange からの引用事項は、この規格の該当事項と同等である。

**JIS X 0208** 7ビット及び8ビットの2バイト情報交換用符号化漢字集合

**JIS X 0221-1** 国際符号化文字集合 (UCS) — 第 1 部：体系及び基本多言語面

**備考** ISO/IEC 10646-1 Information technology—Universal Multiple-Octet Coded Character Set (UCS) —Part 1: Architecture and Basic Multilingual Plane が、この規格と一致している。

**JIS X 0520** バーコードシンボル印刷品質の評価仕様 — 1次元シンボル

**備考** ISO/IEC 15416 Information technology—Automatic identification and data capture techniques—Bar code print quality test specification—Linear symbols が、この規格と一致している。

**JIS X 0530** データキャリア識別子 (シンボル体系識別子を含む)

**備考** ISO/IEC 15424 Information technology—Automatic identification and data capture techniques—Data Carrier Identifiers (including Symbology Identifiers) が、この規格と一致している。

**CEN EN 1556** Bar Coding—Terminology

**AIM Inc.** Extended Channel Interpretations: Identification Schemes and Protocol

**参考** この規格の対応国際規格 ISO/IEC 18004 は、SC31 及び AIM Inc.が共同開発した。

4. **定義** この規格で用いる主な用語の定義は、CEN EN 1556 によるほか、次による。
- a) **位置合せパターン (alignment pattern)** マトリックスシンボル体系における、決められた位置にある、固定参照パターン。このパターンは、画像にある程度のひずみが生じた場合でも、復号ソフトウェアが画像の座標マップを再編成することを可能にする。
  - b) **文字数指示子 (character count indicator)** モード内におけるデータ文字列の長さを定義するビット列。
  - c) **ECI 指定 (ECI designator)** 特定の ECI 割当てを表す 6 けたの番号。
  - d) **符号化領域 (encoding region)** 機能パターン以外のデータの符号化及び誤り訂正コード語に使用されるシンボル領域。
  - e) **ECI (extended channel interpretation)** いくつかのシンボル仕様で、既定の文字集合の解釈とは異なるデータ列の出力を可能にするために用いられる規約。
  - f) **拡張パターン (extension pattern)** モデル 1 の QR コードシンボルの中でデータを符号化することのない機能パターン。
  - g) **位置検出パターン (finder pattern)** QR コードシンボルの三つの隅に配置される 3 個の位置検出要素パターンから構成される機能パターン。これは、画像領域でシンボル位置の検索を助ける。
  - h) **形式情報 (format information)** シンボルに適用する誤り訂正レベル及び使用するマスク処理パターンに関する情報を持ち、符号化領域を復号するのに必要な符号化パターン。
  - i) **機能パターン (function pattern)** 復号を補助するシンボル位置の検索又は特性の識別に必要なシンボルのオーバーヘッド部分。
  - j) **マスクパターン参照子 (mask pattern reference)** シンボルに適用されるマスク処理パターンのための 3 ビットの識別子。
  - k) **マスク処理 (masking)** 明及び暗のモジュール数を均一化し、画像の高速処理の障害となるパターンの発生を抑えるために行う、符号化領域のビットパターンとマスク処理パターンとの XOR 処理。
  - l) **モード (mode)** ビット列として定義される文字集合の表示方法。
  - m) **モード指示子 (mode indicator)** 次のデータ列がどのモードで符号化されるかを示す 4 ビットの識別子。
  - n) **モジュール (module)** シンボルを構成する単位セル。1 ビットが 1 モジュールに相当する。

参考 通常、モジュールの形状は正方形であるが、印字装置によっては円形などの形状となることがある。

- o) 埋め草コード語 (**pad codeword**) コード語の総数がシンボルの容量に満たない場合、空のコード語位置を充てんする目的で使用する、データを示さない仮のコード語。
- p) 埋め草ビット (**padding bit**) データビット列の終端パターンのある最終コード語の空の位置を充てんする目的で使用する、データを示さないゼロのビット。
- q) 位置検出要素パターン (**position detection pattern**) 位置検出パターンを構成する3個の同一要素の一つ。
- r) 残余ビット (**remainder bit**) 符号化領域が8ビットのシンボル文字で割り切れない場合に、最終シンボル文字のあるシンボル符号化領域の空の位置を充てんする目的で使用するデータを示さないゼロのビット。
- s) 残余コード語 (**remainder codeword**) データ及び誤り訂正コード語の総数が、シンボルの容量を満たさない場合に、シンボルを完成させるために空のコード語位置を充てんするために使用する埋め草コード語。
- t) セグメント (**segment**) ECI モード及び符号化モードの規則に基づいて符号化されるデータ列。
- u) 分離パターン (**separator**) 位置検出パターンをシンボルから分離するために用いる1モジュール幅のすべてが明モジュールの機能パターン。
- v) 構造的接続 (**structured append**) 一つのメッセージを幾つかのシンボルに分割するデータの表示方法。それらのシンボルは、完全なメッセージが再構築される前にすべての読出しを必要とする。また、この場合、メッセージ内でのシンボルの順番及び総数が定義される。
- w) 終端パターン (**terminator**) データを示すビット列の終了に使用する0000のビットパターン。
- x) タイミングパターン (**timing pattern**) シンボル内のモジュール座標を決定する明モジュールと暗モジュールとの交互列。
- y) 型番 (**version**) 21×21モジュール(1型)～177×177モジュール(40型)の間において、番号で示されるシンボルの大きさ。シンボルに適用された誤り訂正レベルを示す場合もある。
- z) 型番情報 (**version information**) モデル2におけるシンボルの型番を表すデータ及びその誤り訂正ビットをもつ符号化パターン。

参考 用語に関しては **JIS X 0500** (データキャリア用語) も参照。

## 5. 記号 公式及び数式で使用する記号は、それらが使用される公式又は数式の直後で定義する。

この規格では、次の演算記号を適用する。

div 整数の除算

mod 除算後の整数剰余

XOR 二つの入力等しくない場合に、出力が“1”となる排他的論理和。記号 ⊕ で表す。

## 6. 記法

- a) モジュール位置 参照を容易にするため、モジュール位置は、シンボルの行及び列の座標 (i, j) で定義する。ここに、i 及び j は、0 から数え始め、モジュールの位置する行 (上から下へ数える。) 及び列 (左から右へ数える。) を示す。したがって、モジュール (0, 0) は、シンボルの左上隅に位置する。
- b) バイト表記 バイトの内容は、16進法で示す。

- c) **型番参照** シンボルの型番は、V-E の形式で参照される。ここに、V は型番号 (1~40) を示し、E は誤り訂正レベル (L, M, Q 及び H) を示す。

参考 L, M, Q 及び H は、7.1 e) 参照。

7. **QR コードの仕様** ここでは、QR コードモデル 2 シンボルに適用できる仕様を規定する。附属書 M で別に規定する場合を除いて、QR コードモデル 1 シンボルにも適用する。

7.1 **基本的特性** QR コードは、次の特性をもつマトリックス式シンボル体系である。

a) **符号化文字集合**

- 1) 数字データ (数字 0~9)
- 2) 英数字データ (数字 0~9, 大文字 A~Z, 9 個の特殊文字: スペース, \$, %, \*, +, -, ., /, :)
- 3) 8 ビットバイトデータ (JIS X 0201 に基づくラテン文字・片仮名用 8 ビット符号の文字集合)
- 4) 漢字データ (JIS X 0208 の附属書 1 のシフト符号化表現で規定された文字。QR コードで符号化する漢字データは、8140<sub>HEX</sub>~9FFC<sub>HEX</sub> 及び E040<sub>HEX</sub>~EBBF<sub>HEX</sub> としてあるので、13 ビットに圧縮することが可能となっている。)

- b) **データ表示** 暗のモジュールは 2 進法の 1, 明のモジュールは 2 進法の 0 を意味する。

- c) **シンボルサイズ** (クワイエットゾーンを除く。) 21×21 モジュール~177×177 モジュール (1 型~40 型, 型番が一つ上がるごとに一辺につき 4 モジュールずつ増加)

- d) **シンボル当たりの文字数** (最大シンボルサイズ, 40-L 型に対して)

- |                |          |
|----------------|----------|
| 1) 数字データ       | 7 089 文字 |
| 2) 英数字データ      | 4 296 文字 |
| 3) 8 ビットバイトデータ | 2 953 文字 |
| 4) 漢字データ       | 1 817 文字 |

e) **選択可能な誤り訂正**

次の 4 段階の誤り訂正が用意しており、それぞれ示してある割合までのシンボルの中の符号語を復元することができる。

- |   |      |
|---|------|
| L | 7 %  |
| M | 15 % |
| Q | 25 % |
| H | 30 % |

- f) **コード形式** マトリックス

- g) **方向の独立性** 独立している。

図 1 に 1 型の QR コードシンボルの 1 例を示す。

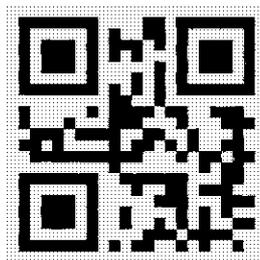


図 1 QR コードシンボルの例

7.2 追加機能のまとめ 次の追加機能は、QR コードに必ず（須）のものか、任意機能のものかのいずれかとする。

- a) 構造的接続（任意） この機能によって、データのファイルを、論理的に連続する最大 16 個までの QR コードシンボルに分けて表現することができる。任意の順序で読み取ってもオリジナルデータを正しく再構成できる。
- b) マスク処理〔必ず（須）〕 復号効率を妨げるようなモジュール配置が現れるのを最小限に抑え、シンボルの明及び暗のモジュール比率を 1 : 1 に近づける。
- c) 拡張チャネル解釈（任意） この機能によって、既定の符号化文字集合でない文字集合（アラビア文字、キリル文字、ギリシャ文字など）を使い別のデータ解釈（定義された圧縮方式を用いた圧縮済みデータ）を使ったデータ、又は他の業界専用の要件を使ったデータの符号化を行うことができる。

7.3 シンボルの構造 各 QR コードシンボルは、正方形様のモジュールを真性正方形配列に並べ、符号化領域及び機能パターン、すなわち、位置検出パターン、分離パターン、タイミングパターン及び位置合せパターンの組合せとして構成する。機能パターンは、データの符号化領域としては使用しない。シンボルは、その四辺の周囲にクワイエットゾーンをもつ。図 2 は 7 型の QR コードシンボルの構造を示す。

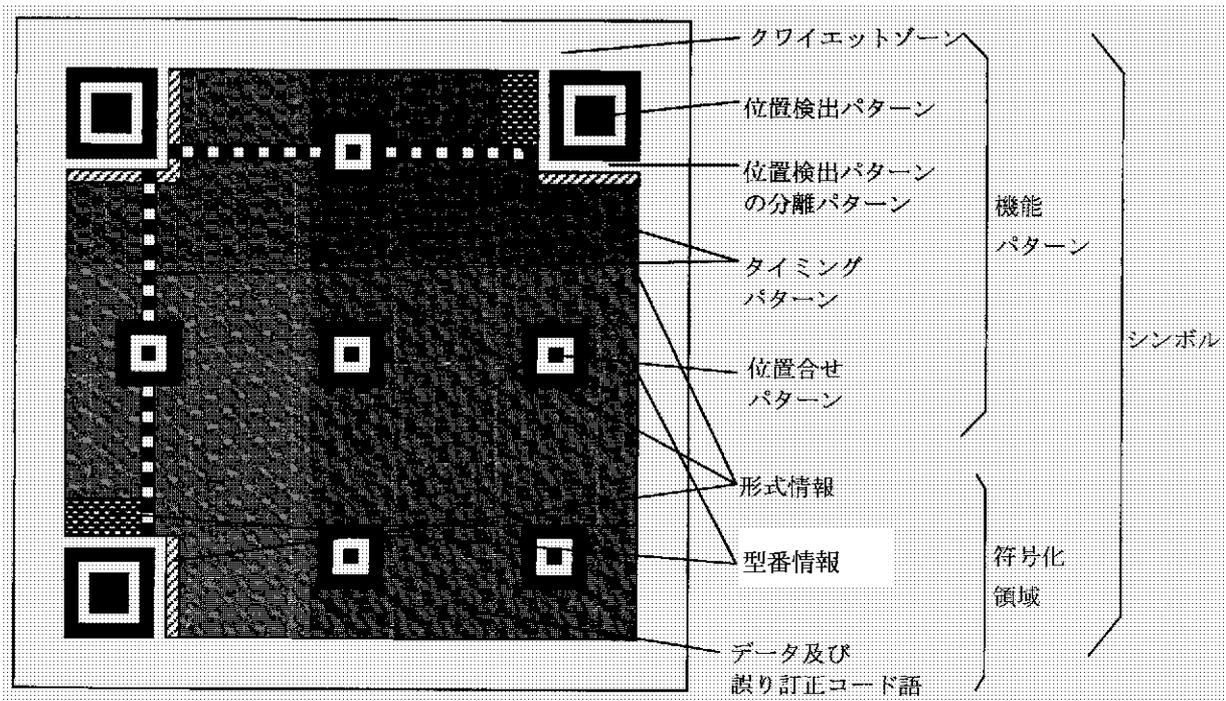
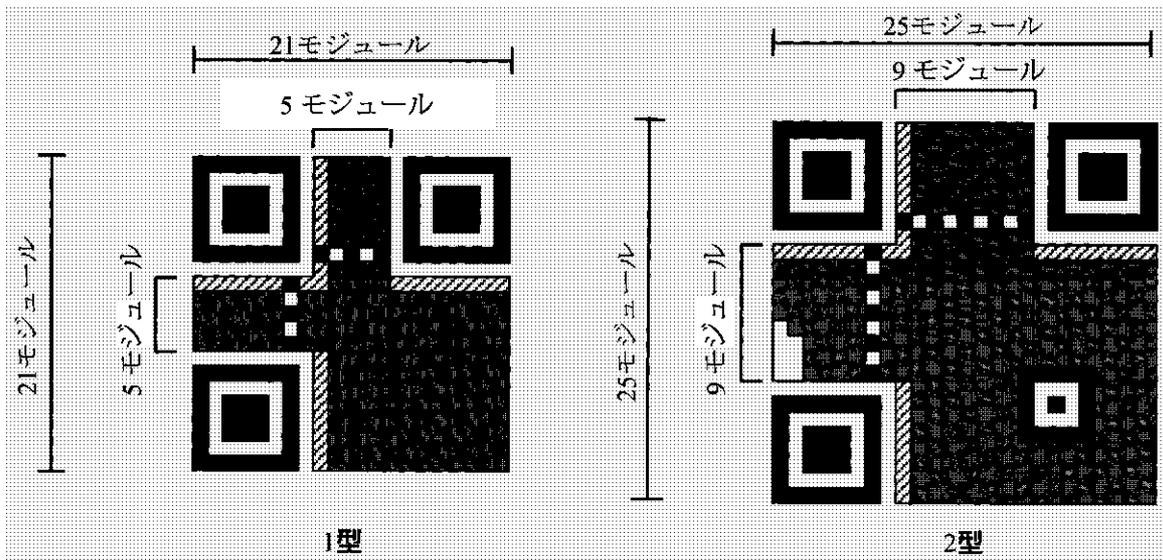


図 2 QR コードシンボルの構造

7.3.1 シンボルの型番及び大きさ QR コードシンボルには、1 型～40 型までの 40 種類の型番がある。1 型は 21×21 モジュール、2 型は 25×25 モジュール、40 型は 177×177 モジュールというように、型番が一つ上がるごとに一辺につき 4 モジュールずつ増加する。図 3～図 8 は、1 型、2 型、6 型、7 型、14 型、21 型及び 40 型の構造を示す。



- 備考
-  データ及び誤り訂正コード語
  -  形式情報及びその誤り訂正符号
  -  型番情報及びその誤り訂正符号
  -  残余ビット

図 3 1型及び2型のシンボル

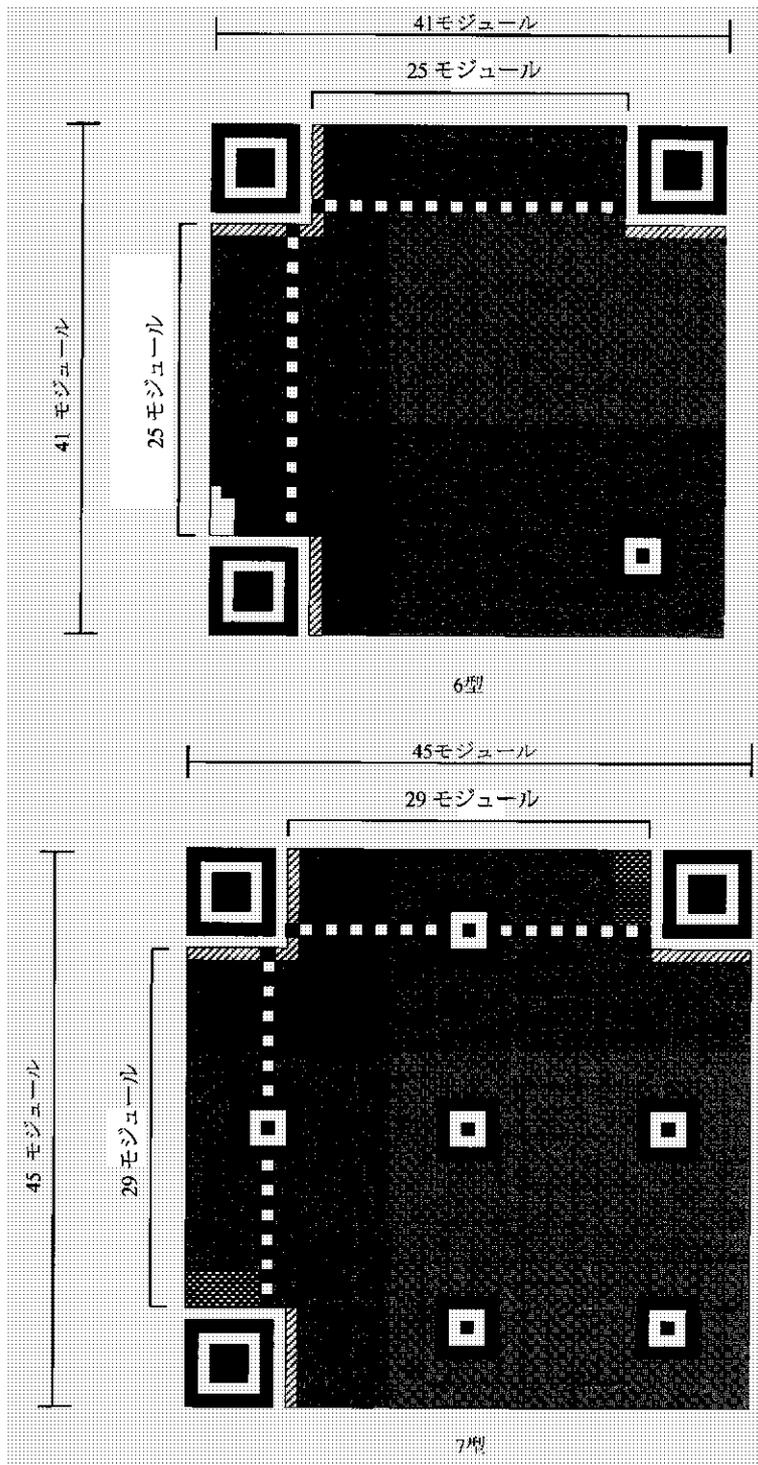


図 4 6 型及び 7 型のシンボル

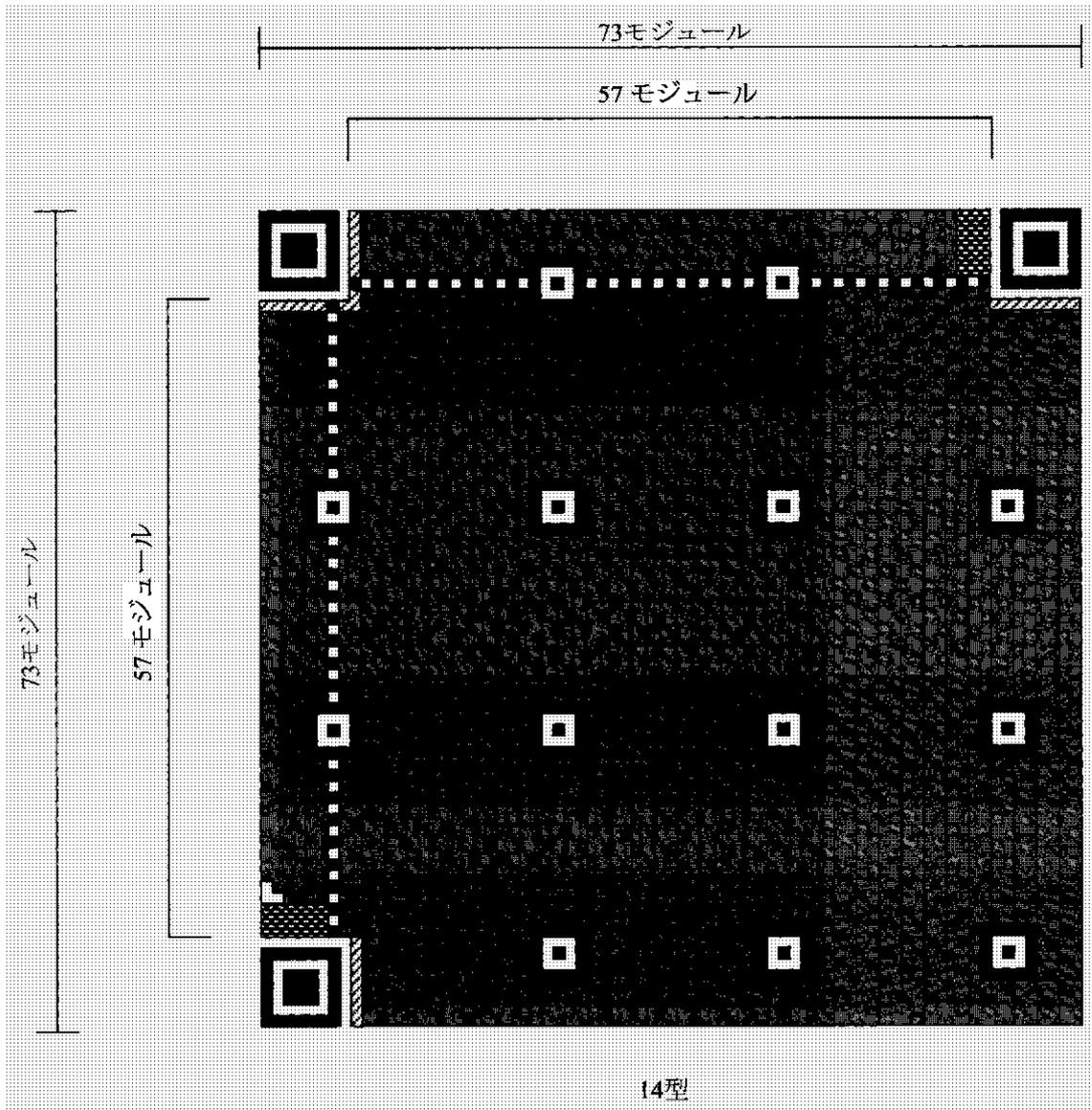


図 5 14 型のシンボル

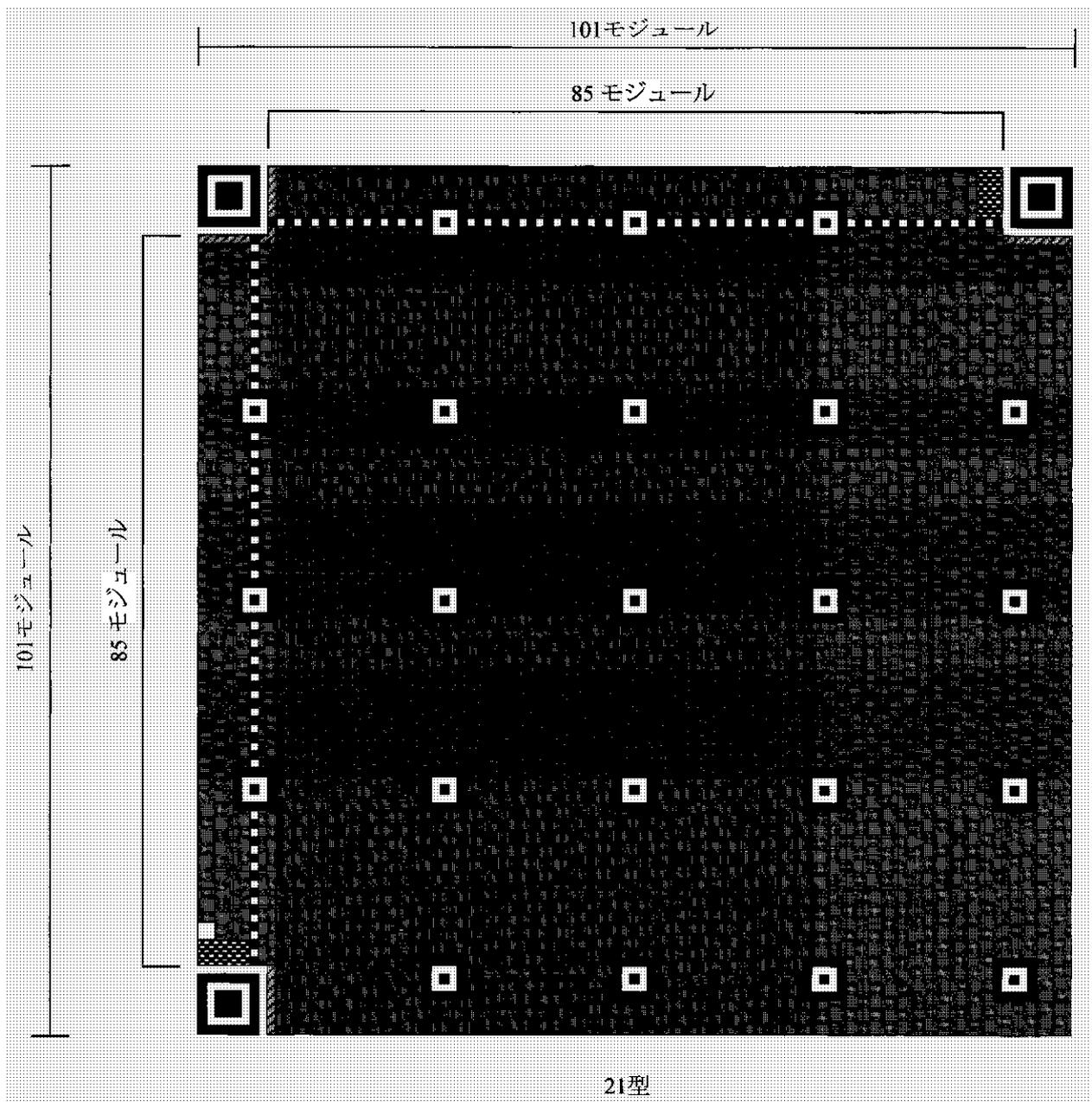


図 6 21 型のシンボル

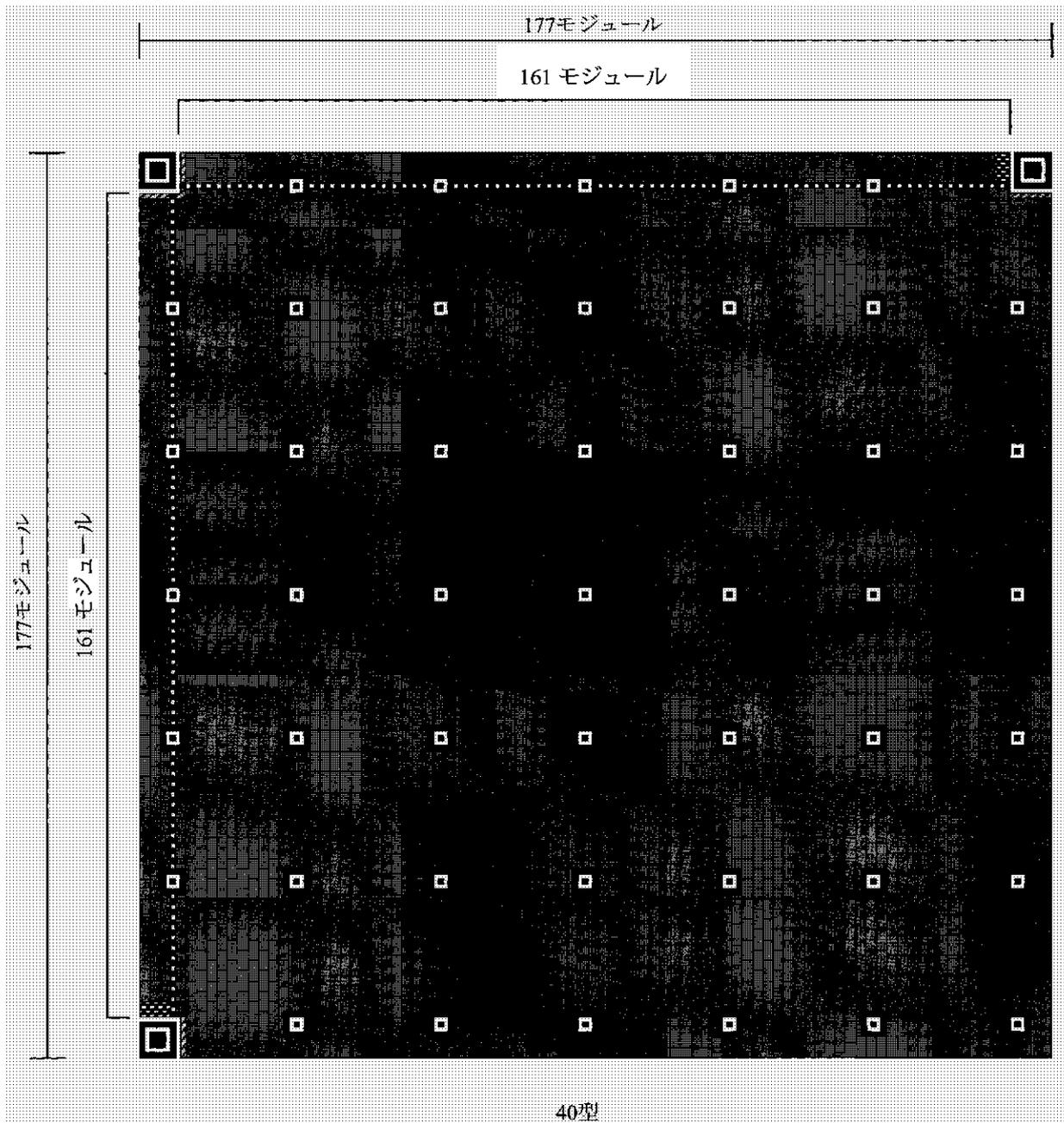


図 7 40 型のシンボル

**7.3.2 位置検出パターン** 位置検出パターンは、図 2 に示すシンボルの左上、右上及び左下に配置される 3 個の位置検出要素パターンから構成する。各位置検出要素パターンは、3 個の同心正方形が重なった形状で、暗の 7×7 モジュール、明の 5×5 モジュール及び暗の 3×3 モジュールから構成される。各位置検出要素パターンにおけるモジュール幅の比率は、図 8 に示すように 1:1:3:1:1 とする。これに類似するパターンがシンボルの任意の位置に出現する可能性を抑え、視野内で QR コードシンボルの認識を容易にすることができるようにシンボルを符号化する。位置検出パターンを構成する 3 個の位置検出要素パターンを識別することによって、視野内でのシンボルの位置及び方向を明確に認識できる。

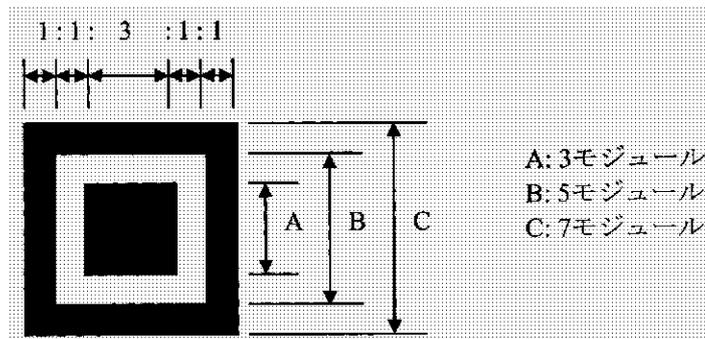


図 8 位置検出要素パターンの構造

**7.3.3 分離パターン** 1 モジュール幅の分離パターンは、図 2 に示すように各位置検出要素パターンと符号化領域との間に配置され、それはすべて明のモジュールで構成される。

**7.3.4 タイミングパターン** 水平及び垂直タイミングパターンは、それぞれ 1 モジュール幅で暗と明とが交互になっているモジュールの行又は列から構成され、暗のモジュールで始まり、暗のモジュールで終了する。水平タイミングパターンは、シンボルの 6 行目、上部位置検出要素パターンに対する分離パターンと分離パターンとの間にわたっている。垂直タイミングパターンは、同様にシンボルの 6 列目、左側の位置検出要素パターンに対する分離パターンと分離パターンとの間にわたっている。これらによって、シンボル密度及び型番を決めることができ、モジュール座標を決めるデータ位置が得られる。

**7.3.5 位置合せパターン** 各位置合せパターンは、3 個の同心正方形が重なった形状で、暗の 5×5 モジュール、明の 3×3 モジュール及び中心となる暗の 1 モジュールで構成される。位置合せパターンの数は、シンボルの型番によって決まり、附属書 E に規定する位置に配置する。

**7.3.6 符号化領域** この領域には、データを示すシンボルキャラクタ、それらに対する誤り訂正コード語、形式情報及び型番情報を含む。シンボルキャラクタの詳細は 8.7.1、形式情報の詳細は 8.9、及び型番情報の詳細は 8.10 による。

**7.3.7 クワイエットゾーン** シンボルの四辺の周囲を囲む、何も表示されない 4 モジュール幅の領域で、その公称反射値は、明のモジュール値と等しい。

**7.3.8 明暗（白黒）反転** この規格には、QR コードシンボルの明モジュール及び暗モジュールを逆にしたシンボルを含める。

**7.3.9 表裏反転** この規格には QR コードシンボルの表裏を逆にしたシンボルを含める。

## 8. 要求事項

**8.1 符号化手順の概要** ここでは、入力データを QR コードシンボルに変換する際に要求する手順の概要を規定する。

- 手順 1 データ分析** 符号化する種々の異なる文字を識別するため、入力データ列を分析する。QR コードは拡張チャネル解釈機能に対応しており、既定の文字集合とは異なるデータを符号化することができる。QR コードには、異なる文字集合を効果的にシンボルキャラクタに変換するための幾つかのモード（8.3 参照）がある。最も効果的にデータを 2 進文字列へ変換するため、必要に応じてモードを切り替える。必要な誤り検出及び誤り訂正レベルを選択する。利用者が使用するシンボル型番を指定していない場合は、データが収容できる最小型番を選択する。シンボル型番及びそのデータ容量の一覧を表 1 に示す。
- 手順 2 データの符号化** 8.4.1～8.4.5 で定義するとおり、適用中のモードの規則に従ってデータ文字をビット列に変換する。新しいモードセグメントの開始時にモード変更に必要なモード指示子を挿入し、データ列の最後に終端パターンを挿入する。得られたビット列を 8 ビットコード語に分割する。その型番が要求するデータコード語数を満たすために必要に応じて埋め草キャラクタを付加する。
- 手順 3 誤り訂正符号** 誤り訂正アルゴリズムを実行するため、コード語列を要求するブロック数（表 13～表 22 で定義）に分割する。各ブロックに対して誤り訂正コード語を生成し、データコード語列の後に付加する。
- 手順 4 最終的なメッセージの構築** 8.6 c) で規定するとおり各ブロックのデータ及び誤り訂正コード語をインタリーブにし、必要に応じて残余ビットを付加する。
- 手順 5 マトリックスにおけるモジュールの配置** マトリックスに位置検出パターン、分離パターン、タイミングパターン及び位置合せパターンとともにコード語モジュールを配置する。
- 手順 6 マスク処理** シンボルの符号化領域にマスク処理パターンを適用する。明及び暗のモジュールバランスを最適にし、また望ましくないパターンの出現を最小限に抑えるパターンを評価し、選択する。
- 手順 7 形式情報及び型番情報** 形式情報及び型番情報（適用する場合）を生成し、シンボルを完成する。

表 1 QR コードの全型番のデータ容量

型番	モジュール数/ 辺(A)	機能パターン モジュール(B)	形式情報及び 型番情報	(C)以外のデータ モジュール	データ容量 [コード語] <sup>(1)</sup> (E)	残余 ビット
	モジュール(C) (D4=A <sup>2</sup> -B-C)					
1	21	202	31	208	26	0
2	25	235	31	359	44	7
3	29	243	31	567	70	7
4	33	251	31	807	100	7
5	37	259	31	1079	134	7
6	41	267	31	1383	172	7
7	45	390	67	1568	196	0
8	49	398	67	1936	242	0
9	53	406	67	2336	292	0
10	57	414	67	2768	346	0
11	61	422	67	3232	404	0
12	65	430	67	3728	466	0
13	69	438	67	4256	532	0
14	73	611	67	4651	581	3
15	77	619	67	5243	655	3
16	81	627	67	5867	733	3
17	85	635	67	6523	815	3
18	89	643	67	7211	901	3
19	93	651	67	7931	991	3
20	97	659	67	8683	1085	3
21	101	882	67	9252	1156	4
22	105	890	67	10068	1258	4
23	109	898	67	10916	1364	4
24	113	906	67	11796	1474	4
25	117	914	67	12708	1588	4
26	121	922	67	13652	1706	4
27	125	930	67	14628	1828	4
28	129	1203	67	15371	1921	3
29	133	1211	67	16411	2051	3
30	137	1219	67	17483	2185	3
31	141	1227	67	18587	2323	3
32	145	1235	67	19723	2465	3
33	149	1243	67	20891	2611	3
34	153	1251	67	22091	2761	3
35	157	1574	67	23008	2876	0
36	161	1582	67	24272	3034	0
37	165	1590	67	25568	3196	0
38	169	1598	67	26896	3362	0
39	173	1606	67	28256	3532	0
40	177	1614	67	29648	3706	0

注<sup>(1)</sup> すべてのコード語は、8ビット長とする。

**8.2 データ分析** 内容を決定するために入力データ文字列を分析し、8.4 に従って各文字列を符号化するために既定又は他の適切な ECI 及び適切なモードを選択する。数字モードから漢字モードに進むにつれ、文字当たりに要求されるビット数が増加する。混在モードでは、データのビット列の長さを最少にするため、シンボル内でモードを切り替えることができ、部分的には、一つのモードで、他の部分より効率的に符号化できる（例えば、数字列の後に英数字列が続く場合）。一般的には、文字当たりの最少ビットを要求するモードでデータを符号化することが最も効果的であるが、モード変更に伴ってモード指示子及び文字数指示子のオーバーヘッドがあるため、文字数の少ない場合にもモードを変更することが常に最短のビット列となるわけではない。これに関する手引きは、**附属書 H** に示す。また、シンボルの容量は、ある型番から次の型番へと段階的に増加するため、すべての事例で最大の変換効率を達成することは、必ずしも必要ではない。

**8.3 モード** 次に定義するモードは、文字値及び既定の ECI 値を伴う割当てに基づいている。他の ECI を適用する場合には、最適なデータ圧縮モードを選択するのに特定の文字割当ての代わりにバイト値を使用する。例えば、データのバイト値が 30<sub>HEX</sub>~39<sub>HEX</sub> の範囲にある文字列の場合には、数字モードを適切とする。この場合、既定の数字又は英字相当のバイト値を用いて圧縮する。

**8.3.1 拡張チャネル解釈モード** ECI プロトコルによって、出力データ列は、省略時の文字集合とは異なる解釈を得ることができる。ECI プロトコルは、多くのシンボル体系に適用できるように定義されている。QR コードでは、次の 4 種類の解釈を用意している。

- a) 各国で使用される文字
- b) 暗号化又は圧縮のような一般目的
- c) クローズシステム向けの利用者定義による解釈
- d) 未編集モードでの構造的接続に対する制御情報

ECI プロトコルは、AIM Inc. 発行の“拡張チャネル解釈の割当て [Extended Channel Interpretation (ECI) Assignments]”において定義する。このプロトコルは、印字前及び復号後のバイト値の特定の解釈に関する一貫した規定方法を示す。

**JIS X 0201** 及び **JIS X 0208** の文字集合を表現する QR コードの既定の ECI 値は、ECI 000020 とする。

**8.3.2 数字モード** 数字モードは、10 進数集合 (0~9) のデータを通常、3 文字を 10 ビットで符号化する。

**8.3.3 英数字モード** 英数字モードは、45 文字 [10 個の数字 (0~9)], 26 個のアルファベット文字 (A~Z) 及び 9 個の記号 (スペース, \$, %, \*, +, -, ., /, :) のデータを符号化する。通常、2 文字を 11 ビットで符号化する。

**8.3.4 8ビットバイトモード** 8ビットバイトモードは、**JIS X 0201** に基づく 8 ビットのラテン文字・片仮名用 8 ビット符号に規定された文字を扱う。このモードにおいて、データは、1 文字を 8 ビットで符号化する。

参考 このモードで使用する符号化文字集合は、QR コードを使用するアプリケーションで指定する符号化文字集合とする。ただし、アプリケーションごとに使用する符号化文字集合が異なると、複数のアプリケーションを扱うシステムでは符号化文字集合の切替が必要となり、システムが複雑になる。このため、国際的なアプリケーションとしては **JIS X 0201**、**ISO 8859-15** 及び **JIS X 0221-1 (UTF-8)** の使用を推奨する。各国国内に限られたアプリケーションでは、**ISO 8859** の各パートの使用を推奨する。日本では **JIS X 0201** の使用を推奨する。UTF-8 は UCS-2 (Universal Multiple-Octet Coded Character Set-2) で定義される文字集合を用いて記述された文字列をバイト

列に変換する方式の一つで、1文字を1~6バイトの可変長マルチバイトに変換する。

**8.3.5 漢字モード** 漢字モードは、JIS X 0208 の附属書 1 で符号化を行う漢字集合（漢字のほかに仮名、英数字等を含む。）を扱う。各2バイトの文字値は、13ビット2進コード語に圧縮する。

**8.3.6 混在モード** QR コードは、8.3.1~8.3.5 に規定する任意のモードの組合せによるデータ列を扱うことができる。

混在モードにおける入力データ文字列の最も効率的な表現方法の選択については、附属書 H による。

**8.3.7 構造的接続モード** 構造的接続モードは、データの符号化を幾つかの QR コードシンボル上に分割するために使用する。すべてのシンボルを読み出すことを要求し、データメッセージを、正しい順序で再構成する。データメッセージの長さ及びシンボルの位置を識別するために、各シンボルの構造的接続ヘッダを符号化し、すべてのシンボルが同じメッセージに属することを検証する。構造的接続モードの符号化の詳細については 9. を参照する。

**8.3.8 FNC1 モード** FNC1 モードは、UCC/EAN アプリケーション識別子規格又は AIM Inc. が認めた特定の工業規格のいずれかに従うデータ形式から構成するメッセージに使用する。

**8.4 データの符号化** 最初の ECI が既定の ECI 値以外の場合は、入力データは、それぞれのモードで1個以上のセグメントが後続する ECI ヘッダで構成されるビット列に変換する。既定の ECI 値では、ビット列は、モード指示子で開始する。

ECI ヘッダ（存在する場合）は、次をもつ。

- ECI モード指示子（4ビット）
- ECI 指定（8ビット、16ビット又は24ビット）

残りのビット列は、次をもつセグメントで構成する。

- モード指示子（4ビット）
- 文字数指示子
- データビット列

ECI ヘッダは、ECI モード指示子の最初（最上位けた）のビットで始まり、ECI 指定の最後（最下位けた）のビットで終了する。各モードセグメントは、モード指示子の最初（最上位けた）のビットで始まり、データビット列の最後（最下位けた）のビットで終了する。セグメントの長さは、適用中のモードの規則及び入力データの文字数によって明確に定義するので、セグメントとセグメントとの間には明確な分離を示すパターンは存在しない。

規定のモードで入力データ列を符号化するには、8.4.1~8.4.6 に定義する手順に従う。表 2 に、各モードのモード指示子を示す。表 3 に、適用するモード及びシンボル型番によって変化する、文字数指示子の長さを定義する。

表 2 モード指示子

モード	指示子
ECI	0111
数字	0001
英数字	0010
8ビットバイト	0100
漢字	1000
FNC1	0101 (1番目の位置) 1001 (2番目の位置)
構造的接続	0011
終端パターン (メッセージの終了)	0000
予約	0110 1010 1011 1100 1101 1110 1111

表 3 文字数指示子のビット数

型番	数字 モード	英数字 モード	8ビットバイト モード	漢字 モード
1~9	10	9	8	8
10~26	12	11	16	10
27~40	14	13	16	12

シンボルにおけるデータの最後は、4ビット終端パターン0000で示す。しかし、データビット列の後ろの残りのシンボル容量が4ビット未満の場合、4ビット終端パターン0000は、省略又は短縮する。

**8.4.1 拡張チャンネル解釈モード** このモードは、この形式のデータを前処理する方法を規定する AIM Inc.の ECI 仕様に従って、バイト値を別の解釈（すなわち別の文字集合）に従うデータを符号化するために用いられ、モード指示子0111によって機能呼び出す。QRコードの既定のECI値(000020)は、**JIS X 0208**のシフト符号化表現の文字集合に対応しており、任意のシンボルの開始時に特にこれを指定する必要はない。

ECIは、データキャリア識別子を送信できる読取装置によってだけ使用できる。データキャリア識別子を送信できない読取装置は、ECIを含むどのシンボルからのデータも送信できない。

入力したECIデータは、一連の8ビットバイト値として符号化システムで処理する。

ECI手順におけるデータは、その意味に関係なく、どのモード又は最も効率的な符号化を行うモードによって符号化してもよい。例えば、30<sub>HEX</sub>~39<sub>HEX</sub>の範囲にあるバイト列は、実際には数字を意味しない場合でも、0~9の数字列のように、数字モード(8.4.2参照)で符号化できる。文字数指示子の値を決定するために、バイト数(漢字モードではバイトの組の数)を用いる。

**8.4.1.1 ECI 指定** ECIは、6けたの割当て番号で指定する。この番号は、ECIモード指示子に続く最初の1個~3個のコード語としてQRコードシンボルに符号化する。符号化規則は、表4で定義する。ECI指定は、5C<sub>HEX</sub>(ISO 646 IRVでは“\”すなわち逆斜線、JIS X 0201では“¥”すなわち円マーク)に続く6けたの割当て番号が符号化されたデータとして出現する。実データとして5C<sub>HEX</sub>が出現する場合には、シンボルに符号化する前の、ECIプロトコルを適用するデータ列の中では5C<sub>HEX</sub>を二重にしなければ

ならない。

復号の際、最初の ECI 割当てコード語 (ECI モードでのモード指示子に続くコード語) の 2 進パターンによって ECI 指定を示すコードワード語の長さが決まる。最初の 0 ビットの前の 1 ビットの数によって、ECI 割当て番号の定義に使用した第 1 コード語の後の追加コードワード数が決まる。最初の 0 ビット後のビット列は、ECI 割当て番号を示す 2 進値とする。ECI の割当て番号が小さいときは、複数の方法で符号化できるが、短い方法が望ましい。

表 4 ECI 割当て番号の符号化

ECI 指定	コード語数	コード語値
000000~000127	1	0bbbbbbb
000000~016383	2	10bbbbbb bbbbbbbb
000000~999999	3	110bbbbbb bbbbbbbb bbbbbbbb
		ここに、b...b=ECI 割当て番号の 2 進値

例 1-H 型シンボルで ISO 8859-7 (ECI 000009) の文字集合を使用し、データをギリシャ文字で符号化する場合

符号化するデータ： **ΑΒΓΔΕ** (文字値 A1<sub>HEX</sub>, A2<sub>HEX</sub>, A3<sub>HEX</sub>, A4<sub>HEX</sub>, A5<sub>HEX</sub>)  
 シンボルのビット列  
 ECI モード指示子： **0111**  
 ECI 割当て番号 (000009)： **00001001**  
 モード指示子 (8 ビットバイト)： **0100**  
 文字数指示子 (5 けた)： **00000101**  
 データ： **10100001 10100010 10100011 10100100 10100101**  
 最終的なビット文字列： **0111 00001001 0100 00000101 10100001 10100010 10100011 10100100 10100101**

復号したデータの送信例については、15.2 参照。

**8.4.1.2 複数の ECI** ECI データセグメントにおける連続する ECI の効果を指定する AIM Inc. の ECI 仕様を参照する。例えば、文字集合 ECI が適用されたデータは、最初の ECI と共存できる非文字集合 ECI を用いて、暗号化又は圧縮することがある。また、二番目の ECI が最初の ECI を無効にし、新たな ECI セグメントを開始する効果をもつことがある。データ中に、ECI 指定が出現するところでは、いずれの ECI 指定も 8.4.1.1 に従って QR コードシンボルに符号化し、新たなモードセグメントを開始する。

**8.4.1.3 ECI 及び構造的接続** 適用する任意の ECI は、8.4.1.2 の規則及び AIM Inc. の ECI 仕様に従って、符号化するデータの終了又は ECI の変更 (モード指示子 0111 による) が発生するまで適用する。ECI において符号化したデータが構造的接続モードで 2 個以上のシンボルにわたる場合、ECI を適用し続けている後続シンボルでは、構造的接続ヘッダの直後に、ECI モード指示子及び適用する各 ECI に対する ECI 指定番号から構成する ECI ヘッダが必要となる。

**8.4.2 数字モード** 入力データ文字列を 3 けたのグループに分割し、各グループを 10 ビットの 2 進値に変換する。入力けた数が 3 の倍数でない場合、最終の 1 けた又は 2 けたは、それぞれ 4 ビット又は 7 ビットに変換する。次に 2 進データを接続し、モード指示子及び文字数指示子を最初の部分に付加する。

数字モードの文字数指示子は、表 3 で定義するとおり 10 ビット、12 ビット又は 14 ビット長とする。入力データ文字数を、10 ビット、12 ビット又は 14 ビットの 2 進値に変換し、モード指示子の後で 2 進データ列の前に付加する。

**例1. 1-H型シンボル**

- 入力データ **01234567**
1. 3けたのグループに分割: **012 345 67**
  2. 各グループを2進値に変換: **012=0000001100**  
**345=0101011001**  
**67 =1000011**
  3. 2進データを順に接続: **0000001100 0101011001 1000011**
  4. 文字数指示子を2進値(1-H型は10ビット)に変換:  
入力データ文字数 **8=0000001000**
  5. モード指示子0001及び文字数指示子を2進データに付加:  
**0001 0000001000 0000001100 0101011001 1000011**

**例2. 1-H型シンボル**

- 入力データ **0123456789012345**
1. 3けたのグループに分割: **012 345 678 901 234 5**
  2. 各グループを2進値に変換: **012=0000001100**  
**345=0101011001**  
**678=1010100110**  
**901=1110000101**  
**234=0011101010**  
**5 =0101**
  3. 2進データを順に接続:  
**0000001100 0101011001 1010100110 1110000101 0011101010 0101**
  4. 文字数指示子を2進値(1-H型は10ビット)に変換:  
入力データ文字数 **16=0000010000**
  5. モード指示子0001及び文字数指示子を2進データに付加:  
**0001 0000010000 0000001100 0101011001 1010100110 1110000101 0011101010**  
**0101**

数字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B=4+C+10(D \text{ DIV } 3)+R$$

ここに、

- B*: ビット列のビット数  
*C*: 文字数指示子のビット数(表3から)  
*D*: 入力データ文字数  
*R*: ( $D \text{ MOD } 3$ )=0ならば0, ( $D \text{ MOD } 3$ )=1ならば4, ( $D \text{ MOD } 3$ )=2ならば7

**8.4.3 英数字モード** 各入力文字に、表5に基づいて0~44の文字値を割り当てる。

表 5 英数字モードの符号化・復号表

文字	値	文字	値	文字	値	文字	値	文字	値	文字	値	文字	値	文字	値
0	0	6	6	C	12	I	18	O	24	U	30	SP	36	.	42
1	1	7	7	D	13	J	19	P	25	V	31	\$	37	/	43
2	2	8	8	E	14	K	20	Q	26	W	32	%	38	:	44
3	3	9	9	F	15	L	21	R	27	X	33	*	39		
4	4	A	10	G	16	M	22	S	28	Y	34	+	40		
5	5	B	11	H	17	N	23	T	29	Z	35	-	41		

入力文字を、2けたのグループに分け、11ビットの2進コードに符号化する。最初の文字の文字値は、45で乗算し、2番目の文字値をそれに加算する。合計値を11ビットの2進値に変換する。入力文字数が2の倍数でない場合、最終文字の文字値は、6ビットの2進値に変換する。これらの2進データは、接続し、モード指示子及び文字数指示子を前に付加する。英数モードの文字数指示子は、表3で定義するとおり9ビット、11ビット又は13ビット長とする。入力データ文字数を、9ビット、11ビット又は13ビットの2進値に変換し、モード指示子の後で2進データ列の前に付加する。

#### 例 1-H型シンボル

入力データ AC-42

- 表6に従って文字値を割当て： AC-42 (10, 12, 41, 4, 2)
- 結果を2けたの10進数のグループに分割： (10, 12) (41, 4) (2)
- 各グループを11ビットの2進値に変換：
 

(10, 12)	$10 \times 45 + 12 = 462$	00111001110
(41, 4)	$41 \times 45 + 4 = 1849$	11100111001
(2)	2	000010
- 2進データを順に接続： 00111001110 11100111001 000010
- 文字数指示子を2進値(1-H型は9ビット)に変換：
 

入力データ文字数 5=000000101
- モード指示子0010及び文字数指示子を2進データに付加：

0010 000000101 00111001110 11100111001 000010

英数字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B = 4 + C + 11(D \text{ DIV } 2) + 6(D \text{ MOD } 2)$$

ここに、

- B: ビット列のビット数  
 C: 文字数指示子のビット数(表3から)  
 D: 入力データ文字数

**8.4.4 8ビットバイトモード** このモードでは、1個の8ビットコード語は、表6に示す入力データ文字に対する文字値を直接割り当てる(密度は、1文字当たり8ビット)。既定のECI以外のECIでは、8ビットバイトの値を直接表す。

表 6 ラテン文字・片仮名用8ビット符号の符号化・復号表

文字 16進												
NUL	00	SP	20	@	40	'	60	80	A0	タ	C0	E0
SOH	01	!	21	A	41	a	61	81	。	A1	チ	E1
STX	02	”	22	B	42	b	62	82	「	A2	ツ	E2
ETX	03	#	23	C	43	c	63	83	」	A3	テ	E3
EOT	04	\$	24	D	44	d	64	84	、	A4	ト	E4
ENQ	05	%	25	E	45	e	65	85	・	A5	ナ	E5
ACK	06	&	26	F	46	f	66	86	ヲ	A6	ニ	E6
BEL	07	'	27	G	47	g	67	87	ア	A7	ヌ	E7
BS	08	(	28	H	48	h	68	88	イ	A8	ネ	E8
HT	09	)	29	I	49	i	69	89	ウ	A9	ノ	E9
LF	0A	*	2A	J	4A	j	6A	8A	エ	AA	ハ	EA
VT	0B	+	2B	K	4B	k	6B	8B	オ	AB	ヒ	EB
FF	0C	,	2C	L	4C	l	6C	8C	ヤ	AC	フ	EC
CR	0D	-	2D	M	4D	m	6D	8D	ユ	AD	ヘ	ED
SO	0E	.	2E	N	4E	n	6E	8E	ヨ	AE	ホ	EE
SI	0F	/	2F	O	4F	o	6F	8F	ツ	AF	マ	EF
DLE	10	0	30	P	50	p	70	90	ー	B0	ミ	F0
DC1	11	1	31	Q	51	q	71	91	ア	B1	ム	F1
DC2	12	2	32	R	52	r	72	92	イ	B2	メ	F2
DC3	13	3	33	S	53	s	73	93	ウ	B3	モ	F3
DC4	14	4	34	T	54	t	74	94	エ	B4	ヤ	F4
NAK	15	5	35	U	55	u	75	95	オ	B5	ユ	F5
SYN	16	6	36	V	56	v	76	96	カ	B6	ヨ	F6
ETB	17	7	37	W	57	w	77	97	キ	B7	ラ	F7
CAN	18	8	38	X	58	x	78	98	ク	B8	リ	F8
EM	19	9	39	Y	59	y	79	99	ケ	B9	ル	F9
SUB	1A	:	3A	Z	5A	z	7A	9A	コ	BA	レ	FA
ESC	1B	;	3B	[	5B	{	7B	9B	サ	BB	ロ	FB
IS4	1C	<	3C	¥	5C		7C	9C	シ	BC	ワ	FC
IS3	1D	=	3D	]	5D	}	7D	9D	ス	BD	ン	FD
IS2	1E	>	3E	^	5E	~	7E	9E	セ	BE	°	FE
IS1	1F	?	3F	_	5F	DEL	7F	9F	ソ	BF	°	FF

参考1. JIS X 0208 の 8 ビット符号のバイト値 80<sub>HEX</sub>~9F<sub>HEX</sub> 及び E0<sub>HEX</sub>~FF<sub>HEX</sub> までは未定義で、保留とする。これらの幾つかは、シフト符号化表現では第 1 バイトとして使用し、8 ビット符号とシフト符号化表現とを区別するのも適用できる。詳細については、JIS X 0208 の附属書 1 (シフト符号化表現) を参照する。

2. JIS X 0208 の 8 ビット符号のバイト値 00<sub>HEX</sub>~FF<sub>HEX</sub> は、5C<sub>HEX</sub> 及び 7E<sub>HEX</sub> を除いて、ISO/IEC 646 に対応している。

次に 2 進データを接続し、モード指示子及び文字数指示子を前に付加する。8 ビットバイトモードの文

字数指示子は、表 3 で定義するとおり 8 ビット又は 16 ビット長とする。入力データ文字数を、8 ビット又は 16 ビット 2 進値に変換し、モード指示子の後で 2 進データ列の前に付加する。

8 ビットバイトモードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B=4+C+8D$$

ここに、

- B: ビット列のビット数
- C: 字数指示子のビット数 (表 3 から)
- D: 入力データ文字数

**8.4.5 漢字モード JIS X 0208** のシフト符号化表現では、漢字は、2 バイトの組合せによって表現する。漢字モードの入力データ文字は、次に示すように、13 ビット 2 進コード語に圧縮する。2 進データを接続し、モード指示子及び字数指示子を前に付ける。漢字モードの字数指示子は、表 3 で定義するとおり 8 ビット、10 ビット又は 12 ビット長とする。入力データ文字数を、8 ビット、10 ビット又は 12 ビットの 2 進値に変換し、モード指示子の後で 2 進データ列の前に付加する。

a) 8140<sub>HEX</sub>～9FFC<sub>HEX</sub> までのシフト符号化表現値の文字

- 1) シフト符号化表現値から 8140<sub>HEX</sub> を減算する。
- 2) 得られた値の上位バイトに C0<sub>HEX</sub> を乗じる。
- 3) 2) の結果に下位バイトを加算する。
- 4) 結果を 13 ビット 2 進文字列に変換する。

参考 JIS X 0208 のシフト符号化表現では、下位バイトは、40<sub>HEX</sub>～FC<sub>HEX</sub> (7F を除く。) の範囲のため、下位バイトがこの範囲以外の場合には、8 ビットバイトモードで符号化する。

b) E040<sub>HEX</sub>～EBBF<sub>HEX</sub> までのシフト符号化表現値の文字

- 1) シフト符号化表現値から C140<sub>HEX</sub> を減算する。
- 2) 得られた値の上位バイトに C0<sub>HEX</sub> を乗じる。
- 3) 2) の結果に下位バイトを加算する。
- 4) 結果を 13 ビット 2 進文字列に変換する。

参考 JIS X 0208 のシフト符号化表現では、下位バイトは、40<sub>HEX</sub>～FC<sub>HEX</sub> (7F を除く。) の範囲のため、下位バイトがこの範囲以外の場合には、8 ビットバイトモードで符号化する。

例

入力文字 (シフト符号化表現値)	“点”	“茗”
	<b>935F</b>	<b>E4AA</b>
1. 8140 又は C140 を引く。	<b>935F - 8140 = 121F</b>	<b>E4AA - C140 = 236A</b>
2. 上位バイトに C0 を乗じる。	<b>12 × C0 = D80</b>	<b>23 × C0 = 1A40</b>
3. 下位バイトを足す。	<b>D80 + 1F = D9F</b>	<b>1A40 + 6A = 1AAA</b>
4. 13 ビット 2 進に変換する。	<b>0D9F =</b>	<b>1AAA =</b>
	<b>0 1101 1001 1111</b>	<b>1 1010 1010 1010</b>

c) 8140<sub>HEX</sub>～9FFC<sub>HEX</sub> までのシフト符号化表現値の文字及び E040<sub>HEX</sub>～EBBF<sub>HEX</sub> までのシフト符号化表現値の文字すべてに対し、入力文字の 2 進列の前にモード指示子 (1000) 及び字数指示子 2 進値 (8, 10 又は 12 ビット) を付加する。

漢字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られ

る。

$$B=4+C+13D$$

ここに、

$B$  : ビット列のビット数

$C$  : 文字数指示子 (表 3 から) のビット数

$D$  : 入力文字数

**8.4.6 混在モード** データの内容によって、あるモードにおけるデータ列を別のモードへ変更する場合又は符号化密度を高める場合、この任意機能を用いる。手引きに関しては、**附属書 H** による。データの各セグメントは、モード指示子、文字数指示子、データ及び次のセグメントを開始するモード指示子によって続く基本構造で、**8.4.1**~**8.4.5** に示す適切なモードで符号化する。**図 9** に、 $n$  個のセグメントを含むデータの構造を図示する。

セグメント 1			セグメント 2			...	セグメント n		
モード	文字数	データ	モード	文字数	データ	...	モード	文字数	データ
指示子 1	指示子		指示子 2	指示子			指示子 n	指示子	

**図 9 混在モードの形式**

**8.4.7 FNC1 モード** 特定のあらかじめ定義された業界仕様又はアプリケーション仕様に従うメッセージ形式を符号化するシンボルを識別するために、**8.3.1**~**8.3.8** 及び **8.4.1**~**8.4.6** で定義するモード及び累積的に使用する二つのモード指示子を用いる。この二つのモード指示子(任意のパラメータデータを伴う)は、データを効率よく符号化するために使用するモード指示子の前に置かれる。これらのモード指示子を使用する場合、復号器は、**15.1** 及び**附属書 F** で定義するデータキャリア識別子を送信する必要がある。

**8.4.7.1 FNC1 (1 番目の位置)** このモード指示子は、UCC/EAN アプリケーション識別子規格に従うデータ形式を符号化するシンボルを認識する。このために、シンボル内で 1 回だけ使用し、効率的なデータ符号化(数字データ、英数字データ、8 ビットバイトデータ及び漢字データ)のために使用する最初のモード指示子の前で、かつ、任意の ECI ヘッダ又は構造的接続ヘッダの後に置く。UCC/EAN 仕様で、データのフィールド分離パターン(つまり、可変長データのフィールドの最後で)として使用する FNC1 (他のシンボルではこの特殊文字を使用する。)を要求する場合、QR コードシンボルは、この機能を実行するために、英数字モードにおける%又は 8 ビットバイトモードにおける IS3 (1D<sub>HEX</sub>)を使用する。%がデータの一部として現れる場合は、%%として符号化する。復号器はシンボル内で%に出会うと 1D<sub>HEX</sub>として送信する。%%の場合は、1 個の%として送信する。

**例1.**

入力データ

**0104912345123459**(アプリケーション識別子 01=UCC/EAN 物品番号, 固定長データ 04912345123459)

**15970331** (アプリケーション識別子 15=最小耐久期日 YYMMDD, 固定長データ 97 年 3 月 31 日)

**30128** (アプリケーション識別子 30=数量, 可変長データ 128) (分離パターン文字が必要)

**10ABC123** (アプリケーション識別子 10=バッチ番号, 可変長データ ABC123)

符号化するデータ

**01049123451234591597033130128%10ABC123**

シンボルのビット列

**0101** [モード指示子 FNC1 (1 番目の位置)]

**0001** (モード指示子 数字モード)

**0000011101** (文字数カウント指示子 29 けた)  
 <01049123451234591597033130128 に対するデータビット>

**0010** (モード指示子英数字モード)

**000001001** (文字数カウント指示子 9 けた)  
 <%10ABC123 に対するデータビット>

送信データ (15.1 及び附属書 F 参照)

]Q301049123451234591597033130128<1D<sub>HEX</sub>>10ABC123

**例 2.** データにおける%の符号化及び送信

入力データ	123%
符号化	123%%
送信	123%

**8.4.7.2 FNC1 (2 番目の位置)** このモード指示子は, AIM Inc. が認めた特定の業界仕様又はアプリケーション仕様に従って形式付けするシンボルを認識する。AIM Inc. によって関係付けされた仕様を識別するアプリケーション指示子の値とする 1 バイトのコード語が後続する。このために, シンボル内で 1 回だけ使用し, 効率的なデータ符号化 (数字データ, 英数字データ, 8 ビットバイトデータ, 漢字データ) のために使用する最初のモード指示子の前で, かつ, 任意の ECI ヘッダ又は構造的接続ヘッダの後に置く。アプリケーション指示子は, (a~z 及び A~Z) の任意のラテンアルファベット文字 (文字の 8 ビット符号値に 100 を加えた) 又は 2 けたの数字 (直接数値で表示) の形式とし, 復号器によって, データの直前に最初の 1 個又は 2 個の文字として送信する。

**例**

(アプリケーション指示子 37 は, この規格の発行時には割当て機関による割当てをしておらず, 例のデータ内容は, 全くの架空とする。)

アプリケーション指示子 37

入力データ AA1234BBB112text text text text<CR>

シンボルのビット列

**1001** [モード指示子 FNC1 (2 番目の位置)]

**00100101** (アプリケーション指示子 37)

**0010** (モード指示子 英数字モード)

**000001100** (文字数カウント指示子 12 けた)

<AA1234BBB112 に対するデータビット>

**0100** (モード指示子 8 ビットバイトモード)

**00010100** (文字数カウント指示子 20 けた)

<text text text text<CR>に対するデータビット>

送信データ

]Q537AA1234BBB112text text text text<CR>

**8.4.8 終端パターン** シンボルのデータの最後は、終端パターン **0000** で示し、最終のモードセグメントに続いてデータビット列に付加する。データビット列がシンボル容量を完全に満たしている場合は終端パターンを省略し、シンボルの残りの容量が 4 ビット未満の場合は短縮する。

**8.4.9 コード語変換に対するビット列** 各モードセグメントに対応するビット列を順番に接続する。データビット列が完全にシンボル容量を満たしていないときは、終端パターンをビット列に付加する。得られたメッセージのビット列を、コード語に分割する。最後のコード語の長さが 8 ビットでないビット列の場合、最後のコード語に 2 進値 0 の埋め草ビットを付加し、8 ビット長にする。埋め草ビットを、データ列の最終ビット（最下位けた）の後に付加する。メッセージビット列は、埋め草コード語 **11101100** 及び **00010001** を交互に付加することによって、表 7～表 10 で定義する、型番及び誤り訂正レベルに対応するシンボルのデータ容量を満たす長さにする。得られた一連のコード語、すなわち、データコード語列は、8.5 で規定するとおり処理し、誤り訂正コード語をメッセージに付加する。シンボルのある型番においては、シンボル容量を完全に満たすためにメッセージの最後に 3 個、4 個又は 7 個の残余ビット（すべて値 0）を付加する必要がある（表 1 参照）。

表 7 1型～10型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8ビット バイト	漢字
1	L	19	152	41	25	17	10
	M	16	128	34	20	14	8
	Q	13	104	27	16	11	7
	H	9	72	17	10	7	4
2	L	34	272	77	47	32	20
	M	28	224	63	38	26	16
	Q	22	176	48	29	20	12
	H	16	128	34	20	14	8
3	L	55	440	127	77	53	32
	M	44	352	101	61	42	26
	Q	34	272	77	47	32	20
	H	26	208	58	35	24	15
4	L	80	640	187	114	78	48
	M	64	512	149	90	62	38
	Q	48	384	111	67	46	28
	H	36	288	82	50	34	21
5	L	108	864	255	154	106	65
	M	86	688	202	122	84	52
	Q	62	496	144	87	60	37
	H	46	368	106	64	44	27
6	L	136	1 088	322	195	134	82
	M	108	864	255	154	106	65
	Q	76	608	178	108	74	45
	H	60	480	139	84	58	36
7	L	156	1 248	370	224	154	95
	M	124	992	293	178	122	75
	Q	88	704	207	125	86	53
	H	66	528	154	93	64	39
8	L	194	1 552	461	279	192	118
	M	154	1 232	365	221	152	93
	Q	110	880	259	157	108	66
	H	86	688	202	122	84	52
9	L	232	1 856	552	335	230	141
	M	182	1 456	432	262	180	111
	Q	132	1 056	312	189	130	80
	H	100	800	235	143	98	60
10	L	274	2 192	652	395	271	167
	M	216	1 728	513	311	213	131
	Q	154	1 232	364	221	151	93
	H	122	976	288	174	119	74

注<sup>(2)</sup> すべてのコード語は、8ビット長とする。

<sup>(3)</sup> データビット数には、モード指示子及び文字数指示子も含む。

表 8 11 型～20 型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8 ビット バイト	漢字
11	L	324	2 592	772	468	321	198
	M	254	2 032	604	366	251	155
	Q	180	1 440	427	259	177	109
	H	140	1 120	331	200	137	85
12	L	370	2 960	883	535	367	226
	M	290	2 320	691	419	287	177
	Q	206	1 648	489	296	203	125
	H	158	1 264	374	227	155	96
13	L	428	3 424	1 022	619	425	262
	M	334	2 672	796	483	331	204
	Q	244	1 952	580	352	241	149
	H	180	1 440	427	259	177	109
14	L	461	3 688	1 101	667	458	282
	M	365	2 920	871	528	362	223
	Q	261	2 088	621	376	258	159
	H	197	1 576	468	283	194	120
15	L	523	4 184	1 250	758	520	320
	M	415	3 320	991	600	412	254
	Q	295	2 360	703	426	292	180
	H	223	1 784	530	321	220	136
16	L	589	4 712	1 408	854	586	361
	M	453	3 624	1 082	656	450	277
	Q	325	2 600	775	470	322	198
	H	253	2 024	602	365	250	154
17	L	647	5 176	1 548	938	644	397
	M	507	4 056	1 212	734	504	310
	Q	367	2 936	876	531	364	224
	H	283	2 264	674	408	280	173
18	L	721	5 768	1 725	1 046	718	442
	M	563	4 504	1 346	816	560	345
	Q	397	3 176	948	574	394	243
	H	313	2 504	746	452	310	191
19	L	795	6 360	1 903	1 153	792	488
	M	627	5 016	1 500	909	624	384
	Q	445	3 560	1 063	644	442	272
	H	341	2 728	813	493	338	208
20	L	861	6 888	2 061	1 249	858	528
	M	669	5 352	1 600	970	666	410
	Q	485	3 880	1 159	702	482	297
	H	385	3 080	919	557	382	235

注<sup>(2)</sup>及び<sup>(3)</sup>は、表 7 の注参照。

表 9 21 型～30 型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8 ビット バイト	漢字
21	L	932	7 456	2 232	1 352	929	572
	M	714	5 712	1 708	1 035	711	438
	Q	512	4 096	1 224	742	509	314
	H	406	3 248	969	587	403	248
22	L	1 006	8 048	2 409	1 460	1 003	618
	M	782	6 256	1 872	1 134	779	480
	Q	568	4 544	1 358	823	565	348
	H	442	3 536	1 056	640	439	270
23	L	1 094	8 752	2 620	1 588	1 091	672
	M	860	6 880	2 059	1 248	857	528
	Q	614	4 912	1 468	890	611	376
	H	464	3 712	1 108	672	461	284
24	L	1 174	9 392	2 812	1 704	1 171	721
	M	914	7 312	2 188	1 326	911	561
	Q	664	5 312	1 588	963	661	407
	H	514	4 112	1 228	744	511	315
25	L	1 276	10 208	3 057	1 853	1 273	784
	M	1 000	8 000	2 395	1 451	997	614
	Q	718	5 744	1 718	1 041	715	440
	H	538	4 304	1 286	779	535	330
26	L	1 370	10 960	3 283	1 990	1 367	842
	M	1 062	8 496	2 544	1 542	1 059	652
	Q	754	6 032	1 804	1 094	751	462
	H	596	4 768	1 425	864	593	365
27	L	1 468	1 744	3 517	2 132	1 465	902
	M	1 128	9 024	2 701	1 637	1 125	692
	Q	808	6 464	1 933	1 172	805	496
	H	628	5 024	1 501	910	625	385
28	L	1 531	12 248	3 669	2 223	1 528	940
	M	1 193	9 544	2 857	1 732	1 190	732
	Q	871	6 968	2 085	1 263	868	534
	H	661	5 288	1 581	958	658	405
29	L	1 631	13 048	3 909	2 369	1 628	1 002
	M	1 267	10 136	3 035	1 839	1 264	778
	Q	911	7 288	2 181	1 322	908	559
	H	701	5 608	1 677	1 016	698	430
30	L	1 735	13 880	4 158	2 520	1 732	1 066
	M	1 373	10 984	3 289	1 994	1 370	843
	Q	985	7 880	2 358	1 429	982	604
	H	745	5 960	1 782	1 080	742	457

注<sup>(2)</sup>及び<sup>(3)</sup>は、表 7 の注参照。

表 10 31型～40型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8ビット バイト	漢字
31	L	1 843	14 744	4 417	2 677	1 840	1 132
	M	1 455	11 640	3 486	2 113	1 452	894
	Q	1 033	8 264	2 473	1 499	1 030	634
	H	793	6 344	1 897	1 150	790	486
32	L	1 955	15 640	4 686	2 840	1 952	1 201
	M	1 541	12 328	3 693	2 238	1 538	947
	Q	1 115	8 920	2 670	1 618	1 112	684
	H	845	6 760	2 022	1 226	842	518
33	L	2 071	16 568	4 965	3 009	2 068	1 273
	M	1 631	13 048	3 909	2 369	1 628	1 002
	Q	1 171	9 368	2 805	1 700	1 168	719
	H	901	7 208	2 157	1 307	898	553
34	L	2 191	17 528	5 253	3 183	2 188	1 347
	M	1 725	13 800	4 134	2 506	1 722	1 060
	Q	1 231	9 848	2 949	1 787	1 228	756
	H	961	7 688	2 301	1 394	958	590
35	L	2 306	18 448	5 529	3 351	2 303	1 417
	M	1 812	14 496	4 343	2 632	1 809	1 113
	Q	1 286	10 288	3 081	1 867	1 283	790
	H	986	7 888	2 361	1 431	983	605
36	L	2 434	19 472	5 836	3 537	2 431	1 496
	M	1 914	15 312	4 588	2 780	1 911	1 176
	Q	1 354	10 832	3 244	1 966	1 351	832
	H	1 054	8 432	2 524	1 530	1 051	647
37	L	2 566	20 528	6 153	3 729	2 563	1 577
	M	1 992	15 936	4 775	2 894	1 989	1 224
	Q	1 426	11 408	3 417	2 071	1 423	876
	H	1 096	8 768	2 625	1 591	1 093	673
38	L	2 702	21 616	6 479	3 927	2 699	1 661
	M	2 102	16 816	5 039	3 054	2 099	1 292
	Q	1 502	12 016	3 599	2 181	1 499	923
	H	1 142	9 136	2 735	1 658	1 139	701
39	L	2 812	22 496	6 743	4 087	2 809	1 729
	M	2 216	17 728	5 313	3 220	2 213	1 362
	Q	1 582	12 656	3 791	2 298	1 579	972
	H	1 222	9 776	2 927	1 774	1 219	750
40	L	2 956	23 648	7 089	4 296	2 953	1 817
	M	2 334	18 672	5 596	3 391	2 331	1 435
	Q	1 666	13 328	3 993	2 420	1 663	1 024
	H	1 276	10 208	3 057	1 852	1 273	784

注<sup>(2)</sup>及び<sup>(3)</sup>は、表7の注参照。

## 8.5 誤り訂正

**8.5.1 誤り訂正能力** QR コードは、データを損失することなく、シンボルが損傷に耐えることができるよう、データコード語列に付加する一連の誤り訂正コード語を生成するリードソロモン誤り訂正を備えている。誤り訂正には、表 11 に示すように利用者が選択できる 4 段階のレベルがあり、次のように損傷の度合いに応じた復元能力が得られる。

表 11 誤り訂正レベル

誤り訂正レベル	復元能力% (概数)
L	7
M	15
Q	25
H	30

附属書 I の I.3 に、シンボルに適用する誤り訂正の適正レベルに関する手引きを示す。

誤り訂正コード語は、2 種類の誤りコード語、すなわち、棄却誤り（位置が分かっている誤りコード語）及び代入誤り（位置が分からない誤りコード語）を訂正することができる。棄却誤りは、未走査又は復号不能のシンボルキャラクタとする。代入誤りは、間違って復号したシンボルキャラクタとする。QR コードは、マトリックス式シンボル体系のため、暗から明へ又は明から暗へのモジュール変換での不具合は、シンボルキャラクタとしては有効であるが、異なるコード語に復号する結果になる。データ内におけるそのような代入誤りを訂正するには、2 個の誤り訂正コード語が必要となる。

訂正可能な棄却誤りと代入誤りとの数は、次の式で得られる。

$$e + 2t \leq d - p$$

ここに、

- $e$  : 棄却誤り数
- $t$  : 代入誤り数
- $d$  : 誤り訂正コード語数
- $p$  : 復号誤り防止コード語数

例えば、6-H 型には全部で 172 のコード語があり、そのうちの 112 が誤り訂正コード語とする（残りの 60 は、データコード語とする。）。112 個の誤り訂正コード語は、56 の復号誤り又は代入誤りを訂正することができる（すなわち、56/172 又はシンボル容量の 32.6%）。

上の式において、1-L 型のシンボルでは  $p=3$ 、1-M 型及び 2-L 型シンボルでは  $p=2$ 、1-Q 型、1-H 型及び 3-L 型シンボルでは  $p=1$ 、その他の場合では  $p=0$  となる。 $p>0$  の場合、誤り検出コード語として機能し、誤り数が誤り訂正能力を超える場合に、シンボルからのデータ送信を防止するコード語が  $p$ （1、2 又は 3）個となり、 $e$  は  $d/2$  未満となる。例えば、2-L 型において、コード語の合計が 44 の場合、そのうち 34 は、データコード語で、10 が誤り訂正コード語となる。表 12 から、 $e=0$  とした場合、誤り訂正能力は、4 であることがわかる。上の式に代入すると、次のとおりになる。

$$0 + (2 \times 4) = 10 - 2$$

この式は、4 個の誤り訂正には 8 個の誤り訂正コード語が必要で、残りの 2 個の誤り訂正コード語はそれに加えて他の誤りを検出（訂正ではない）することができ、誤りが 4 個を超える場合、復号できないことを示す。

型番及び誤り訂正レベルによっては、データコード語列を一つ以上の RS（リードソロモン）ブロックに分ける。それぞれの RS ブロックに対し、誤り訂正アルゴリズムを別々に適用する。表 12～表 16 に、各型

番及び誤り訂正レベルについて、コード語の総数、誤り訂正コード語の総数、RS ブロックの構造及び数を示す。

シンボル容量の残りを満たすために残余ビットが必要な型番に対しては、それらのビットは、すべて 0 とする。

表 12 1 型～8 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (*)
1	26	L	7	1	(26, 19, 2) (♫)
		M	10	1	(26, 16, 4) (♫)
		Q	13	1	(26, 13, 6) (♫)
		H	17	1	(26, 9, 8) (♫)
2	44	L	10	1	(44, 34, 4) (♫)
		M	16	1	(44, 28, 8)
		Q	22	1	(44, 22, 11)
		H	28	1	(44, 16, 14)
3	70	L	15	1	(70, 55, 7) (♫)
		M	26	1	(70, 44, 13)
		Q	36	2	(35, 17, 9)
		H	44	2	(35, 13, 11)
4	100	L	20	1	(100, 80, 10)
		M	36	2	(50, 32, 9)
		Q	52	2	(50, 24, 13)
		H	64	4	(25, 9, 8)
5	134	L	26	1	(134, 108, 13)
		M	48	2	(67, 43, 12)
		Q	72	2	(33, 15, 9)
				2	(34, 16, 9)
		H	88	2	(33, 11, 11)
				2	(34, 12, 11)
6	172	L	36	2	(86, 68, 9)
		M	64	4	(43, 27, 8)
		Q	96	4	(43, 19, 12)
		H	112	4	(43, 15, 14)
7	196	L	40	2	(98, 78, 10)
		M	72	4	(49, 31, 9)
		Q	108	2	(32, 14, 9)
				4	(33, 15, 9)
		H	130	4	(39, 13, 13)
				1	(40, 14, 13)
8	242	L	48	2	(121, 97, 12)
		M	88	2	(60, 38, 11)
				2	(61, 39, 11)
		Q	132	4	(40, 18, 11)
				2	(41, 19, 11)
		H	156	4	(40, 14, 13)
2	(41, 15, 13)				

注(\*)  $(c, k, r)$   $c$ =総コード語数 [表 1 (E) を参照],  $k$ =データコード語数,  $r$ =誤り訂正数

(♫) 誤り訂正数は、復号誤りの可能性を低減するために誤り訂正コード語数の半分未満となっている。

表 13 9型～14型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注)
9	292	L	60	2	(146, 116, 15)
				3	(58, 36, 11)
		M	110	2	(59, 37, 11)
				4	(36, 16, 10)
		Q	160	4	(37, 17, 10)
				4	(36, 12, 12)
H	192	4	(37, 13, 12)		
		4			
10	346	L	72	2	(86, 68, 9)
				2	(87, 69, 9)
		M	130	4	(69, 43, 13)
				1	(70, 44, 13)
		Q	192	6	(43, 19, 12)
				2	(44, 20, 12)
		H	224	6	(43, 15, 14)
				2	(44, 16, 14)
11	404	L	80	4	(101, 81, 10)
				1	(80, 50, 15)
		M	150	4	(81, 51, 15)
				4	(50, 22, 14)
		Q	224	4	(51, 23, 14)
				4	(36, 12, 12)
		H	264	3	(37, 13, 12)
				8	
12	466	L	96	2	(116, 92, 12)
				2	(117, 93, 12)
		M	176	6	(58, 36, 11)
				2	(59, 37, 11)
		Q	260	4	(46, 20, 13)
				6	(47, 21, 13)
		H	308	7	(42, 14, 14)
				4	(43, 15, 14)
13	532	L	104	4	(133, 107, 13)
				8	(59, 37, 11)
		M	198	1	(60, 38, 11)
				8	(44, 20, 12)
		Q	288	4	(45, 21, 12)
				12	(33, 11, 11)
		H	352	4	(34, 12, 11)
				4	
14	581	L	120	3	(145, 115, 15)
				1	(146, 116, 15)
		M	216	4	(64, 40, 12)
				5	(65, 44, 12)
		Q	320	11	(36, 16, 10)
				5	(37, 17, 10)
		H	384	11	(36, 12, 12)
				5	(37, 13, 12)

注(注)は、表 12 の注参照。

表 14 15型～19型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RSブロック数	RSブロック <sup>(*)</sup>
15	655	L	132	5	(109, 87, 11)
				1	(110, 88, 11)
		M	240	5	(65, 41, 12)
				5	(66, 42, 12)
Q	360	5	(54, 24, 15)		
		7	(55, 25, 15)		
H	432	11	(36, 12, 12)		
		7	(37, 13, 12)		
16	733	L	144	5	(122, 98, 12)
				1	(123, 99, 12)
		M	280	7	(73, 45, 14)
				3	(74, 46, 14)
Q	408	15	(43, 19, 12)		
		2	(44, 20, 12)		
H	480	3	(45, 15, 15)		
		13	(46, 16, 15)		
17	815	L	168	1	(135, 107, 14)
				5	(136, 108, 14)
		M	308	10	(74, 46, 14)
				1	(75, 47, 14)
Q	448	1	(50, 22, 14)		
		15	(51, 23, 14)		
H	532	2	(42, 14, 14)		
		17	(43, 15, 14)		
18	901	L	180	5	(150, 120, 15)
				1	(151, 121, 15)
		M	338	9	(69, 43, 13)
				4	(70, 44, 13)
Q	504	17	(50, 22, 14)		
		1	(51, 23, 14)		
H	588	2	(42, 14, 14)		
		19	(43, 15, 14)		
19	991	L	196	3	(141, 113, 14)
				4	(142, 114, 14)
		M	364	3	(70, 44, 13)
				11	(71, 45, 13)
Q	546	17	(47, 21, 13)		
		4	(48, 22, 13)		
H	650	9	(39, 13, 13)		
		16	(40, 14, 13)		

注(\*)は、表 12 の注参照。

表 15 20 型～25 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注)
20	1 085	L	224	3	(135, 107, 14)
				5	(136, 108, 14)
		M	416	3	(67, 41, 13)
				13	(68, 42, 13)
Q	600	15	(54, 24, 15)		
		5	(55, 25, 15)		
H	700	15	(43, 15, 14)		
		10	(44, 16, 14)		
21	1 156	L	224	4	(144, 116, 14)
				4	(145, 117, 14)
		M	442	17	(68, 42, 13)
				6	(50, 22, 14)
Q	644	17	(50, 22, 14)		
		6	(51, 23, 14)		
H	750	19	(46, 16, 15)		
		6	(47, 17, 15)		
22	1 258	L	252	2	(139, 111, 14)
				7	(140, 112, 14)
		M	476	17	(74, 46, 14)
				7	(54, 24, 15)
Q	690	16	(55, 25, 15)		
		34	(37, 13, 12)		
23	1 364	L	270	4	(151, 121, 15)
				5	(152, 122, 15)
		M	504	4	(75, 47, 14)
				14	(76, 48, 14)
Q	750	11	(54, 24, 15)		
		14	(55, 25, 15)		
H	900	16	(45, 15, 15)		
		14	(46, 16, 15)		
24	1 474	L	300	6	(147, 117, 15)
				4	(148, 118, 15)
		M	560	6	(73, 45, 14)
				14	(74, 46, 14)
Q	810	11	(54, 24, 15)		
		16	(55, 25, 15)		
H	960	30	(46, 16, 15)		
		2	(47, 17, 15)		
25	1 588	L	312	8	(132, 106, 13)
				4	(133, 107, 13)
		M	588	8	(75, 47, 14)
				13	(76, 48, 14)
Q	870	7	(54, 24, 15)		
		22	(55, 25, 15)		
H	1 050	22	(45, 15, 15)		
		13	(46, 16, 15)		

注(注)は、表 12 の注参照。

表 16 26 型～30 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注 <sup>*)</sup> )
26	1 706	L	336	10	(142, 114, 14)
				2	(143, 115, 14)
		M	644	19	(74, 46, 14)
				4	(75, 47, 14)
Q	952	28	(50, 22, 14)		
		6	(51, 23, 14)		
H	1 110	33	(46, 16, 15)		
		4	(47, 17, 15)		
27	1 828	L	360	8	(152, 122, 15)
				4	(153, 123, 15)
		M	700	22	(73, 45, 14)
				3	(74, 46, 14)
Q	1 020	8	(53, 23, 15)		
		26	(54, 24, 15)		
H	1 200	12	(45, 15, 15)		
		28	(46, 16, 15)		
28	1 921	L	390	3	(147, 117, 15)
				10	(148, 118, 15)
		M	728	3	(73, 45, 14)
				23	(74, 46, 14)
Q	1 050	4	(54, 24, 15)		
		31	(55, 25, 15)		
H	1 260	11	(45, 15, 15)		
		31	(46, 16, 15)		
29	2 051	L	420	7	(146, 116, 15)
				7	(147, 117, 15)
		M	784	21	(73, 45, 14)
				7	(74, 46, 14)
Q	1 140	1	(53, 23, 15)		
		37	(54, 24, 15)		
H	1 350	19	(45, 15, 15)		
		26	(46, 16, 15)		
30	2 185	L	450	5	(145, 115, 15)
				10	(146, 116, 15)
		M	812	19	(75, 47, 14)
				10	(76, 48, 14)
Q	1 200	15	(54, 24, 15)		
		25	(55, 25, 15)		
H	1 440	23	(45, 15, 15)		
		25	(46, 16, 15)		

注(\*)は、表 12 の注参照。

表 17 31 型～35 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注 <sup>4</sup> )
31	2 323	L	480	13	(145, 115, 15)
				3	(146, 116, 15)
		M	868	2	(74, 46, 14)
				29	(75, 47, 14)
Q	1 290	42	(54, 24, 15)		
		1	(55, 25, 15)		
H	1 530	23	(45, 15, 15)		
		28	(46, 16, 15)		
32	2 465	L	510	17	(145, 115, 15)
				10	(74, 46, 14)
		M	924	23	(75, 47, 14)
				10	(54, 24, 15)
Q	1 350	35	(55, 25, 15)		
		19	(45, 15, 15)		
H	1 620	35	(46, 16, 15)		
		17	(145, 115, 15)		
33	2 611	L	540	1	(146, 116, 15)
				14	(74, 46, 14)
		M	980	21	(75, 47, 14)
				29	(54, 24, 15)
Q	1 440	19	(55, 25, 15)		
		11	(45, 15, 15)		
H	1 710	46	(46, 16, 15)		
		L	570	13	(145, 115, 15)
M	1 036			6	(146, 116, 15)
		Q	1 530	14	(74, 46, 14)
H	1 800			23	(75, 47, 14)
		L	570	44	(54, 24, 15)
M	1 064			7	(55, 25, 15)
		Q	1 590	59	(46, 16, 15)
H	1 890			1	(47, 17, 15)
		L	570	12	(151, 121, 15)
M	1 064			7	(152, 122, 15)
		Q	1 590	12	(75, 47, 14)
H	1 890			26	(76, 48, 14)
		L	570	39	(54, 24, 15)
M	1 064			14	(55, 25, 15)
		Q	1 590	22	(45, 15, 15)
H	1 890			41	(46, 16, 15)

注<sup>4</sup>)は、表 12 の注参照。

表 18 36 型~40 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注 <sup>*)</sup> )
36	3 034	L	600	6	(151, 121, 15)
				14	(152, 122, 15)
		M	1 120	6	(75, 47, 14)
				34	(76, 48, 14)
Q	1 680	46	(54, 24, 15)		
		10	(55, 25, 15)		
H	1 980	2	(45, 15, 15)		
		64	(46, 16, 15)		
37	3 196	L	630	17	(152, 122, 15)
				4	(153, 123, 15)
		M	1 204	29	(74, 46, 14)
				14	(75, 47, 14)
Q	1 770	49	(54, 24, 15)		
		10	(55, 25, 15)		
H	2 100	24	(45, 15, 15)		
		46	(46, 16, 15)		
38	3 362	L	660	4	(152, 122, 15)
				18	(153, 123, 15)
		M	1 260	13	(74, 46, 14)
				32	(75, 47, 14)
Q	1 860	48	(54, 24, 15)		
		14	(55, 25, 15)		
H	2 220	42	(45, 15, 15)		
		32	(46, 16, 15)		
39	3 532	L	720	20	(147, 117, 15)
				4	(148, 118, 15)
		M	1 316	40	(75, 47, 14)
				7	(76, 48, 14)
Q	1 950	43	(54, 24, 15)		
		22	(55, 25, 15)		
H	2 310	10	(45, 15, 15)		
		67	(46, 16, 15)		
40	3 706	L	750	19	(148, 118, 15)
				6	(149, 119, 15)
		M	1 372	18	(75, 47, 14)
				31	(76, 48, 14)
Q	2 040	34	(54, 24, 15)		
		34	(55, 25, 15)		
H	2 430	20	(45, 15, 15)		
		61	(46, 16, 15)		

注<sup>(\*)</sup>は、表 12 の注参照。

**8.5.2 誤り訂正コード語の生成** データコード語（必要に応じ、埋め草コード語を含む。）は、表 12～表 18 で示す RS ブロック数に分割する。誤り訂正コード語は、各 RS ブロックに対して計算され、データコード語に付加する。

QR コードの多項式は、2 を法とする算術及び 100011101 を法とする算術（体の原始多項式  $x^8+x^4+x^3+x^2+1$  の係数を示す 100011101 をもつ  $2^8$  のガロア体）を用いて計算する。

データコード語は、多項式の項の係数で、最高次項を最初のデータコード語、最低次項を最終データコード語とする。

誤り訂正コード語は、誤り訂正検出及び訂正（附属書 A 参照）で使用される多項式  $g(x)$  によってデータコード語を除算して得られた剰余とする。この剰余の最上位係数が最初の誤り訂正コード語となり、ゼロ乗の係数が最終誤り訂正コード語で、RS ブロックにおける最終コード語となる。

誤り訂正コード語の生成には、31 の異なる生成多項式を使用する。附属書 A の A.1 にこれらの式を示す。

これを実施するには、図 10 に示すような除算回路を利用する。レジスタ  $b_0 \sim b_{k-1}$  は、ゼロに初期化される。符号化には 2 段階がある。最初の段階では、データコード語は、スイッチが下向きになった状態で出力及び回路を通る。第 1 段階は  $n$  個のクロックパルスで完了する。第 2 段階 ( $n+1, \dots, n+k$  個のクロックパルス) では、スイッチは上向きの状態で、データの入力を 0 にしたまま、レジスタを順番にフラッシュし、出力の補数をとることで、誤り訂正コード語  $\varepsilon_{k-1}, \dots, \varepsilon_0$  を生成する。

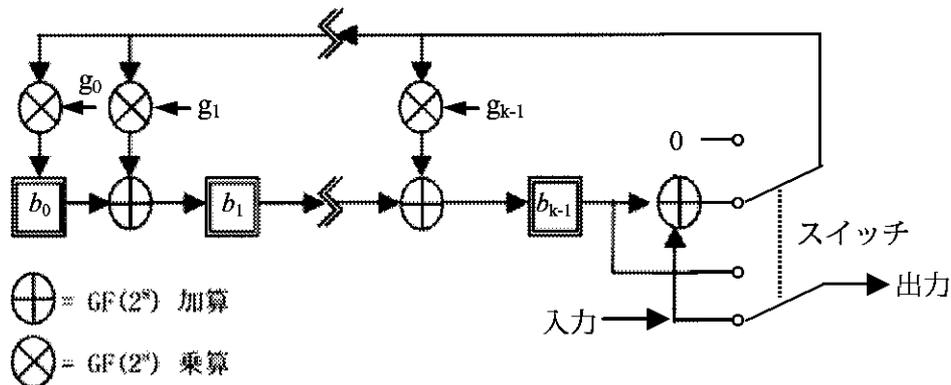


図 10 誤り訂正コード語の符号化回路

**8.6 最終的なメッセージコード語列の構成** メッセージのコード語の総数は、表 7～表 10 及び表 12～表 18 で示すように、シンボル表示が可能なコード語の総数と常に等しい。

コード語の最終列（データ、誤り訂正コード語及び必要であれば、残余コード語を付加する。）を構成するには、次の手順に従う。

- a) 表 12～表 18 で定義される型番及び誤り訂正レベルに基づき、データコード語列を  $n$  個の RS ブロックに分割する。
- b) 各データブロックに対して、8.5.2 及び附属書 A で定義するように対応するデータブロックの誤り訂正コード語を計算する。
- c) 各 RS ブロックから順次、データ及び誤り訂正コード語をとり、最終列を構成する。データブロック 1～データブロック  $n$  の最初のコード語、データブロック 1～データブロック  $n$  の 2 番目のコード語、…データブロック 1～データブロック  $n$  の最終コード語、次に誤り訂正ブロック 1～誤り訂正ブロック  $n$  の最初のコード語、誤り訂正ブロック 1～誤り訂正ブロック  $n$  の 2 番目のコード語、…誤り訂正ブロック 1～誤り訂正ブロック  $n$  の最終コード語を並べて、QR コードシンボルとする。QR コードシンボ

ルは、シンボルのコード語サイズを常に満たすデータ及び誤り訂正ブロックで構成する。しかし、ある型番においては、符号化領域のモジュール数を完全に満たすために最終メッセージの後に付加する3個、4個又は7個の残余ビットを必要とする。

一番短いデータブロックをデータ列の最初に配置し、すべてのデータコード語は、誤り訂正コード語の前に配置する。例えば、5-H型シンボルは4個のデータ及び誤り訂正ブロックで構成するが、最初の2ブロックはそれぞれ、11個のデータ及び22個の誤り訂正コード語、3番目及び4番目のブロックはそれぞれ、12個のデータ及び22個の誤り訂正コード語を含むRSブロックとする。このシンボルにおけるキャラクタ配列は、表19のようになる。表の各行は、誤り訂正コード語（Enで示す。）のブロックとともにデータコード語（Dnで示す。）の1ブロックに対応する。シンボルにおけるキャラクタ配置の列は、表の各列を順次、下にとっていくことで得られる

表 19 5-H型シンボルのRSブロック構成

RS ブロック	データコード語				誤り訂正コード語			
	D1	D2	...	D11	E1	E2	...	E22
RS ブロック 1	D1	D2	...	D11	E1	E2	...	E22
RS ブロック 2	D12	D13	...	D22	E23	E24	...	E44
RS ブロック 3	D23	D24	...	D33	D34	E45	E46	...
RS ブロック 4	D35	D36	...	D45	D46	E67	E68	...

したがって、5-H型シンボルにおける最終的なメッセージコード語列は、次のようになる。すなわち、D1, D12, D23, D35, D2, D13, D24, D36, ..., D11, D22, D33, D45, D34, D46, E1, E23, E45, E67, E2, E24, E46, E68, ..., E22, E44, E66, E88 となる。必要に応じて、最終コード語の後に残余ビット“0”を付加することでシンボルのモジュール容量を満たす。

## 8.7 コード語の配置

**8.7.1 シンボルキャラクタの表示** QRコードシンボルには規則的及び不規則的な2種類のシンボルキャラクタを使う。これらの使用は、シンボルの位置に依存し、他のシンボルキャラクタ及び機能パターンに関係する。

シンボルにおいて多くのコード語は、規則的な2×4モジュールブロックとして表示する。これらブロックの配置は、2種類とし、垂直配列（幅が2モジュール、高さが4モジュール）と、必要に応じて配列方向を変更する場合の水平配列（幅が4モジュール、高さが2モジュール）となる。不規則的なシンボルキャラクタは、配列方向が変わる場合、位置合せパターン又は他の機能パターン付近で使用する。

**8.7.2 機能パターンの配置** 使用する型番に従って、水平及び垂直に配置するモジュールで正方形の空マトリックスを構成する。位置検出パターン、分離パターン、タイミングパターン及び位置合せパターンに対応する位置は、特定の暗モジュール又は明モジュールで占める。形式情報及び型番情報のモジュール位置は、一時的に空とする。これらの位置は、すべての型番に共通し、図14及び図15に示す。位置合せパターンの位置については、附属書Eに示す。

**8.7.3 シンボルキャラクタの配置** QRコードシンボルの符号化領域において、シンボルキャラクタは、2モジュール幅の縦列で、シンボルの右下隅から開始し、上方向及び下方向を交互に走査し、右から左へ配置する。主なキャラクタ及びキャラクタ内のビット配置方法については、次による。これらの規則を適用した2型及び7型のシンボルを図14及び図15に示す。

a) 列におけるビット配列方向は、右から左とし、シンボルキャラクタの配置方向によって上向き又は下向きのいずれかとする（図11参照）。

- b) 各コード語の最上位ビット（ビット 7 で示す。）は、最初に利用できるモジュール位置に配置する。後続するビットは、次のモジュール位置に配置する（図 11 参照）。

したがって、最上位ビットは、配置方向が上向きの場合は規則的なシンボルキャラクタの右下モジュールに、配置方向が下向きの場合は右上モジュールに置く。しかし、それは、直前のキャラクタが右側のモジュール列で終了する場合などは、不規則的なシンボルキャラクタの左下に置くこともある（図 13 参照）。

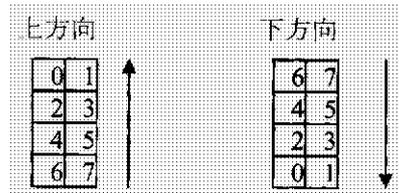


図 11 上方向及び下方向の規則的な配置

- c) シンボルキャラクタが両モジュール列で位置合せパターン又はタイミングパターンの水平境界に達する場合、符号化領域が連続しているとし、それらのパターンの上又は下に連続する（図 12 参照）。
- d) シンボルキャラクタ領域の上側又は下側の境界に到達した場合（すなわち、シンボルの端、形式情報、型番情報又は分離パターン）、コード語の残りのビットは左側の次の列に配置し、配置方向が反転する（図 12 参照）。

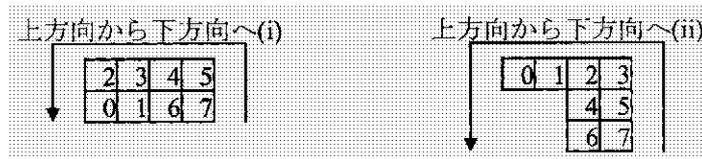


図 12 配置方向転換の例

- e) シンボルキャラクタ列の右側のモジュール列が位置合せパターン又は型番情報で占められている領域に到達した場合、ビットは、位置合せパターン又は型番情報に隣接する 1 列のモジュールに沿って延長する不規則なシンボルキャラクタを形成するように配置する。また、2 列が次のシンボルキャラクタに使用される前にキャラクタが終了する場合、次のキャラクタの最上位ビットは 1 列で配置する（図 13 参照）。

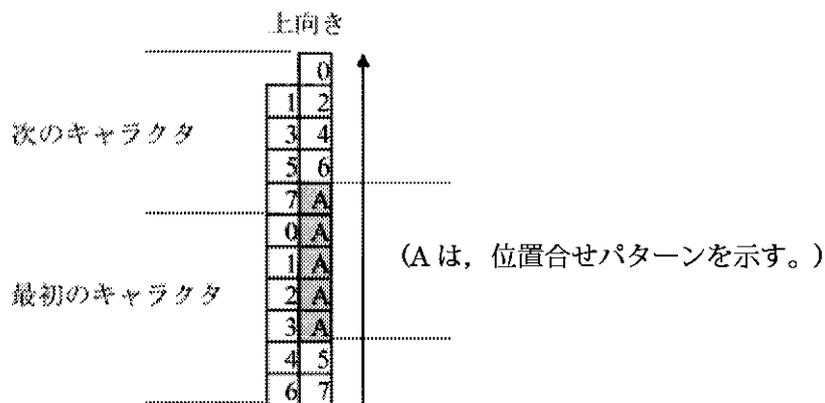


図 13 位置合せパターン付近の配置例

シンボルにおける配置の別法（結果的に同じ。）は、単一ビット列として 2 モジュール幅の左右交互配置

コード語列とみなす。それは、シンボルの右から左へ、上方向及び下方向を交互に 2 モジュール幅の縦列で配置する。各列において、ビットは、右及び左のモジュールを交互に、配置方向によって上向き又は下向きに、機能モジュールで占められている領域を飛ばし、列の一番上又は一番下で方向を変えながら配置する。各ビットは、常に最初に利用できるモジュールから配置する。

シンボルのデータ容量が 8 ビットのシンボルキャラクタで割り切れない場合、シンボル容量を満たすために適切な個数の残余ビットを使用する。これらの残余ビットは、8.8 によるマスク処理の前では、常に値 0 とする。

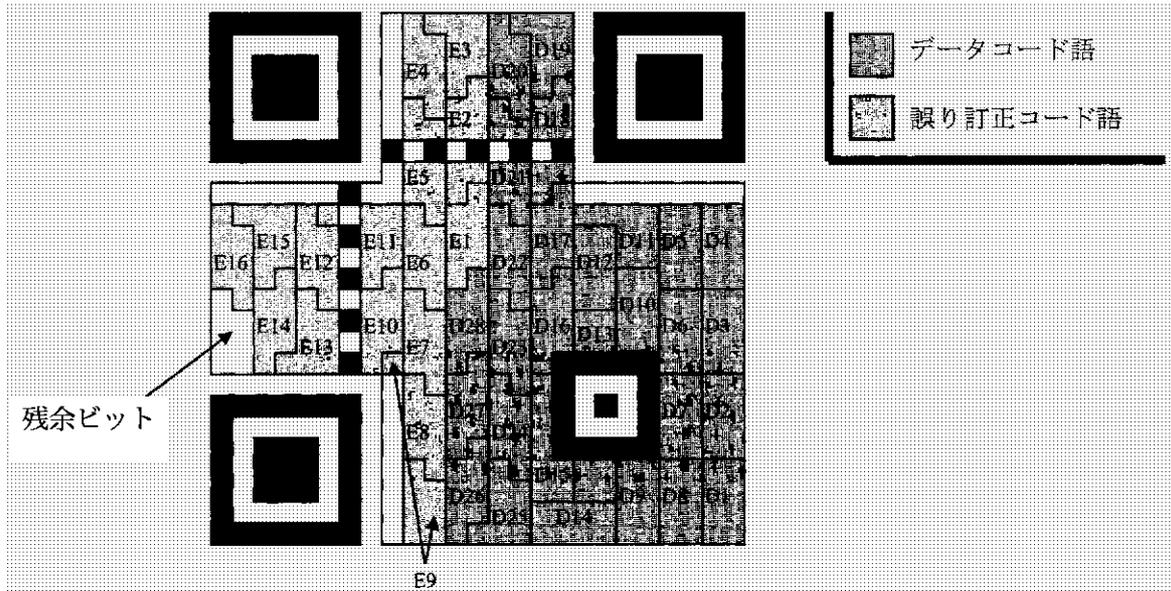
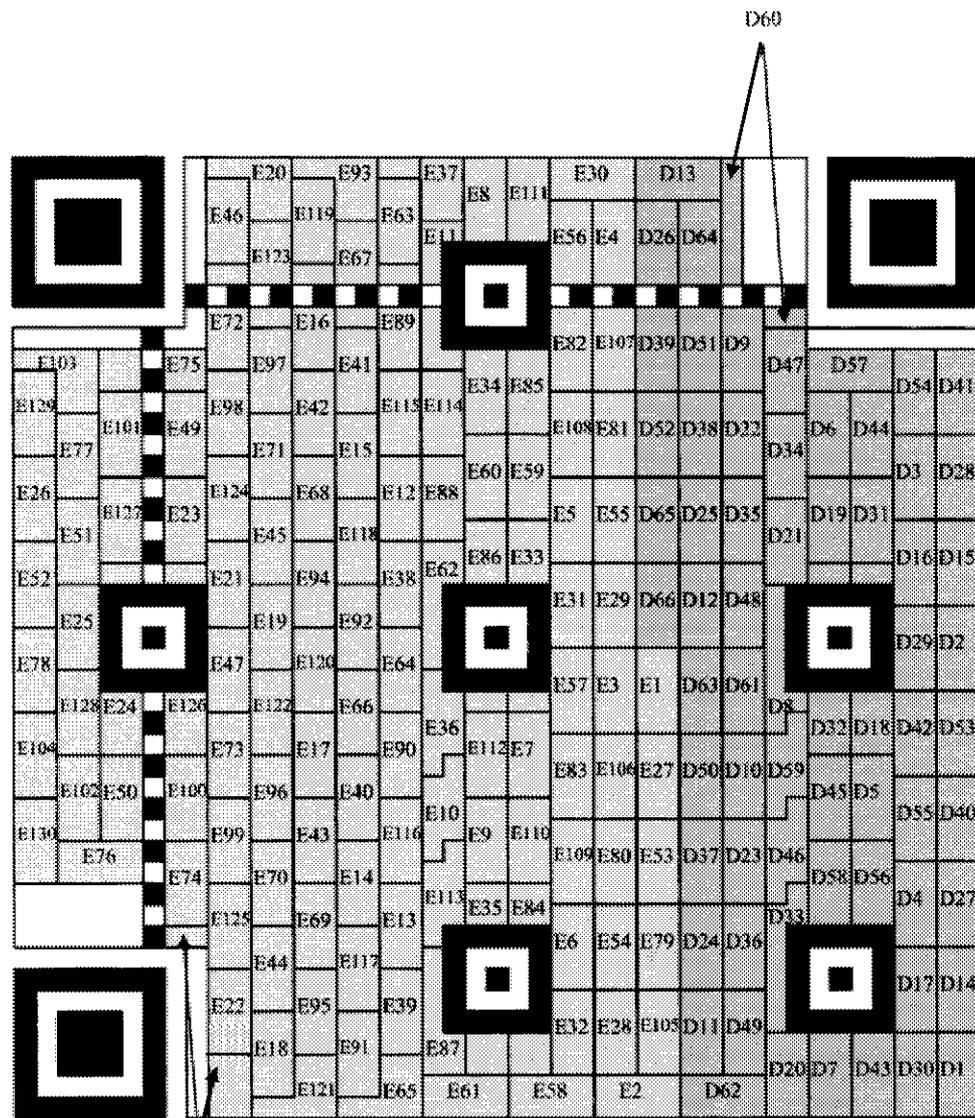


図 14 2-M 型におけるシンボルキャラクタ配列



E48

D1~D13 :	データブロック1
D14~D26 :	データブロック2
D27~D39 :	データブロック3
D40~D52 :	データブロック4
D53~D66 :	データブロック5
E1~E26 :	誤り訂正ブロック1
E27~E52 :	誤り訂正ブロック2
E53~E78 :	誤り訂正ブロック3
E79~E104 :	誤り訂正ブロック4
E105~E130 :	誤り訂正ブロック5

図 15 7-H型におけるシンボルキャラクタの配列

**8.8 マスク処理** QR コードの読出しを確実にするためには、明及び暗のモジュールをバランスよくシンボル内に配列することが望ましい。位置検出要素パターンに特徴的に見られる 1011101 のビットパターンがシンボル内にできるだけ存在しないようにする。上の条件を満たすため、次の手順に従ってマスク処理を適用する。

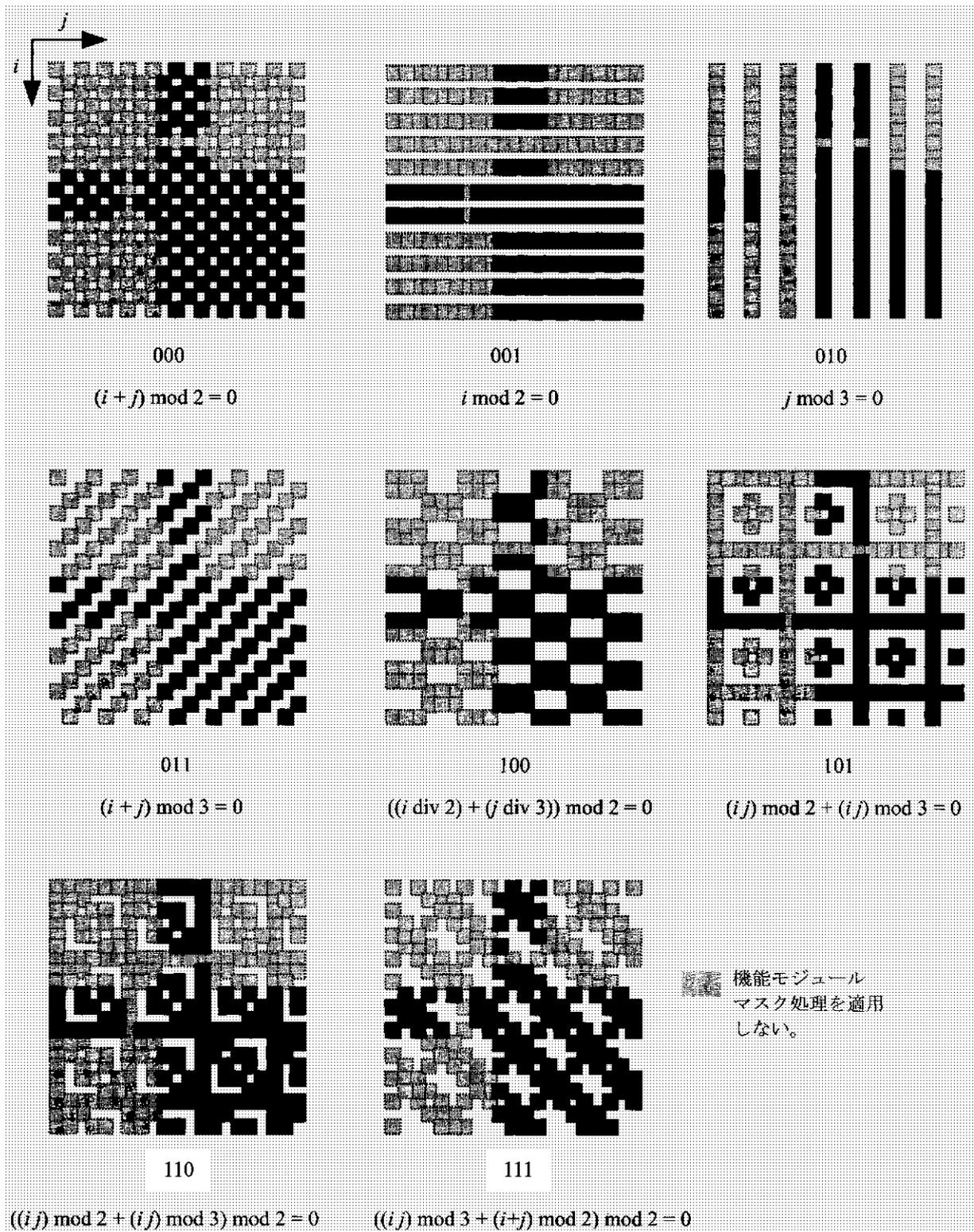
- a) マスク処理は、機能パターンには適用しない。
- b) 符号化領域（形式情報及び型番情報を除く。）で与えられたモジュールパターンと複数のマスク処理マトリックスパターンとで順に XOR 演算による変換をする。XOR 演算とは、各マスク処理マトリックスパターン上にモジュールパターンを重ね合わせ、マスク処理パターンの暗モジュールに対応するモジュールを反転（明から暗へ、暗から明へ）することとする。
- c) それぞれの変換結果において、望ましくない項目に失点を課すことですべての変換結果を評価する。
- d) 失点の合計が一番低いパターンを選択する。

**8.8.1 マスクパターン** 表 20 は、マスクパターン参照子（形式情報で使用される 2 進値）及びマスクパターン生成条件を示す。マスクパターンは、符号化領域（形式情報及び型番情報の領域を除く。）において条件が真であるモジュールを暗と定義することで、マスクパターンを生成する。条件において、 $i$  はモジュールの行位置を示し、 $j$  は列位置を示す。 $(i, j) = (0, 0)$  は、シンボルの左上モジュールを示す。

表 20

マスクパターン参照子	条件
000	$(i+j) \bmod 2=0$
001	$i \bmod 2=0$
010	$j \bmod 3=0$
011	$(i+j) \bmod 3=0$
100	$((i \operatorname{div} 2)+(j \operatorname{div} 3)) \bmod 2=0$
101	$(i \bmod 2)+(j \bmod 3)=0$
110	$((i \bmod 2)+(j \bmod 3) \bmod 2)=0$
111	$((i \bmod 3)+(j \bmod 2) \bmod 2)=0$

図 16 は、1 型シンボルに対するすべてのマスクパターンを示す。図 17 は、マスクパターン参照子 000～111 によるマスク処理効果のシミュレーションを示す。



備考1. 各パターン下の3ビットがマスクパターン情報となる。

2. 3ビットの下の式は、マスクパターン生成条件を示し、式を満たすモジュールが暗モジュールに対応する。

3. ここに示すマスクパターンの大きさは、1型シンボルのマスクパターンとする。

図 16 1型シンボルに対する全マスクパターン

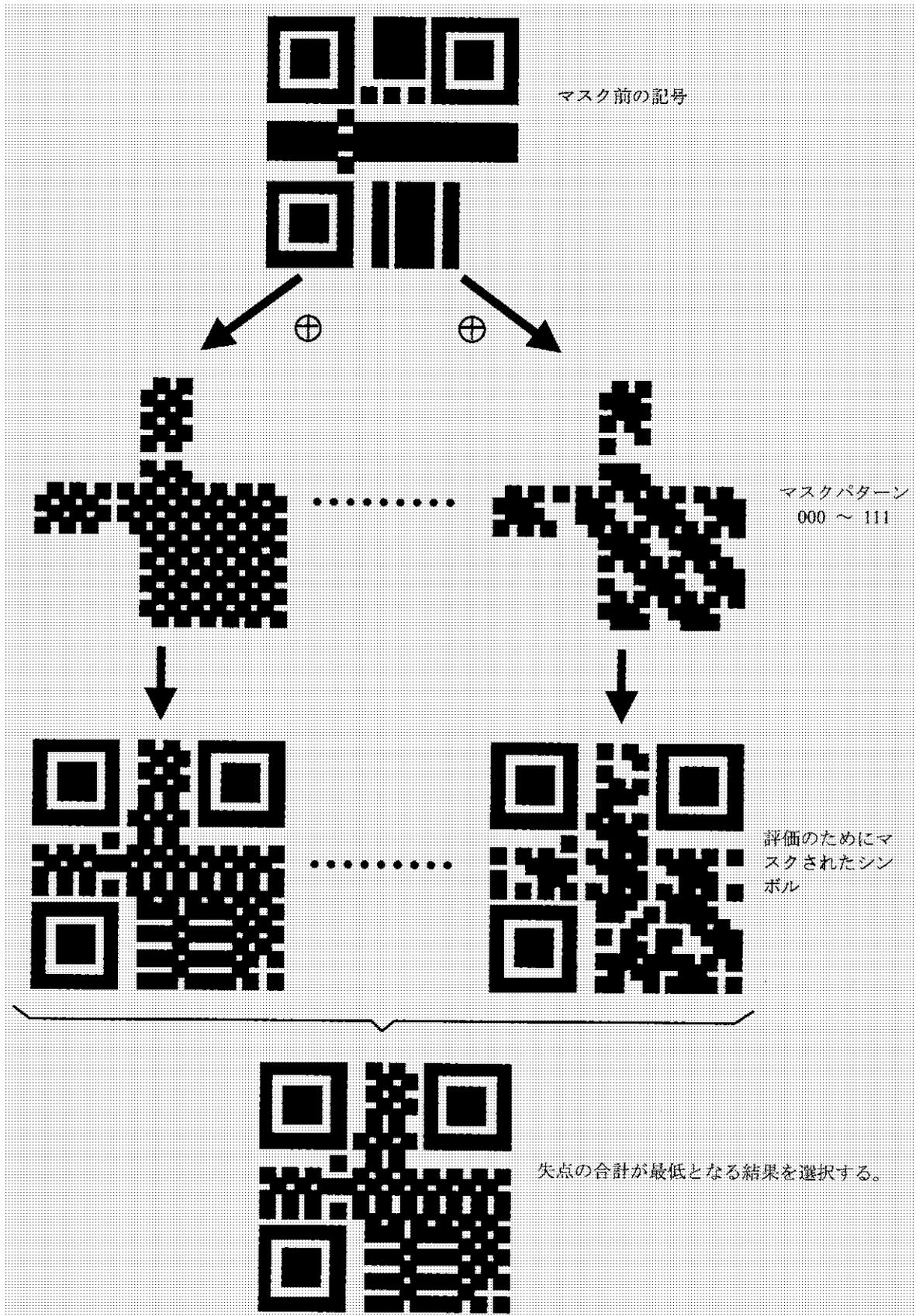


図 17 マスク処理のシミュレーション

**8.8.2 マスク処理結果の評価** 各マスクパターンで順次マスク処理を実行した後、次の特徴の発生に失点を課すことで結果を評価する。失点が高いほど、結果が受け入れにくい。表 21 で、変数  $N_1 \sim N_4$  は、望ましくない特徴 ( $N_1=3, N_2=3, N_3=40, N_4=10$ ) に対する重み付けされた失点を示す。 $i$  は同色の隣接モジュール数が 5 個を超える分で、 $k$  はシンボル内の暗モジュール比率で、50 %から 5 %ごとの偏差の度合いとする。マスク処理は、シンボルの符号化領域〔形式情報及び型番情報（存在する場合）を除く。〕について行われるが、評価される領域はシンボル全体とする（図 2 参照）。

表 21 マスク処理結果の失点

特徴	評価条件	失点
同色の行・列の隣接モジュール	モジュール数 = $(5+i)$	$N_1+i$
同色のモジュールブロック	ブロックサイズ = $2 \times 2$	$N_2$
行・列における 1:1:3:1:1 比率の (暗:明:暗:明:暗) のパターン	1:1:3:1:1 比率のパターンの前又は後ろに比率 4 の幅以上の明パターンが存在する。	$N_3$
全体に対する暗モジュールの占める割合	$50 \pm (5 \times k) \% \sim 50 \pm (5 \times (k+1)) \%$	$N_4 \times k$

失点の合計が最小となるマスクパターンをシンボルに適用する。

**参考...**

一 同色の行又は列の隣接モジュール

横及び縦の両方向について、明（白）モジュール又は暗（黒）モジュールが 5 個以上連続するブロックをカウントする。

なお、失点は、5 個連続のブロック 1 個につき 3 点、6 個連続のブロック 1 個につき 4 点とし、ブロック数が増加する度に 1 点ずつ増加する。例えば“暗暗暗暗暗暗暗”のブロックがあった場合には、7 個連続するモジュールが 1 ブロックとみなし、失点は 5 点とする（暗の 5 個連続が 1 個、暗の 6 個連続が 1 個、暗の 7 個連続が 1 個と重複してカウントしない）。

一 同色のモジュールブロック

暗又は明の  $2 \times 2$  のブロックの個数をカウントし、その個数を失点とする。例えば暗の  $3 \times 3$  ブロックには、暗の  $2 \times 2$  ブロックが 4 個あるので、4 個  $\times$  3 点 = 12 点を失点とする。

一 行/列における 1:1:3:1:1 比率のパターン

1:1:3:1:1 の比率の暗:明:暗:明:暗のパターンに続いて比率 4 の幅以上の明パターンが存在する場合、失点は 40 点とする。

一 全体に占める暗モジュールの割合

50 % (0 点) から 5 % 増減する度に 10 点を加算する。例えば暗のモジュールの比率が 45 % ~ 55 % の場合、失点は 10 点、40 % ~ 60 % の場合、失点は 20 点とする。

**8.9 形式情報** 形式情報は (15, 5) BCH コードで計算する 10 誤り訂正ビット及び 5 データビットをもつ 15 ビット列とする。形式情報の誤り訂正計算に関する詳細は、附属書 C による。表 22 に示すように、最初の 2 データビットは、シンボルの誤り訂正レベルとする。

表 22 誤り訂正レベル指示子

誤り訂正レベル	2 進指示子
L	01
M	00
Q	11
H	10

形式情報の3番目～5番目のデータビットは、8.8.2に基づき選択されるパターンに対するマスクパターン参照子（表20参照）とする。

10誤り訂正ビットは、附属書Cのとおり計算され、5データビットに付加する。

15ビットの誤り訂正化された形式情報は、誤り訂正レベル及びマスクパターンの組合せによってデータ文字列がすべてゼロにならないよう、特定のビットパターンとでXOR演算する。そのXOR演算用のマスクパターンは、101010000010010とする。

得られたマスク形式情報は、図18に示すようにシンボルの予約領域に割り当てる。形式情報の正確な復号は、完全なシンボルの復号に必ず（須）であるので、その冗長性を与えるため、形式情報はシンボルに2か所存在する。形式情報の最下位ビットは、図18のモジュール数0に位置し、最上位ビットは、モジュール数14に位置する。位置 $(4V+9, 8)$ のモジュールは、常に暗で形式情報の一部ではない（ここに、 $V$ は型番番号）。

#### 例

誤り訂正レベルM:	00
マスクパターン参照子:	101
データ:	00101
BCHビット:	0011011100
マスク前のビット列:	001010011011100
XOR演算用マスクパターン:	010100000100101
形式情報モジュールパターン:	100000011001110

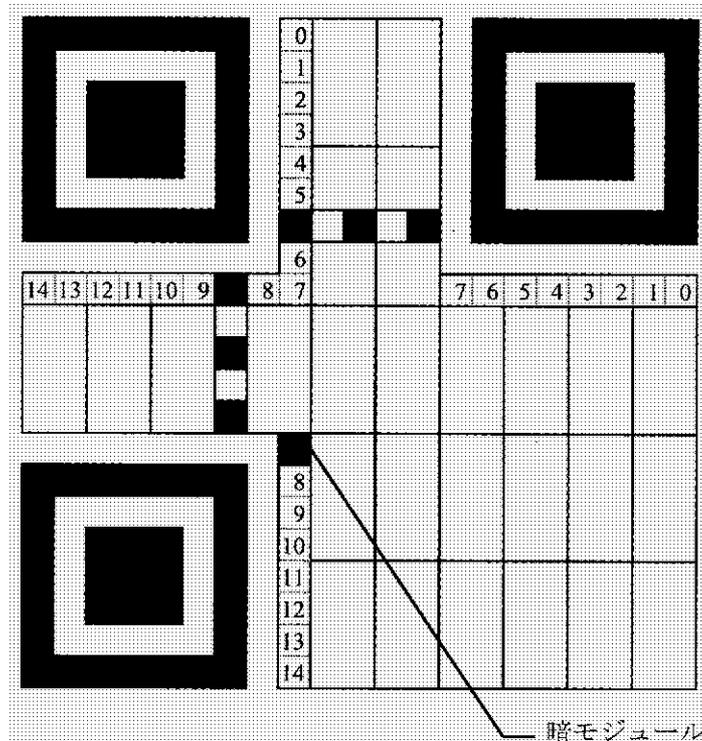


図 18 形式情報の配置

**8.10 型番情報** 型番情報は、(18, 6) 拡張 BCH コードで計算される 12 誤り訂正ビット及び 6 データビットをもつ 18 ビット列とする。型番情報の誤り訂正計算に関する詳細は、**附属書 D** による。6 データビットは、型番情報であり、最上位ビットを先頭とする。

12 誤り訂正ビットは、**附属書 D** のとおり計算され、6 データビットに付加する。

7 型～40 型のシンボルが型番情報を保つために、すべてが 0 のデータ列となる型番情報は存在しない。したがって、型番情報にマスク処理は適用されない。

得られた型番情報は、**図 19** に示すようにシンボルの予約領域に割り当てる。型番情報の正確な復号は、完全なシンボルの復号に必ず（須）であるので、その冗長性を与えるため、型番情報はシンボルに 2 か所存在する。型番情報の最下位ビットは、**図 20** のモジュール数 0 に位置し、最上位ビットはモジュール数 17 に位置する。

#### 例

型番番号：	7
データ：	000111
BCH ビット：	110010010100
型番情報モジュール：	000111110010010100

型番情報の領域は、右上の位置検出要素の左側の 6×3 モジュールブロック、及び左下の位置検出要素の上側の 3×6 モジュールブロックとする。

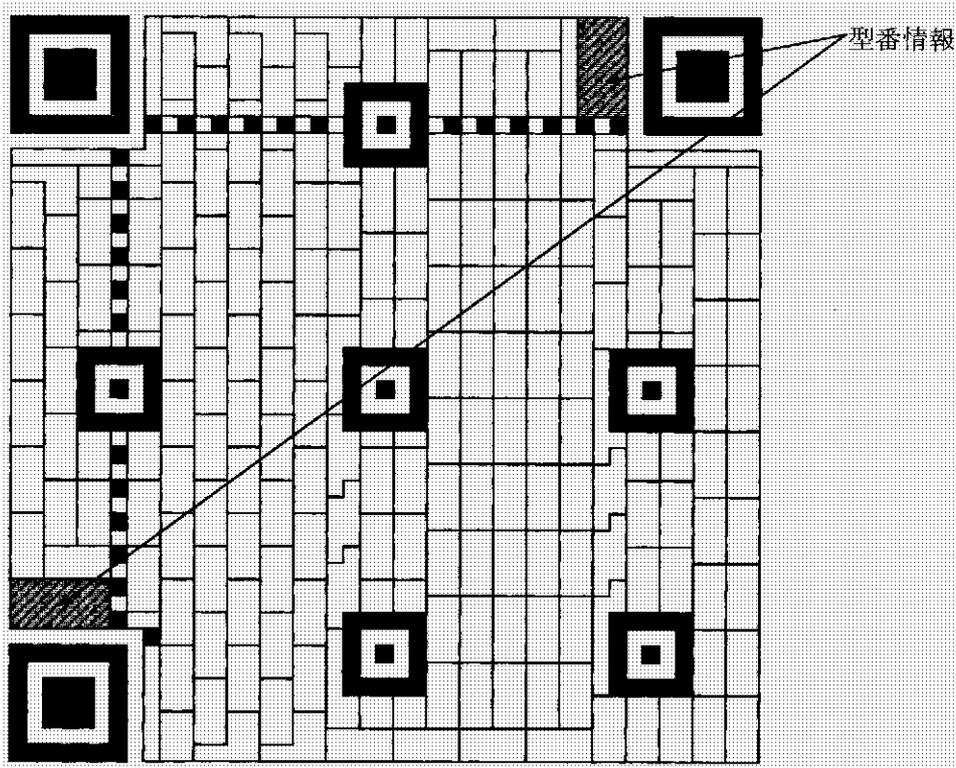


図 19 型番情報の配置

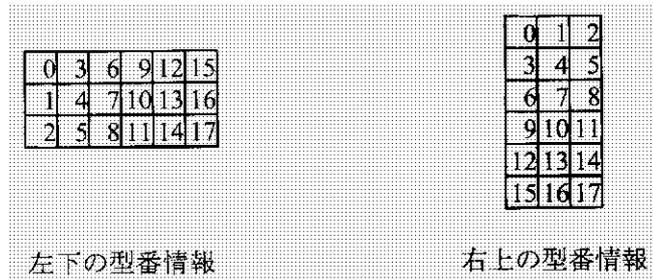


図 20 型番情報のモジュール配列

## 9. 構造的接続

**9.1 基本原理** 最大 16 個までの QR コードシンボルを構造的接続の形で連結することができる。シンボルが構造的接続メッセージの一部の場合には、最初の 3 個のシンボルキャラクタ位置にあるヘッダブロックに構造的に接続されていることを示す。

構造的接続モード指示子 0011 を最初のシンボルキャラクタの上位 4 ビットに置く。

次に、2 個の構造的接続コード語が後続する。最初のシンボルキャラクタの下位 4 ビット、2 番目のシンボルキャラクタ及び 3 番目のシンボルキャラクタの上位 4 ビットにわたる。最初のコード語は、シンボル列指示子とする。2 番目のコード語は、パリティデータで、メッセージの全シンボルで等しく、すべてのシンボルが同じ構造的接続メッセージから読み出されたことが確認できる。このヘッダのすぐ後に、最初のモード指示子で始まるシンボルのデータコード語が続く。既定の ECI 以外の一つ以上の ECI を適用する場合、各 ECI に対する ECI ヘッダ (ECI モード指示子及び ECI 指定で構成する。) は、構造的接続ヘッダの後に続く。

図 21 は、構造的接続 QR コードの例を示す。

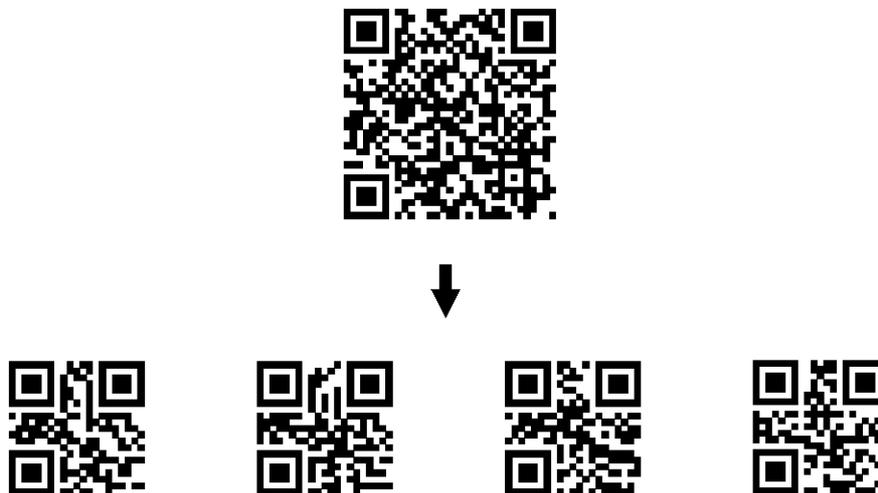


図 21 構造的接続

**9.2 シンボル列指示子** このコード語は、接続構造における QR コードシンボルのセット (最大 16 個まで) の中でのシンボル位置を示す ( $n$  シンボル中の  $m$  番目の形式で)。このコード語の最初の 4 ビットは、そのシンボルの位置を示す。最後の 4 ビットは、構造的接続の形で連結されるシンボルの合計数を示す。4 ビットパターンは、それぞれ  $(m-1)$  及び  $(n-1)$  の 2 進値とする。

**例**

合計 7 シンボル中の第 3 シンボルを示す場合は、次のように符号化する

第 3 位置	0010
全 7 シンボル	0110
ビットパターン	00100110

**9.3 パリティデータ** パリティデータは、シンボル列指示子に続く 8 ビットバイトで、シンボルブロックへの分割前の全入力データをバイトごとに XOR 演算で得られる値とする。モード指示子、文字数指示子、埋め草ビット、終端パターン及び埋め草キャラクタは、この計算から除く。入力データは、漢字の場合には 2 バイトのシフト符号化表現で規定された文字（各バイトは XOR 計算で別個に処理される。）、他の文字は、表 6 に示す 8 ビット符号で表示する。ECI モードでは、データの暗号化又は圧縮後に得られる 8 ビットバイト値を計算に使用する。

例えば、“0123456789 日本”を次のように“0123”、“4567”及び“89 日本”に分割した場合、

第 1 シンボルブロック (“0123”) – 16 進値 30, 31, 32, 33

第 2 シンボルブロック (“4567”) – 16 進値 34, 35, 36, 37

第 3 シンボルブロック (“89 日本”) – 16 進値 38, 39, 93FA, 967B

バイトごとに、順次、データを XOR 演算することで“0123456789 日本”のパリティデータを得る。

$$30 \oplus 31 \oplus 32 \oplus 33 \oplus 34 \oplus 35 \oplus 36 \oplus 37 \oplus 38 \oplus 39 \oplus 93 \oplus \text{FA} \oplus 96 \oplus 7\text{B} = 85$$

**備考** パリティデータの計算は、印字装置の能力に応じて、データを印字装置へ転送する前又は印字装置内で実行する。

**10. シンボルの印刷及びマーキング**

**10.1 寸法** 次の寸法を QR コードシンボルに適用する。

*X* 寸法 モジュール幅は、使用するスキャン技術及びシンボル生成技術を考慮した上でアプリケーションによって指定される。

*Y* 寸法 モジュールの高さは *X* 寸法と同じとする。

最小クワイエットゾーン 四辺すべてにおいて  $4X$  と同値とする。

**10.2 文字による表記** QR コードシンボルは、何千もの文字を符号化することができるので、データ文字を人が読取り可能な文字で表記することはあまり実際的ではない。その代わりとして、テキストそのものを示すのではなく、そのシンボルの内容を記述するものをシンボルに併記する。印刷する場合は、文字の大きさ及びフォントについて特に指定はなく、メッセージは、シンボルの周辺のどこか適切なところに印刷すればよい。人が読取り可能な記述は、シンボル及びクワイエットゾーンを妨害してはならない。

**10.3 マーキングの手引き** 幾つかの異なる技術で QR コードシンボルは、印刷及びマーキングされる。附属書 I に利用者のための手引きを示す。

**11. シンボルの品質** 附属書 K に示す二次元マトリックスコードシンボルの印刷品質手引きを適用し、11.2～11.4 に従って QR コードシンボルの品質を評価する。

**11.1 試験画像の取得** 附属書 K の K.1 に示すように精密ビデオカメラを基本とした装置で、試験するシンボルのグレースケールの画像を得る。ただし、照明の色及び方向は、アプリケーションによって規定する。

## 11.2 シンボル品質のパラメタ

**11.2.1 復号** 13.で設定する参照復号アルゴリズムを試験画像に適用する。データメッセージ全体の復号が成功した場合、等級“A”(4.0)で復号は合格とし、それ以外は等級“F”(0.0)で不合格とする。

**11.2.2 シンボルコントラスト** 参照復号で決定される4モジュール幅のクワイエットゾーンを含むシンボル境界内にある試験画像のすべてのグレースケールピクセルの値を用いて、シンボルのコントラストの等級を決める。この手順を**附属書 K**の**K.2.2**に示す。

**11.2.3 モジュールの伸縮** 試験シンボルの高解像度の2進デジタル画像を作成することで、参照復号を開始し、シンボルのタイミングパターンを二分する“明暗モジュールの中心線”の位置を設定する。明暗パターンを通して、線の“デューティサイクル”が約50%かどうかを検査することでモジュールの伸縮を評価する。

二つのタイミングパターンをそれぞれとり(水平方向及び垂直方向の伸縮は異なるので)、各タイミングパターンに隣接する位置検出要素パターンの外側の正方形の内側から各タイミングパターンに沿って、もう一つの位置検出要素パターンに隣接する分離パターンの外縁まで進み、出現する明( $N_L$ )及び暗( $N_D$ )のピクセル数を合計する。各方向でのモジュールの伸縮の測定結果は、 $D=N_D/(N_L+N_D)$ となる。これは、**附属書 K**の**K.2.3**に示す $D_{NOM}=0.50$ 、 $D_{MIN}=0.35$ 及び $D_{MAX}=0.65$ に対する等級とする。垂直及び水平のタイミングパターンに沿って得られるモジュールの伸縮の等級は、低くなる。

**11.2.4 軸の非均一性** 参照復号アルゴリズムによって、試験画像の全領域を通してデータモジュール標本化点のグリッドを生成する。これらの標本化点の水平及び垂直方向の正確な間隔を、軸の非均一性の評価の基準とする。

隣接するデータモジュール間の水平及び垂直方向の間隔をそれぞれ用いて、シンボル全体に関してそれらの平均値 $X_{AVG}$ 及び $Y_{AVG}$ を計算する。**附属書 K**の**K.2.4**で定義する手順で計算して、軸の非均一性等級は、二つの平均間隔値どしどしどれだけ近い値かを基準にする。

**11.2.5 未使用誤り訂正** QRコードは、リードソロモン誤り符号を使用する。小さいシンボルは一つのリードソロモンブロックを含むのに対し、大きなシンボルは二つ以上のリードソロモンブロックに分割される。あらゆる場合において、各リードソロモンブロックは、**附属書 K**の**K.2.5**に示すように別々に等級を決める。未使用誤り訂正等級は、これらの各ブロックの等級値の中で一番低い値とする。

なお、この計算は、形式情報及び型番情報には適用しない。

**11.3 シンボル等級の総合評価** QRコードシンボルの印刷品質等級の総合評価は、上で得られる5項目の等級値の中で一番低い値とする。**表 23**に等級規準の一覧を示す。

表 23 シンボル等級規準

等級	参照復号	シンボル コントラスト	モジュールの伸縮	軸の非均一性	未使用 誤り訂正
A (4.0)	合格	$SC \geq 0.70$	$-0.50 \leq D' \leq 0.50$	$AN \leq 0.06$	$UEC \geq 0.62$
B (3.0)		$SC \geq 0.55$	$-0.70 \leq D' \leq 0.70$	$AN \leq 0.08$	$UEC \geq 0.50$
C (2.0)		$SC \geq 0.40$	$-0.85 \leq D' \leq 0.85$	$AN \leq 0.10$	$UEC \geq 0.37$
D (1.0)		$SC \geq 0.20$	$-1.00 \leq D' \leq 1.00$	$AN \leq 0.12$	$UEC \geq 0.25$
F (0.0)	不合格	$SC < 0.20$	$D' < -1.00$ or $D' > 1.00$	$AN > 0.12$	$UEC < 0.25$

**11.4 プロセス制御の測定** QRコードシンボルの生成プロセスを監視及び制御する種々の有用な測定ツール及び測定方法が存在する。これらのツール及び方法は、次を含む。

a) バーコード検証機を使用したシンボルコントラストの読取り

- b) バーコード検証機を使用した位置検出要素パターンの両軸の測定による水平（垂直）モジュールの伸縮
  - c) 物理的な測定による軸の非均一性の決定
  - d) グリッドの非均一性及び欠陥に対する位置検出及びタイミングパターンの目視検査
- a)～d) のツール及び方法については、それぞれ**附属書 L**に示す。

**12. 復号手順の概要** QR コードシンボルの読取りからデータ文字出力に至る復号手順は、符号化手順の逆手順とする。**図 22**にこの過程の概要を示す。

- a) シンボルの位置を検出し、画像を取得する。“0”及び“1”のビット列として明及び暗のモジュールを認識する。
- b) 形式情報を読み出す（マスク処理を解除し、必要に応じて誤り訂正を実行し、誤り訂正レベル及びマスクパターン参照を認識する。）。
- c) シンボルの型番を決定する。
- d) 形式情報で得られたマスクパターンを使って符号化領域ビットパターンを XOR 演算し、マスク処理を解除する。
- e) 配置規則に従い、シンボルキャラクタを読み取り、メッセージのデータ及び誤り訂正コード語を復元する。
- f) 誤りがあるかどうかの検出をする。誤りが検出された場合、これを訂正する。
- g) モード指示子及び文字数指示子に基づいて、データコード語をセグメントに分割する。
- h) 最後に、使用モードに基づいてデータ文字を復号し、結果を出力する。

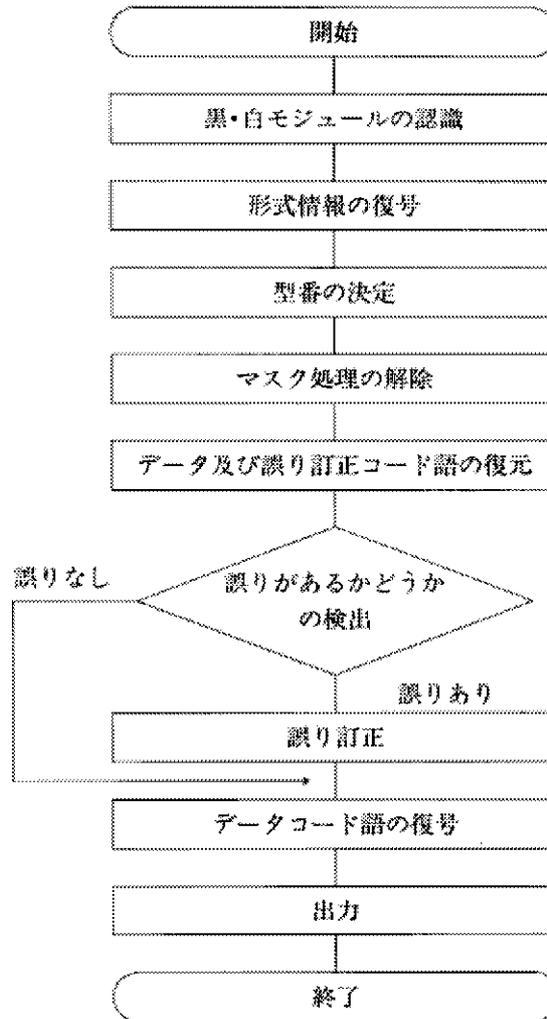


図 22 QR コードの復号手順

**13. QR コードの参照復号アルゴリズム** ここでは、参照復号アルゴリズムを規定する。この参照復号アルゴリズムは、画像内のシンボルを検出し、復号する。この復号アルゴリズムは、画像内の明暗状態を参照する。

- a) 画像内における最大反射値及び最小反射値の間の中間反射値をしきい値とする。これを使って、画像を暗及び明のピクセル集合に変換する。
- b) 位置検出パターンの位置を求める。QR コードの位置検出パターンは、シンボルの 4 隅の内の 3 隅に配置される同一の位置検出要素パターンで構成する。7.3.2 で示すように、各位置検出要素パターンのモジュール幅は、1:1:3:1:1 の比率をもつ暗-明-暗-明-暗の列で構成する。このアルゴリズムにおいて、これらの幅の許容値は、0.5 とする（つまり、1 個のモジュールに対しては 0.5~1.5 の範囲、3 個のモジュールに対しては 2.5~3.5 の範囲となる。）。
  - 1) 領域の候補が検出されたときに、画像内のピクセル線が位置検出要素パターンの外縁に接する最初の点 A 及び最後の点 B の位置を記憶する（図 23 参照）。画像内の x 軸方向に対して位置検出要素パターンの内側の暗の正方形を横切るすべての線が認識されるまで、画像内の隣接するピクセル線についてこれを繰り返す。

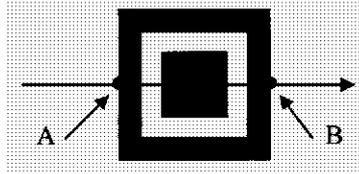


図 23 位置検出要素パターンの走査線

- 2) 画像内の y 軸方向に対して、位置検出要素パターンの内側の暗の正方形を横切るピクセル列について手順 1) を繰り返す。
- 3) パターンの中心位置を求める。x 軸方向に対して位置検出要素パターンの内側の暗の正方形を横切る一番外側のピクセル線上の点 A 及び点 B の中央を通る線を求める。同様に y 軸方向に対して位置検出要素パターンの内側の暗の正方形を横切る一番外側のピクセル列上の点 A 及び点 B の中央を通る線を求める。パターンの中心は、この 2 本の線の交点として求める。
- 4) 1)～3) の手順を繰り返し、他の 2 個の位置検出要素パターンの中心位置を求める。
- e) シンボルにおいて、左上の位置検出要素パターン及びシンボルの回転角を認識する位置検出要素パターンの中心座標を分析することによって、シンボルの向きを求める。
- d) シンボルの最大幅を横切る、左上の位置検出要素パターン及び右上の位置検出要素パターンの中心間の距離  $D$  及び二つのパターンの幅 ( $W_{UL}$  及び  $W_{UR}$ ) を求める (図 24 参照)。

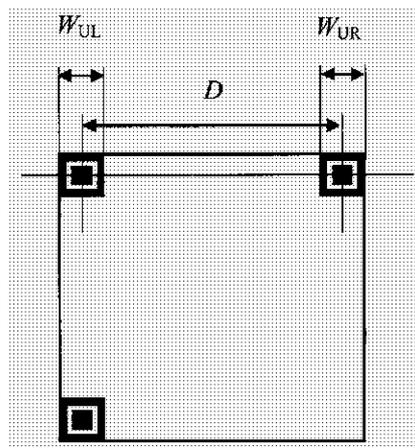


図 24 上側の位置検出要素パターン

- e) シンボルの公称  $X$  寸法を計算する。
 
$$X = (W_{UL} + W_{UR}) / 14$$
- f) シンボルの型番  $V$  を仮に求める。
 
$$V = [(D/X) - 10] / 4$$
- g) 仮のシンボル型番が 6 以下の場合、この値を型番として使用する。仮のシンボル型番が 7 以上の場合は、次のように型番情報を復号する。
  - 1) 右上の位置検出要素パターンの幅  $W_{UR}$  を 7 で除し、モジュールサイズ  $CP_{UR}$  を求める。
 
$$CP_{UR} = W_{UR} / 7$$
  - 2) 図 25 に示す 3 個の位置検出要素パターンの中心位置 A, B 及び C から補助線 AC 及び AB を求める。この補助線の平行線、位置検出要素パターンの中心座標及びモジュールサイズ  $CP_{UR}$  に基づき、型番情報 1 領域の各モジュールの中心に対するサンプリンググリッドを設定する。サンプリンググリ

ッド上の明又は暗の状態から 2 進値 0 及び 1 を決定する。

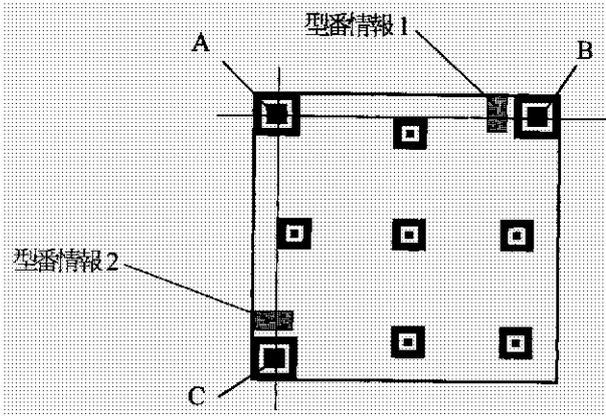


図 25 位置検出パターンと型番情報

- 3) 型番情報に適用されている拡張 BCH 誤り訂正 (附属書 D 参照) に基づき、誤りがあれば、それを検出訂正し、型番を決定する。
- 4) 誤り訂正能力を超える誤りを検出した場合、左下の位置検出要素パターンのパターン幅  $W_{DL}$  を計算し、同様の方法で上の手順 1)、2) 及び 3) を実施し、型番情報 2 を復号する。
- h) 位置合せパターンのない 1 型シンボルについては、附属書 M の手順 g) 及び h) を実施し、このアルゴリズムの手順 i) に戻る。2 型以上のシンボルについては、7.3.5 及び附属書 E に示す座標で定義する各位置合せパターンの中心座標を求め、サンプリンググリッドを設定する (図 26 参照)。

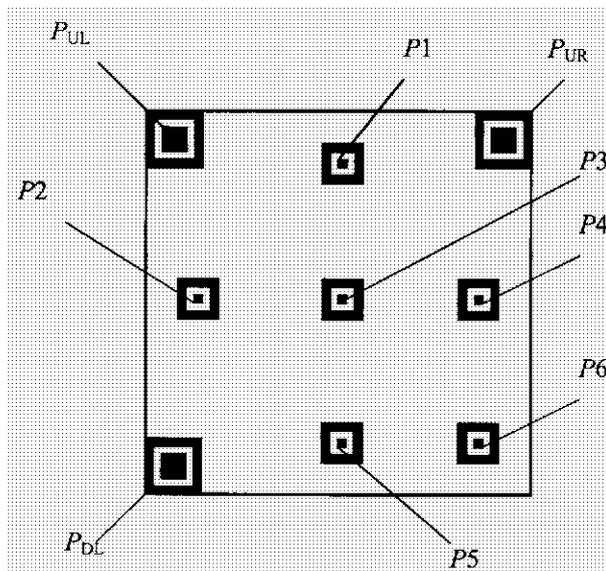


図 26 位置検出要素パターン及び位置合せパターン

- 1) 左上の位置検出要素パターン  $P_{UL}$  の幅  $W_{UL}$  を 7 で除し、モジュールの大きさ  $CP_{UL}$  を求める。  

$$CP_{UL} = W_{UL} / 7$$
- 2) 左上の位置検出要素パターン  $P_{UL}$  の中心座標  $A$ 、g) の 2) で求めた補助線 AC 及び AB の平行線並びにモジュールの大きさ  $CP_{UL}$  に基づいて、位置合せパターン  $P1$  及び  $P2$  の仮の中心座標を求める。
- 3) 仮の中心座標のピクセルから、それぞれ、位置合せパターン  $P1$  及び  $P2$  の白い正方形の外形を走査し、実際の中心座標  $X_i$  及び  $Y_j$  を求める (図 27 参照)。

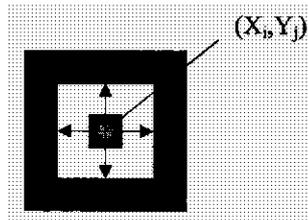


図 27 位置合せパターンの中心座標

- 4) 左上の位置検出要素パターン  $P_{UL}$  の中心座標並びに 3) で求めた位置合せパターン  $P1$  及び  $P2$  の実際の中心座標から、位置合せパターン  $P3$  の仮の中心座標を推定する。
- 5) 3) と同様の手順で、位置合せパターン  $P3$  の実際の中心座標を求める。
- 6) 位置合せパターン  $P2$  及び  $P3$  の中心距離  $L_x$  並びに位置合せパターン  $P1$  及び  $P3$  の中心距離  $L_y$  を求める。 $L_x$  及び  $L_y$  を定義されている位置合せパターン間隔で割り、シンボルの左上領域 (図 28 参照) の下辺及び右辺におけるモジュールピッチ  $CP_x$  及び  $CP_y$  を求める。

$$CP_x = L_x / AP$$

$$CP_y = L_y / AP$$

ここに、 $AP$  は、位置合せパターンの中心モジュール間隔 (附属書 E 表 1 参照)

同様に左上の位置検出要素パターン  $P_{UL}$  の中心座標及び位置合せパターン  $P1$  の中心座標の水平距離  $L_x'$ 、左上の位置検出要素パターンの中心座標及び位置合せパターン  $P2$  の中心座標の垂直距離  $L_y'$  を求める。 $L_x'$  及び  $L_y'$  を次の式の間隔で割り、シンボルの左上領域の上辺及び左辺におけるモジュールピッチ  $CL_x'$ 、 $CP_y'$  を求める。

$$CP_x' = L_x' / (\text{位置合せパターン } P1 \text{ の中心の列座標} - \text{左上の位置検出要素パターン } P_{UL} \text{ の中心の列座標})$$

$$CP_y' = L_y' / (\text{位置合せパターン } P2 \text{ の中心の行座標} - \text{左上の位置検出要素パターン } P_{UL} \text{ の中心の行座標})$$

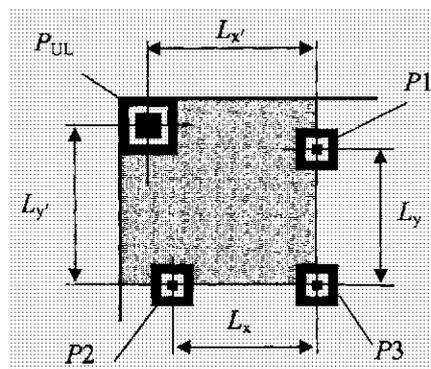


図 28 シンボルの左上領域

- 7) シンボルの左上領域の各辺のモジュールピッチ  $CP_x$ 、 $CP_x'$ 、 $CP_y$  及び  $CP_y'$  に基づき、左上の領域を網羅するサンプリンググリッドを設定する。
- 8) 同様に、シンボルの右上領域 (右上の位置検出要素パターン  $P_{UR}$ 、位置合せパターン  $P1$ 、 $P3$  及び  $P4$  で囲まれる) 及び左下領域 (左下の位置検出要素パターン  $P_{DL}$ 、位置合せパターン  $P2$ 、 $P3$  及び  $P5$  で囲まれる) のサンプリンググリッドを設定する。
- 9) 位置合せパターン  $P6$  (図 29 参照) については、位置合せパターン  $P3$ 、 $P4$  及び  $P5$  の間隔から求め

たモジュールピッチ  $CP_x$  及び  $CP_y$ ，位置合せパターン  $P3$  及び  $P4$  の中心を通る補助線， $P3$  及び  $P5$  の中心を通る補助線及びそれら位置合せパターンの中心座標から仮の中心座標を推定する。

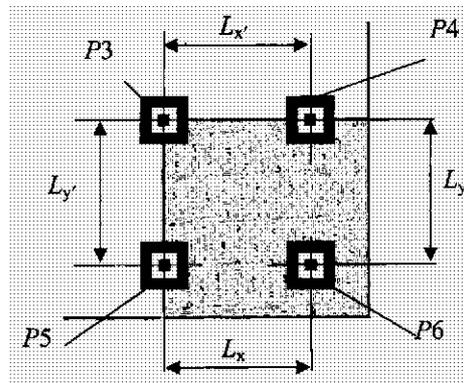


図 29 シンボルの右下領域

- 10) 手順 5)～8) を繰り返して，シンボルの右下領域のサンプリンググリッドを設定する。
- 11) 同様の原理で未処理の領域のサンプリンググリッドを設定する。
  - i) グリッド線の各交点上の画像ピクセルをサンプリングし，しきい値に基づき，それらが明暗のいずれかを定める。暗のピクセルを 2 進 1 とし，明のピクセルを 2 進 0 とするビットマトリックスを構築する。
  - j) 左上の位置検出要素パターンに隣接する形式情報を復号し，シンボルに適用された誤り訂正レベル及びマスクパターンを得る。形式情報の誤り訂正能力を超える誤りが検出された場合は，右上及び左下の位置検出要素パターンに隣接する形式情報を復号し，同様の手順を実施する。
  - k) シンボルの符号化領域でマスクパターンを XOR 演算することによって，マスク処理を解除し，データ及び誤り訂正を示すシンボルキャラクタを復元する。これは，符号化過程で適用されたマスク処理の手順を逆順することとする。
  - l) 8.7.3 の配置規則に従い，シンボルのコード語を求める。
  - m) シンボル型番及び誤り訂正レベルに応じ，8.6 の手順 c) に示すインタリーブの過程を逆順することによって，ブロック化されたコード語列を並び替える。
  - n) 附属書 B の誤り検出及び訂正復号手順に従い，シンボル型番及び誤り訂正レベルに対する最大訂正容量までの棄却誤り及び代入誤りを訂正する。
  - o) 列におけるデータブロックを結合することによって，元のメッセージを復元する。
  - p) データビット列をモード指示子で始まるセグメントに更に分割する。その長さは，モード指示子に続く文字数指示子によって決まる。
  - q) 適用中のモードの規則に従い，各セグメントを復号する。

14. 自動識別能力 QR コードは，他の多くのシンボル体系とともに自動識別環境で使用することができる（附属書 J 参照）。また，モデル 1 及びモデル 2 シンボルは，形式情報のマスクパターンを分析することによって自動識別が可能となる。

**15. 送信データ** すべての符号化データは、データ送信に含まれる。機能パターン、形式情報、型番情報、誤り訂正キャラクタ、埋め草及び残余キャラクタは、送信されない。すべてのデータに対する既定の送信モードは、8 ビット符号値又は 16 ビットシフト符号値とする。キャラクタ値が割り当てられているので、これは明らかに数字、英数字、片仮名及び漢字データの任意の列の送信となる。送信の前に全メッセージを再構築するバッファモードで運用されている場合、復号器によって構造的接続ヘッダブロックは、送信されないが、非バッファモードで運用されている場合、それぞれのシンボルの最初の 2 バイトである構造的接続ヘッダブロックは、送信される。拡張チャネル解釈におけるデータ送信を含むより複雑なものは、**15.2** による。

**15.1 データキャリア識別子** JIS X 0530 は、復号器に設定される任意機能及びシンボルに見られる特別な特徴とともに読み出されるシンボルの送信に関する標準手順を規定している。

データ（任意の ECI の使用を含む。）の構造を識別すると、復号器は、適切なデータキャリア識別子を送信データの前に付加する。ECI が使用されている場合、データキャリア識別子を必要とする。データキャリア識別子及び QR コードに適用される任意機能値に関しては、**附属書 F** による。

**15.2 拡張チャネル解釈** ECI プロトコルを使用できるシステムでは、送信ごとにシンボル体系識別子の送信を必要とする。ECI モード指示子がある場合、それはエスケープ文字  $5C_{HEX}$  として送信する。 $5C_{HEX}$  は、JIS X 0201（ECI 割当て文書及び ISO 646 IRV の文字集合において、この“¥”の値は、逆斜線“\”とする。）に基づく QR コードの既定の符号化において、“¥”で示す。ECI 指定を示すコード語は、**表 4** に定義されている規則で 6 けたの数字に変換する。この 6 けたの数字は、エスケープ文字に続いて  $30_{HEX}$  ~  $39_{HEX}$  までの範囲で対応する 8 ビット値として送信する。

\nnnnnn を認識するアプリケーションソフトウェアは、それに続くすべてのキャラクタは 6 けたの指定によって定義される ECI からのものとして解釈する。この解釈は次のことが発生するまで有効とする。

a) 符号化データの最後

b) モード指示子 0111 による別の ECI への変更（AIM Inc. の ECI 仕様に従う。）

既定の解釈へ復帰する場合、復号器は、データの前に適切なエスケープ列を付加し、出力する。

文字“¥”を符号化データとして使用する必要がある場合、 $5C_{HEX}$  を 2 個送信する。 $5C_{HEX}$  の 1 回の出現は常にエスケープ文字であり、2 回の出現はデータを意味する。

**例1.**

符号化データ	ABC¥1234
送信データ	ABC¥¥1234
符号化データ	ABC に続く ECI123456 の規則による〈あるデータ〉の符号化
送信データ	ABC¥123456 〈あるデータ〉

**例2. (8.4.1.1 のデータを使用)**

メッセージは、ECI モード指示子、ECI 指定、モード指示子、文字数指示子及びデータで次のとおり構成する。

0111 00001001 0100 00000101 10100001 10100010 10100011 10100100 10100101

シンボル体系識別子] Q1 (**附属書 F** 参照) は、送信データに付加されなければならない。

送信 (16 進値)            5D 51 31 5C 30 30 30 30 39 A1 A2 A3 A4 A5

000009 における符号化データ    ABΓΔE

構造的接続モードで、ECI モード指示子のいずれかがシンボルの先頭に出現した場合、後続するデータ文字は、前のシンボルの最後で適用されていた ECI からのものとして解釈する。

**備考** 逆斜線“\” (ASCII 値 5C<sub>HEX</sub>) は JIS X 0201 における“¥” と等価とする。

**15.3 FNC1** FNC1 が 1 番目又は 2 番目の位置に適用されているモードにおいて、FNC1 に対応する文字が存在しないので、適用されている文字を直接、送信することはできない。したがって、適切なシンボル識別子 (JQ3, JQ4, JQ5 又は JQ6) の送信によって、1 番目又は 2 番目の位置における FNC1 の存在を示す必要がある。これらのシンボルにおける他の位置において、FNC1 は、適切なアプリケーション仕様に従うデータのフィールド分離子として使用される場合がある。それは、英数字モードでは%で、8 ビットバイトモードでは IS3 として表現する。両方の場合において、復号器は 1D<sub>HEX</sub>を送信する。

英数字モードで%が符号化データの一部として使用されている場合、シンボル内でその文字は%%として表現する。%%が出現すると、復号器は 1 個の%を送信する。

## 附属書 A (規定) 誤り検出及び訂正の生成多項式

誤り訂正の生成多項式は、データコード語の多項式の除算に使用する。ここで各コード語は、累乗の降順に除算される多項式の係数となる。この除算の剰余の係数は、誤り訂正コード語の値となる。

附属書 A 表 1~5 は、各型番及びレベルで使用する誤り訂正コードに対する生成多項式を示す。同表において、 $\alpha$  は GF(2<sup>8</sup>) 上の原始要素 2 の根とする。各生成多項式は 1 次多項式  $x-2^0$ ,  $x-2^1$ , ...,  $x-2^{n-1}$  から生成される。ここに、 $n$  は、生成多項式の次数とする。

附属書 A 表 1 誤り訂正コードの生成多項式 (誤り訂正コード語数 : 7~26)

誤り訂正 コード語数	生成多項式
7	$x^7 + \alpha^{87}x^6 + \alpha^{229}x^5 + \alpha^{146}x^4 + \alpha^{149}x^3 + \alpha^{238}x^2 + \alpha^{102}x + \alpha^{21}$
10	$x^{10} + \alpha^{251}x^9 + \alpha^{67}x^8 + \alpha^{61}x^6 + \alpha^{118}x^5 + \alpha^{70}x^4 + \alpha^{64}x^3 + \alpha^{94}x^2 + \alpha^{32}x + \alpha^{45}$
13	$x^{13} + \alpha^{74}x^{12} + \alpha^{152}x^{11} + \alpha^{176}x^{10} + \alpha^{100}x^9 + \alpha^{86}x^8 + \alpha^{100}x^7 + \alpha^{106}x^6 + \alpha^{104}x^5 + \alpha^{130}x^4 + \alpha^{218}x^3 + \alpha^{206}x^2 + \alpha^{140}x + \alpha^{78}$
15	$x^{15} + \alpha^{8}x^{14} + \alpha^{183}x^{13} + \alpha^{61}x^{12} + \alpha^{91}x^{11} + \alpha^{202}x^{10} + \alpha^{37}x^9 + \alpha^{51}x^8 + \alpha^{58}x^7 + \alpha^{58}x^6 + \alpha^{237}x^5 + \alpha^{40}x^4 + \alpha^{124}x^3 + \alpha^5x^2 + \alpha^{99}x + \alpha^{105}$
16	$x^{16} + \alpha^{120}x^{15} + \alpha^{104}x^{14} + \alpha^{107}x^{13} + \alpha^{109}x^{12} + \alpha^{102}x^{11} + \alpha^{161}x^{10} + \alpha^{76}x^9 + \alpha^3x^8 + \alpha^{91}x^7 + \alpha^{191}x^6 + \alpha^{147}x^5 + \alpha^{169}x^4 + \alpha^{182}x^3 + \alpha^{194}x^2 + \alpha^{225}x + \alpha^{120}$
17	$x^{17} + \alpha^{43}x^{16} + \alpha^{139}x^{15} + \alpha^{206}x^{14} + \alpha^{78}x^{13} + \alpha^{43}x^{12} + \alpha^{239}x^{11} + \alpha^{123}x^{10} + \alpha^{206}x^9 + \alpha^{214}x^8 + \alpha^{147}x^7 + \alpha^{24}x^6 + \alpha^{99}x^5 + \alpha^{150}x^4 + \alpha^{39}x^3 + \alpha^{243}x^2 + \alpha^{163}x + \alpha^{136}$
18	$x^{18} + \alpha^{215}x^{17} + \alpha^{234}x^{16} + \alpha^{158}x^{15} + \alpha^{94}x^{14} + \alpha^{184}x^{13} + \alpha^{97}x^{12} + \alpha^{118}x^{11} + \alpha^{170}x^{10} + \alpha^{79}x^9 + \alpha^{187}x^8 + \alpha^{152}x^7 + \alpha^{148}x^6 + \alpha^{252}x^5 + \alpha^{179}x^4 + \alpha^5x^3 + \alpha^{98}x^2 + \alpha^{96}x + \alpha^{153}$
20	$x^{20} + \alpha^{17}x^{19} + \alpha^{60}x^{18} + \alpha^{79}x^{17} + \alpha^{50}x^{16} + \alpha^{61}x^{15} + \alpha^{163}x^{14} + \alpha^{26}x^{13} + \alpha^{187}x^{12} + \alpha^{202}x^{11} + \alpha^{180}x^{10} + \alpha^{221}x^9 + \alpha^{225}x^8 + \alpha^{83}x^7 + \alpha^{239}x^6 + \alpha^{156}x^5 + \alpha^{164}x^4 + \alpha^{212}x^3 + \alpha^{212}x^2 + \alpha^{188}x + \alpha^{190}$
22	$x^{22} + \alpha^{210}x^{21} + \alpha^{171}x^{20} + \alpha^{247}x^{19} + \alpha^{242}x^{18} + \alpha^{93}x^{17} + \alpha^{230}x^{16} + \alpha^{14}x^{15} + \alpha^{109}x^{14} + \alpha^{221}x^{13} + \alpha^{53}x^{12} + \alpha^{200}x^{11} + \alpha^{74}x^{10} + \alpha^8x^9 + \alpha^{172}x^8 + \alpha^{98}x^7 + \alpha^{80}x^6 + \alpha^{219}x^5 + \alpha^{134}x^4 + \alpha^{160}x^3 + \alpha^{105}x^2 + \alpha^{165}x + \alpha^{231}$
24	$x^{24} + \alpha^{229}x^{23} + \alpha^{121}x^{22} + \alpha^{135}x^{21} + \alpha^{48}x^{20} + \alpha^{211}x^{19} + \alpha^{117}x^{18} + \alpha^{251}x^{17} + \alpha^{126}x^{16} + \alpha^{159}x^{15} + \alpha^{180}x^{14} + \alpha^{169}x^{13} + \alpha^{152}x^{12} + \alpha^{192}x^{11} + \alpha^{226}x^{10} + \alpha^{228}x^9 + \alpha^{218}x^8 + \alpha^{111}x^7 + \alpha^6x^6 + \alpha^{117}x^5 + \alpha^{232}x^4 + \alpha^{87}x^3 + \alpha^{96}x^2 + \alpha^{227}x + \alpha^{21}$
26	$x^{26} + \alpha^{173}x^{25} + \alpha^{125}x^{24} + \alpha^{158}x^{23} + \alpha^2x^{22} + \alpha^{103}x^{21} + \alpha^{182}x^{20} + \alpha^{118}x^{19} + \alpha^{17}x^{18} + \alpha^{145}x^{17} + \alpha^{201}x^{16} + \alpha^{111}x^{15} + \alpha^{28}x^{14} + \alpha^{165}x^{13} + \alpha^{53}x^{12} + \alpha^{161}x^{11} + \alpha^{21}x^{10} + \alpha^{245}x^9 + \alpha^{142}x^8 + \alpha^{13}x^7 + \alpha^{102}x^6 + \alpha^{48}x^5 + \alpha^{227}x^4 + \alpha^{153}x^3 + \alpha^{145}x^2 + \alpha^{218}x + \alpha^{70}$

附属書 A 表 2 誤り訂正コードの生成多項式 (誤り訂正コード語数 : 28~42)

誤り訂正 コード語数	生成多項式
28	$x^{28} + a^{168}x^{27} + a^{223}x^{26} + a^{200}x^{25} + a^{104}x^{24} + a^{224}x^{23} + a^{234}x^{22} + a^{108}x^{21} + a^{180}x^{20}$ $+ a^{110}x^{19} + a^{190}x^{18} + a^{195}x^{17} + a^{147}x^{16} + a^{205}x^{15} + a^{27}x^{14} + a^{232}x^{13} + a^{201}x^{12}$ $+ a^{21}x^{11} + a^{43}x^{10} + a^{245}x^9 + a^{87}x^8 + a^{42}x^7 + a^{195}x^6 + a^{212}x^5 + a^{119}x^4 + a^{242}x^3 +$ $a^{37}x^2 + a^9x + a^{123}$
30	$x^{30} + a^{41}x^{29} + a^{173}x^{28} + a^{145}x^{27} + a^{152}x^{26} + a^{216}x^{25} + a^{31}x^{24} + a^{179}x^{23} + a^{182}x^{22}$ $+ a^{50}x^{21} + a^{48}x^{20} + a^{110}x^{19} + a^{86}x^{18} + a^{239}x^{17} + a^{96}x^{16} + a^{222}x^{15} + a^{125}x^{14} +$ $a^{42}x^{13} + a^{173}x^{12} + a^{226}x^{11} + a^{193}x^{10} + a^{224}x^9 + a^{130}x^8 + a^{156}x^7 + a^{37}x^6 + a^{251}x^5 +$ $a^{216}x^4 + a^{238}x^3 + a^{40}x^2 + a^{192}x + a^{180}$
32	$x^{32} + a^{10}x^{31} + a^6x^{30} + a^{106}x^{29} + a^{190}x^{28} + a^{249}x^{27} + a^{167}x^{26} + a^4x^{25} + a^{67}x^{24} +$ $a^{209}x^{23} + a^{138}x^{22} + a^{138}x^{21} + a^{32}x^{20} + a^{242}x^{19} + a^{123}x^{18} + a^{89}x^{17} + a^{27}x^{16} +$ $a^{120}x^{15}$ $+ a^{185}x^{14} + a^{80}x^{13} + a^{156}x^{12} + a^{38}x^{11} + a^{60}x^{10} + a^{171}x^9 + a^{60}x^8 + a^{28}x^7 + a^{222}x^6 +$ $a^{80}x^5 + a^{52}x^4 + a^{254}x^3 + a^{185}x^2 + a^{220}x + a^{241}$
34	$x^{34} + a^{111}x^{33} + a^{77}x^{32} + a^{146}x^{31} + a^{94}x^{30} + a^{26}x^{29} + a^{21}x^{28} + a^{108}x^{27} + a^{19}x^{26} +$ $a^{105}x^{25} + a^{94}x^{24} + a^{113}x^{23} + a^{193}x^{22} + a^{86}x^{21} + a^{140}x^{20} + a^{163}x^{19} + a^{125}x^{18} +$ $a^{58}x^{17}$ $+ a^{158}x^{16} + a^{229}x^{15} + a^{239}x^{14} + a^{218}x^{13} + a^{103}x^{12} + a^{56}x^{11} + a^{70}x^{10} + a^{114}x^9 +$ $a^{61}x^8 + a^{183}x^7 + a^{129}x^6 + a^{167}x^5 + a^{13}x^4 + a^{98}x^3 + a^{62}x^2 + a^{129}x + a^{51}$
36	$x^{36} + a^{200}x^{35} + a^{183}x^{34} + a^{98}x^{33} + a^{16}x^{32} + a^{172}x^{31} + a^{31}x^{30} + a^{246}x^{29} + a^{234}x^{28}$ $+ a^{60}x^{27} + a^{152}x^{26} + a^{115}x^{25} + x^{24} + a^{167}x^{23} + a^{152}x^{22} + a^{113}x^{21} + a^{248}x^{20} +$ $a^{238}x^{19} + a^{107}x^{18} + a^{18}x^{17} + a^{63}x^{16} + a^{218}x^{15} + a^{37}x^{14} + a^{87}x^{13} + a^{210}x^{12} +$ $a^{105}x^{11}$ $+ a^{177}x^{10} + a^{120}x^9 + a^{74}x^8 + a^{121}x^7 + a^{196}x^6 + a^{117}x^5 + a^{251}x^4 + a^{113}x^3 + a^{233}x^2 +$ $a^{30}x + a^{120}$
40	$x^{40} + a^{59}x^{39} + a^{116}x^{38} + a^{79}x^{37} + a^{161}x^{36} + a^{252}x^{35} + a^{98}x^{34} + a^{128}x^{33} + a^{205}x^{32}$ $+ a^{128}x^{31} + a^{161}x^{30} + a^{247}x^{29} + a^{57}x^{28} + a^{163}x^{27} + a^{56}x^{26} + a^{235}x^{25} + a^{106}x^{24}$ $+ a^{53}x^{23} + a^{26}x^{22} + a^{187}x^{21} + a^{174}x^{20} + a^{226}x^{19} + a^{104}x^{18} + a^{170}x^{17} + a^7x^{16}$ $+ a^{175}x^{15} + a^{35}x^{14} + a^{181}x^{13} + a^{114}x^{12} + a^{88}x^{11} + a^{41}x^{10} + a^{47}x^9 + a^{163}x^8 +$ $a^{125}x^7 + a^{134}x^6 + a^{72}x^5 + a^{20}x^4 + a^{232}x^3 + a^{53}x^2 + a^{35}x + a^{15}$
42	$x^{42} + a^{250}x^{41} + a^{103}x^{40} + a^{221}x^{39} + a^{230}x^{38} + a^{25}x^{37} + a^{18}x^{36} + a^{137}x^{35} + a^{231}x^{34} +$ $x^{33} + a^3x^{32} + a^{58}x^{31} + a^{242}x^{30} + a^{21}x^{29} + a^{191}x^{28} + a^{110}x^{27} + a^{84}x^{26} + a^{230}x^{25} +$ $a^8x^{24} + a^{188}x^{23} + a^{106}x^{22} + a^{96}x^{21} + a^{147}x^{20} + a^{15}x^{19} + a^{131}x^{18} + a^{139}x^{17} +$ $a^{34}x^{16} + a^{101}x^{15} + a^{223}x^{14} + a^{39}x^{13} + a^{101}x^{12} + a^{213}x^{11} + a^{199}x^{10} + a^{237}x^9 +$ $a^{254}x^8 + a^{201}x^7 + a^{123}x^6 + a^{171}x^5 + a^{162}x^4 + a^{194}x^3 + a^{117}x^2 + a^{50}x + a^{96}$

附属書 A 表 3 誤り訂正コードの生成多項式 (誤り訂正コード語数 : 44~52)

誤り訂正 コード語数	生成多項式
44	$x^{44} + a^{190}x^{43} + a^7x^{42} + a^{61}x^{41} + a^{121}x^{40} + a^{71}x^{39} + a^{246}x^{38} + a^{69}x^{37} + a^{55}x^{36} + a^{168}x^{35} + a^{188}x^{34} + a^{89}x^{33} + a^{243}x^{32} + a^{191}x^{31} + a^{25}x^{30} + a^{72}x^{29} + a^{123}x^{28} + a^9x^{27} + a^{145}x^{26} + a^{14}x^{25} + a^{247}x^{24} + ax^{23} + a^{238}x^{22} + a^{44}x^{21} + a^{78}x^{20} + a^{143}x^{19} + a^{62}x^{18} + a^{224}x^{17} + a^{126}x^{16} + a^{118}x^{15} + a^{114}x^{14} + a^{68}x^{13} + a^{163}x^{12} + a^{52}x^{11} + a^{194}x^{10} + a^{217}x^9 + a^{147}x^8 + a^{204}x^7 + a^{169}x^6 + a^{37}x^5 + a^{130}x^4 + a^{113}x^3 + a^{102}x^2 + a^{73}x + a^{81}$
46	$x^{46} + a^{112}x^{45} + a^{94}x^{44} + a^{88}x^{43} + a^{112}x^{42} + a^{233}x^{41} + a^{224}x^{40} + a^{202}x^{39} + a^{115}x^{38} + a^{187}x^{37} + a^{99}x^{36} + a^{89}x^{35} + a^5x^{34} + a^{54}x^{33} + a^{113}x^{32} + a^{129}x^{31} + a^{44}x^{30} + a^{58}x^{29} + a^{16}x^{28} + a^{135}x^{27} + a^{216}x^{26} + a^{169}x^{25} + a^{211}x^{24} + a^{36}x^{23} + ax^{22} + a^{42}x^{21} + a^{96}x^{20} + a^{60}x^{19} + a^{241}x^{18} + a^{73}x^{17} + a^{104}x^{16} + a^{234}x^{15} + a^8x^{14} + a^{249}x^{13} + a^{245}x^{12} + a^{119}x^{11} + a^{174}x^{10} + a^{52}x^9 + a^{25}x^8 + a^{157}x^7 + a^{224}x^6 + a^{43}x^5 + a^{202}x^4 + a^{223}x^3 + a^{19}x^2 + a^{82}x + a^{15}$
48	$x^{48} + a^{228}x^{47} + a^{25}x^{46} + a^{196}x^{45} + a^{130}x^{44} + a^{211}x^{43} + a^{146}x^{42} + a^{60}x^{41} + a^{24}x^{40} + a^{251}x^{39} + a^{90}x^{38} + a^{39}x^{37} + a^{102}x^{36} + a^{240}x^{35} + a^{61}x^{34} + a^{178}x^{33} + a^{63}x^{32} + a^{46}x^{31} + a^{123}x^{30} + a^{115}x^{29} + a^{18}x^{28} + a^{221}x^{27} + a^{111}x^{26} + a^{135}x^{25} + a^{160}x^{24} + a^{182}x^{23} + a^{205}x^{22} + a^{107}x^{21} + a^{206}x^{20} + a^{95}x^{19} + a^{150}x^{18} + a^{120}x^{17} + a^{184}x^{16} + a^9x^{15} + a^{21}x^{14} + a^{247}x^{13} + a^{156}x^{12} + a^{140}x^{11} + a^{238}x^{10} + a^{191}x^9 + a^{11}x^8 + a^{94}x^7 + a^{227}x^6 + a^{84}x^5 + a^{50}x^4 + a^{163}x^3 + a^{39}x^2 + a^{34}x + a^{108}$
50	$x^{50} + a^{232}x^{49} + a^{125}x^{48} + a^{157}x^{47} + a^{161}x^{46} + a^{164}x^{45} + a^9x^{44} + a^{118}x^{43} + a^{46}x^{42} + a^{209}x^{41} + a^{99}x^{40} + a^{203}x^{39} + a^{193}x^{38} + a^{35}x^{37} + a^3x^{36} + a^{209}x^{35} + a^{111}x^{34} + a^{195}x^{33} + a^{242}x^{32} + a^{203}x^{31} + a^{225}x^{30} + a^{46}x^{29} + a^{13}x^{28} + a^{32}x^{27} + a^{160}x^{26} + a^{126}x^{25} + a^{209}x^{24} + a^{130}x^{23} + a^{160}x^{22} + a^{242}x^{21} + a^{215}x^{20} + a^{242}x^{19} + a^{75}x^{18} + a^{77}x^{17} + a^{42}x^{16} + a^{189}x^{15} + a^{32}x^{14} + a^{113}x^{13} + a^{65}x^{12} + a^{124}x^{11} + a^{69}x^{10} + a^{228}x^9 + a^{114}x^8 + a^{235}x^7 + a^{175}x^6 + a^{124}x^5 + a^{170}x^4 + a^{215}x^3 + a^{232}x^2 + a^{133}x + a^{205}$
52	$x^{52} + a^{116}x^{51} + a^{50}x^{50} + a^{86}x^{49} + a^{186}x^{48} + a^{50}x^{47} + a^{220}x^{46} + a^{251}x^{45} + a^{89}x^{44} + a^{192}x^{43} + a^{46}x^{42} + a^{86}x^{41} + a^{127}x^{40} + a^{124}x^{39} + a^{19}x^{38} + a^{184}x^{37} + a^{233}x^{36} + a^{151}x^{35} + a^{215}x^{34} + a^{22}x^{33} + a^{14}x^{32} + a^{59}x^{31} + a^{145}x^{30} + a^{37}x^{29} + a^{242}x^{28} + a^{203}x^{27} + a^{134}x^{26} + a^{254}x^{25} + a^{89}x^{24} + a^{190}x^{23} + a^{94}x^{22} + a^{59}x^{21} + a^{65}x^{20} + a^{124}x^{19} + a^{113}x^{18} + a^{100}x^{17} + a^{233}x^{16} + a^{235}x^{15} + a^{121}x^{14} + a^{22}x^{13} + a^{76}x^{12} + a^{86}x^{11} + a^{97}x^{10} + a^{39}x^9 + a^{242}x^8 + a^{200}x^7 + a^{220}x^6 + a^{101}x^5 + a^{33}x^4 + a^{239}x^3 + a^{254}x^2 + a^{116}x^{51}$

附属書 A 表 4 誤り訂正コードの生成多項式 (誤り訂正コード語数 : 54~60)

誤り訂正 コード語数	生成多項式
54	$  \begin{aligned}  &x^{54} + a^{183}x^{53} + a^{26}x^{52} + a^{201}x^{51} + a^{87}x^{50} + a^{210}x^{49} + a^{221}x^{48} + a^{113}x^{47} + a^{21}x^{46} \\  &+ a^{46}x^{45} + a^{65}x^{44} + a^{45}x^{43} + a^{50}x^{42} + a^{238}x^{41} + a^{184}x^{40} + a^{249}x^{39} + a^{225}x^{38} \\  &+ a^{102}x^{37} + a^{58}x^{36} + a^{209}x^{35} + a^{218}x^{34} + a^{109}x^{33} + a^{165}x^{32} + a^{26}x^{31} + a^{95}x^{30} \\  &+ a^{184}x^{29} + a^{192}x^{28} + a^{52}x^{27} + a^{245}x^{26} + a^{35}x^{25} + a^{254}x^{24} + a^{238}x^{23} + a^{175}x^{22} \\  &+ a^{172}x^{21} + a^{79}x^{20} + a^{123}x^{19} + a^{25}x^{18} + a^{122}x^{17} + a^{43}x^{16} + a^{120}x^{15} + a^{108}x^{14} \\  &+ a^{215}x^{13} + a^{80}x^{12} + a^{128}x^{11} + a^{201}x^{10} + a^{235}x^9 + a^8x^8 + a^{153}x^7 + a^{59}x^6 + a^{101}x^5 \\  &+ a^{31}x^4 + a^{198}x^3 + a^{76}x^2 + a^{31}x + a^{156}  \end{aligned}  $
56	$  \begin{aligned}  &x^{56} + a^{106}x^{55} + a^{120}x^{54} + a^{107}x^{53} + a^{157}x^{52} + a^{164}x^{51} + a^{216}x^{50} + a^{112}x^{49} + a^{116}x^{48} \\  &+ a^2x^{47} + a^{91}x^{46} + a^{248}x^{45} + a^{163}x^{44} + a^{36}x^{43} + a^{201}x^{42} + a^{202}x^{41} + a^{229}x^{40} + \\  &a^6x^{39} + a^{144}x^{38} + a^{254}x^{37} + a^{155}x^{36} + a^{135}x^{35} + a^{208}x^{34} + a^{170}x^{33} + a^{209}x^{32} + \\  &a^{12}x^{31} \\  &+ a^{139}x^{30} + a^{127}x^{29} + a^{142}x^{28} + a^{182}x^{27} + a^{249}x^{26} + a^{177}x^{25} + a^{174}x^{24} + a^{190}x^{23} \\  &+ a^{28}x^{22} + a^{10}x^{21} + a^{85}x^{20} + a^{239}x^{19} + a^{184}x^{18} + a^{101}x^{17} + a^{124}x^{16} + a^{152}x^{15} \\  &+ a^{206}x^{14} + a^{96}x^{13} + a^{23}x^{12} + a^{163}x^{11} + a^{61}x^{10} + a^{27}x^9 + a^{196}x^8 + a^{247}x^7 + a^{151}x^6 \\  &+ a^{154}x^5 + a^{202}x^4 + a^{207}x^3 + a^{20}x^2 + a^{61}x + a^{10}  \end{aligned}  $
58	$  \begin{aligned}  &x^{58} + a^{82}x^{57} + a^{116}x^{56} + a^{26}x^{55} + a^{247}x^{54} + a^{66}x^{53} + a^{27}x^{52} + a^{62}x^{51} + a^{107}x^{50} + \\  &a^{252}x^{49} + a^{182}x^{48} + a^{200}x^{47} + a^{185}x^{46} + a^{235}x^{45} + a^{55}x^{44} + a^{251}x^{43} + a^{242}x^{42} + \\  &a^{210}x^{41} \\  &+ a^{144}x^{40} + a^{154}x^{39} + a^{237}x^{38} + a^{176}x^{37} + a^{141}x^{36} + a^{192}x^{35} + a^{248}x^{34} + a^{152}x^{33} \\  &+ a^{249}x^{32} + a^{206}x^{31} + a^{85}x^{30} + a^{253}x^{29} + a^{142}x^{28} + a^{65}x^{27} + a^{165}x^{26} + a^{125}x^{25} \\  &+ a^{23}x^{24} + a^{24}x^{23} + a^{30}x^{22} + a^{122}x^{21} + a^{240}x^{20} + a^{214}x^{19} + a^{6}x^{18} + a^{129}x^{17} + \\  &a^{218}x^{16} + a^{29}x^{15} + a^{145}x^{14} + a^{127}x^{13} + a^{134}x^{12} + a^{206}x^{11} + a^{245}x^{10} + a^{117}x^9 + \\  &a^{29}x^8 + a^{41}x^7 + a^{63}x^6 + a^{159}x^5 + a^{142}x^4 + a^{233}x^3 + a^{125}x^2 + a^{148}x + a^{123}  \end{aligned}  $
60	$  \begin{aligned}  &x^{60} + a^{107}x^{59} + a^{140}x^{58} + a^{26}x^{57} + a^{12}x^{56} + a^9x^{55} + a^{141}x^{54} + a^{243}x^{53} + a^{197}x^{52} \\  &+ a^{226}x^{51} + a^{197}x^{50} + a^{219}x^{49} + a^{45}x^{48} + a^{211}x^{47} + a^{101}x^{46} + a^{219}x^{45} + a^{120}x^{44} \\  &+ a^{28}x^{43} + a^{181}x^{42} + a^{127}x^{41} + a^6x^{40} + a^{100}x^{39} + a^{247}x^{38} + a^2x^{37} + a^{205}x^{36} + \\  &a^{198}x^{35} + a^{57}x^{34} + a^{115}x^{33} + a^{219}x^{32} + a^{101}x^{31} + a^{109}x^{30} + a^{160}x^{29} + a^{82}x^{28} + \\  &a^{37}x^{27} \\  &+ a^{38}x^{26} + a^{238}x^{25} + a^{49}x^{24} + a^{160}x^{23} + a^{209}x^{22} + a^{121}x^{21} + a^{86}x^{20} + a^{11}x^{19} \\  &+ a^{124}x^{18} + a^{30}x^{17} + a^{181}x^{16} + a^{84}x^{15} + a^{25}x^{14} + a^{194}x^{13} + a^{87}x^{12} + a^{65}x^{11} + \\  &a^{102}x^{10} + a^{190}x^9 + a^{220}x^8 + a^{70}x^7 + a^{27}x^6 + a^{209}x^5 + a^{16}x^4 + a^{89}x^3 + a^7x^2 + a^{33}x \\  &+ a^{240}  \end{aligned}  $

附属書 A 表 5 誤り訂正コードの生成多項式 (誤り訂正コード語数 : 62~68)

誤り訂正 コード語数	生成多項式
62	$  \begin{aligned}  &x^{62} + a^{65}x^{61} + a^{202}x^{60} + a^{113}x^{59} + a^{98}x^{58} + a^{71}x^{57} + a^{223}x^{56} + a^{248}x^{55} + a^{118}x^{54} \\  &+ a^{214}x^{53} + a^{94}x^{52} + x^{51} + a^{122}x^{50} + a^{37}x^{49} + a^{23}x^{48} + a^2x^{47} + a^{228}x^{46} + a^{58}x^{45} \\  &+ a^{121}x^{44} + a^7x^{43} + a^{105}x^{42} + a^{135}x^{41} + a^{78}x^{40} + a^{243}x^{39} + a^{118}x^{38} + a^{70}x^{37} + \\  &a^{76}x^{36} + a^{223}x^{35} + a^{89}x^{34} + a^{72}x^{33} + a^{50}x^{32} + a^{70}x^{31} + a^{111}x^{30} + a^{194}x^{29} + a^{17}x^{28} \\  &+ a^{212}x^{27} + a^{126}x^{26} + a^{181}x^{25} + a^{35}x^{24} + a^{221}x^{23} + a^{117}x^{22} + a^{235}x^{21} + a^{11}x^{20} + \\  &a^{229}x^{19} \\  &+ a^{149}x^{18} + a^{147}x^{17} + a^{123}x^{16} + a^{213}x^{15} + a^{40}x^{14} + a^{115}x^{13} + a^6x^{12} + a^{200}x^{11} \\  &+ a^{100}x^{10} + a^{26}x^9 + a^{246}x^8 + a^{182}x^7 + a^{218}x^6 + a^{215}x^4 + a^{36}x^3 + a^{186}x^2 + a^{110}x \\  &+ a^{106}  \end{aligned}  $
64	$  \begin{aligned}  &x^{64} + a^{45}x^{63} + a^{51}x^{62} + a^{175}x^{61} + a^9x^{60} + a^7x^{59} + a^{158}x^{58} + a^{159}x^{57} + a^{49}x^{56} + \\  &a^{68}x^{55} \\  &+ a^{119}x^{54} + a^{92}x^{53} + a^{123}x^{52} + a^{177}x^{51} + a^{204}x^{50} + a^{187}x^{49} + a^{254}x^{48} + a^{200}x^{47} \\  &+ a^{78}x^{46} + a^{141}x^{45} + a^{149}x^{44} + a^{119}x^{43} + a^{26}x^{42} + a^{127}x^{41} + a^{53}x^{40} + a^{160}x^{39} \\  &+ a^{93}x^{38} + a^{199}x^{37} + a^{212}x^{36} + a^{29}x^{35} + a^{24}x^{34} + a^{145}x^{33} + a^{156}x^{32} + a^{208}x^{31} \\  &+ a^{150}x^{30} + a^{218}x^{29} + a^{209}x^{28} + a^4x^{27} + a^{216}x^{26} + a^{91}x^{25} + a^{47}x^{24} + a^{184}x^{23} \\  &+ a^{146}x^{22} + a^{47}x^{21} + a^{140}x^{20} + a^{195}x^{19} + a^{195}x^{18} + a^{125}x^{17} + a^{242}x^{16} + a^{238}x^{15} \\  &+ a^{63}x^{14} + a^{99}x^{13} + a^{108}x^{12} + a^{140}x^{11} + a^{230}x^{10} + a^{242}x^9 + a^{31}x^8 + a^{204}x^7 + a^{11}x^6 \\  &+ a^{178}x^5 + a^{243}x^4 + a^{217}x^3 + a^{156}x^2 + a^{213}x + a^{231}  \end{aligned}  $
66	$  \begin{aligned}  &x^{66} + a^5x^{65} + a^{118}x^{64} + a^{222}x^{63} + a^{180}x^{62} + a^{136}x^{61} + a^{136}x^{60} + a^{162}x^{59} + a^{51}x^{58} \\  &+ a^{46}x^{57} + a^{117}x^{56} + a^{13}x^{55} + a^{215}x^{54} + a^{81}x^{53} + a^{17}x^{52} + a^{139}x^{51} + a^{247}x^{50} \\  &+ a^{197}x^{49} + a^{171}x^{48} + a^{95}x^{47} + a^{173}x^{46} + a^{65}x^{45} + a^{137}x^{44} + a^{178}x^{43} + a^{68}x^{42} \\  &+ a^{111}x^{41} + a^9x^{40} + a^{101}x^{39} + a^{41}x^{38} + a^{72}x^{37} + a^{214}x^{36} + a^{169}x^{35} + a^{197}x^{34} \\  &+ a^{95}x^{33} + a^7x^{32} + a^{44}x^{31} + a^{154}x^{30} + a^{77}x^{29} + a^{111}x^{28} + a^{236}x^{27} + a^{40}x^{26} + \\  &a^{121}x^{25} + a^{143}x^{24} + a^{63}x^{23} + a^{87}x^{22} + a^{80}x^{21} + a^{253}x^{20} + a^{240}x^{19} + a^{126}x^{18} + \\  &a^{217}x^{17} \\  &+ a^{77}x^{16} + a^{34}x^{15} + a^{232}x^{14} + a^{106}x^{13} + a^{50}x^{12} + a^{168}x^{11} + a^{82}x^{10} + a^{76}x^9 + \\  &a^{146}x^8 + a^{67}x^7 + a^{106}x^6 + a^{171}x^5 + a^{25}x^4 + a^{132}x^3 + a^{93}x^2 + a^{45}x + a^{105}  \end{aligned}  $
68	$  \begin{aligned}  &x^{68} + a^{247}x^{67} + a^{159}x^{66} + a^{223}x^{65} + a^{33}x^{64} + a^{224}x^{63} + a^{93}x^{62} + a^{77}x^{61} + a^{70}x^{60} + \\  &a^{90}x^{59} + a^{160}x^{58} + a^{32}x^{57} + a^{254}x^{56} + a^{43}x^{55} + a^{150}x^{54} + a^{84}x^{53} + a^{101}x^{52} + \\  &a^{190}x^{51} \\  &+ a^{205}x^{50} + a^{133}x^{49} + a^{52}x^{48} + a^{60}x^{47} + a^{202}x^{46} + a^{165}x^{45} + a^{220}x^{44} + a^{203}x^{43} \\  &+ a^{151}x^{42} + a^{93}x^{41} + a^{84}x^{40} + a^{15}x^{39} + a^{84}x^{38} + a^{253}x^{37} + a^{173}x^{36} + a^{160}x^{35} + \\  &a^{89}x^{34} + a^{227}x^{33} + a^{52}x^{32} + a^{199}x^{31} + a^{97}x^{30} + a^{95}x^{29} + a^{231}x^{28} + a^{52}x^{27} + \\  &a^{177}x^{26} + a^{41}x^{25} + a^{125}x^{24} + a^{137}x^{23} + a^{241}x^{22} + a^{166}x^{21} + a^{225}x^{20} + a^{118}x^{19} + \\  &a^2x^{18} + a^{54}x^{17} \\  &+ a^{32}x^{16} + a^{82}x^{15} + a^{215}x^{14} + a^{175}x^{13} + a^{198}x^{12} + a^{43}x^{11} + a^{238}x^{10} + a^{235}x^9 + \\  &a^{27}x^8 + a^{101}x^7 + a^{184}x^6 + a^{127}x^5 + a^3x^4 + a^5x^3 + a^8x^2 + a^{163}x + a^{238}  \end{aligned}  $

## 附属書 B (規定) 誤り訂正復号手順

1-M 型シンボルを例に規定する。このシンボルでは GF (2<sup>8</sup>) 上の (26, 16, 4) リードソロモン符号を誤り訂正に使用する。シンボルからマスク処理を解除した後のコード語が

$$R = (r_0, r_1, r_2, \dots, r_{25})$$

すなわち,

$$R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{25}x^{25}$$

であったとする。

ここに,  $r_i$  ( $i=0\sim 25$ ) は, GF (2<sup>8</sup>) の元とする。

a) シンドロームを計算する。

シンドローム  $S_i$  ( $i=0\sim 7$ ) を求める。

$$S_0 = R(1) = r_0 + r_1 + r_2 + \dots + r_{25}$$

$$S_1 = R(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{25}\alpha^{25}$$

.

$$S_7 = R(\alpha^7) = r_0 + r_1\alpha^7 + r_2\alpha^{14} + \dots + r_{25}\alpha^{175}$$

ここに,  $\alpha$  は, GF (2<sup>8</sup>) の原始元とする。

b) 誤り位置を求める。

$$S_0\sigma_4 - S_1\sigma_3 + S_2\sigma_2 - S_3\sigma_1 + S_4 = 0$$

$$S_1\sigma_4 - S_2\sigma_3 + S_3\sigma_2 - S_4\sigma_1 + S_5 = 0$$

$$S_2\sigma_4 - S_3\sigma_3 + S_4\sigma_2 - S_5\sigma_1 + S_6 = 0$$

$$S_3\sigma_4 - S_4\sigma_3 + S_5\sigma_2 - S_6\sigma_1 + S_7 = 0$$

上の式を使ってそれぞれの誤り位置変数  $\sigma_i$  ( $i=1\sim 4$ ) を求める。

さらに, 次の多項式にその変数を代入し, GF (2<sup>8</sup>) の元を順次代入する。

$$\sigma(x) = \sigma_4 + \sigma_3x + \sigma_2x^2 + \sigma_1x^3 + x^4$$

この結果,  $\sigma(\alpha_j) = 0$  となる元  $\alpha_j$  に対し,  $j$  けた目 (0 けた目から数えて) に誤りがあることがわかる。

c) 誤りの大きさを求める。

b) において  $j^1, j^2, j^3, j^4$  けた目に誤りがあるとし, 次にその誤りの大きさを求める。

$$Y_1\alpha_j^1 + Y_2\alpha_j^2 + Y_3\alpha_j^3 + Y_4\alpha_j^4 = S_0$$

$$Y_1\alpha_j^2 + Y_2\alpha_j^4 + Y_3\alpha_j^6 + Y_4\alpha_j^8 = S_1$$

$$Y_1\alpha_j^3 + Y_2\alpha_j^6 + Y_3\alpha_j^9 + Y_4\alpha_j^{12} = S_2$$

$$Y_1\alpha_j^4 + Y_2\alpha_j^8 + Y_3\alpha_j^{12} + Y_4\alpha_j^{16} = S_3$$

各誤りの大きさ  $Y_i$  ( $i=1\sim 4$ ) を求めるために上の式を解く。

d) 誤りを訂正

各誤り位置に誤りの大きさの値の補数を加算し, 誤りを訂正する。

## 附属書 C (規定) 形式情報

形式情報は、5 ビットのデータ及び 10 ビットの BCH 誤り訂正からなる 15 ビット列で構成する。この附属書では、誤り訂正ビットの計算及び誤り訂正復号手順について規定する。

**C.1 誤り訂正ビット計算** Bose-Chaudhuri-Hocquenghem (15, 5) 符号を誤り訂正として使用する。データビット文字列を係数とする多項式を生成多項式  $G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$  で除算する。剰余多項式の係数文字列をデータビット文字列に付加し、(15, 5) BCH 符号文字列を形成する。最後に、どのようなマスクパターン及び誤り訂正レベルの組合せでも形式情報のビットパターンがすべてゼロにならないように、また、モデル 1 シンボルがモデル 2 シンボルと自動識別できるように、ビット文字列を 101010000010010 と XOR 演算し、マスク処理をする。

### 例

誤り訂正レベル M ; マスクパターン 101  
 2 進文字列: 00101  
 多項式:  $x^2 + 1$   
 (15-5) 乗する:  $x^{12} + x^{10}$   
 G(x) で除算する:  $= (x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1) x^2 + (x^7 + x^6 + x^4 + x^3 + x^2)$   
 上の多項式の係数文字列を形式情報のデータ文字列に付加する。  
 00101 + 0011011100 → 001010011011100  
 マスクと XOR 演算 101010000010010  
 結果: 100000011001110

本体の 8.9 に規定する方法で、これらのビットを形式情報領域に配置する。

**C.2 誤り訂正復号手順** マスクパターン 101010000010010 でビット列を XOR 演算し、形式情報のマスク処理を解除する。

これによって次のコード語が得られる。

$$R = (r_0, r_1, r_2, \dots, r_{14})$$

すなわち、

$$R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{14}x^{14}$$

ここに、 $r_i$  ( $i=0 \sim 14$ ) は、0 又は 1 とする。

シンδροームを計算する。

シンδροーム  $S_i$  ( $i=1, 3, 5$ ) を求める。

$$S_1 = R(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{14}\alpha^{14}$$

$$S_3 = R(\alpha^3) = r_0 + r_1\alpha^3 + r_2\alpha^6 + \dots + r_{14}\alpha^{42}$$

$$S_5 = R(\alpha^5) = r_0 + r_1\alpha^5 + r_2\alpha^{10} + \dots + r_{14}\alpha^{70}$$

ここに、 $\alpha$  は、GF ( $2^4$ ) の原始元とする。

誤り位置を求める。

$$S_1 + \sigma_1 = 0$$

$$S_3 + S_2\sigma_1 + S_1\sigma_2 + \sigma_3 = 0$$

$$S_5 + S_4\sigma_1 + S_3\sigma_2 + S_2\sigma_3 = 0$$

ここに,  $S_2 = (S_1)^2$ ,  $S_4 = (S_2)^2$  とする。

上の式を使ってそれぞれの誤り位置変数  $\sigma_i$  ( $i=1\sim 3$ ) を求める。

さらに, 次の多項式にその変数を代入し, GF ( $2^4$ ) の元を順次代入する。

$$\sigma(x) = x^3 + \sigma_1x^2 + \sigma_2x + \sigma^3$$

この結果,  $\sigma(\alpha_j) = 0$  となる元  $\alpha_j$  に対し,  $j$  けた目 (0 けた目から数えて) に誤りがあることがわかる。各誤り位置のビット値を反転し, 誤りを訂正する。

## 附属書 D (規定) 型番情報

型番情報は、6 ビットのデータ及び 12 ビットの拡張 BCH 誤り訂正からなる 18 ビット列で構成する。この附属書では誤り訂正ビットの計算及び誤り訂正復号手順について規定する。

**D.1 誤り訂正ビット計算** 拡張 Bose-Chaudhuri-Hocquenghem (18, 6) 符号を誤り訂正として使用する。データビット文字列を係数とする多項式を生成多項式  $G(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$  で除算する。剰余多項式の係数文字列をデータビット文字列に付加し、(18, 6) 拡張 BCH 符号文字列を形成する。

### 例

型番：	7
2 進文字列：	000111
多項式：	$x^2 + x + 1$
(18-6) 乗する：	$x^{14} + x^{13} + x^{12}$
$G(x)$ で除算する：	$= (x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1)x^2 + (x^{11} + x^{10} + x^7 + x^4 + x^2)$

上の多項式の係数文字列を型番情報のデータ文字列に付加する。

000111 + 110010010100 → 000111110010010100

本体の 8.10 で規定されているように、これらのビットを型番情報領域に配置する。

各型番に対する全型番情報のビット列を附属書 D 表 1 に示す。

附属書 D 表 1 各型番における型番情報のビット例

型番	型番情報のビット列	16進表記
7	00 0111 1100 1001 0100	07C94
8	00 1000 0101 1011 1100	085BC
9	00 1001 1010 1001 1001	09A99
10	00 1010 0100 1101 0011	0A4D3
11	00 1011 1011 1111 0110	0BBF6
12	00 1100 0111 0110 0010	0C762
13	00 1101 1000 0100 0111	0D847
14	00 1110 0110 0000 1101	0E60D
15	00 1111 1001 0010 1000	0F928
16	01 0000 1011 0111 1000	10B78
17	01 0001 0100 0101 1101	1145D
18	01 0010 1010 0001 0111	12A17
19	01 0011 0101 0011 0010	13532
20	01 0100 1001 1010 0110	149A6
21	01 0101 0110 1000 0011	15683
22	01 0110 1000 1100 1001	168C9
23	01 0111 0111 1110 1100	177EC
24	01 1000 1110 1100 0100	18EC4
25	01 1001 0001 1110 0001	191E1
26	01 1010 1111 1010 1011	1AFAB
27	01 1011 0000 1000 1110	1B08E
28	01 1100 1100 0001 1010	1CC1A
29	01 1101 0011 0011 1111	1D33F
30	01 1110 1101 0111 0101	1ED75
31	01 1111 0010 0101 0000	1F250
32	10 0000 1001 1101 0101	209D5
33	10 0001 0110 1111 0000	216F0
34	10 0010 1000 1011 1010	228BA
35	10 0011 0111 1001 1111	2379F
36	10 0100 1011 0000 1011	24B0B
37	10 0101 0100 0010 1110	2542E
38	10 0110 1010 0110 0100	26A64
39	10 0111 0101 0100 0001	27541
40	10 1000 1100 0110 1001	28C69



## 附属書 E (規定) 位置合せパターンの位置

位置合せパターンは、シンボルの左上隅～右下隅への対角線の両側で対称的に配置される。それらはタイミングパターンとその反対側との間でできるだけ均等間隔となる。不均等となる間隔は、タイミングパターンと最初の位置合せパターンとの間で調整する。

各型番の位置合せパターンの数及び各位置合せパターンの中心モジュールの行座標又は列座標を**附属書 E 表 1**に示す。

附属書 E 表 1 位置合せパターンの中心の行座標又は列座標

型番	位置合せパターンの数	中心モジュールの行座標又は列座標						
1	0	—						
2	1	6	18					
3	1	6	22					
4	1	6	26					
5	1	6	30					
6	1	6	34					
7	6	6	22	38				
8	6	6	24	42				
9	6	6	26	46				
10	6	6	28	50				
11	6	6	30	54				
12	6	6	32	58				
13	6	6	34	62				
14	13	6	26	46	66			
15	13	6	26	48	70			
16	13	6	26	50	74			
17	13	6	30	54	78			
18	13	6	30	56	82			
19	13	6	30	58	86			
20	13	6	34	62	90			
21	22	6	28	50	72	94		
22	22	6	26	50	74	98		
23	22	6	30	54	78	102		
24	22	6	28	54	80	106		
25	22	6	32	58	84	110		
26	22	6	30	58	86	114		
27	22	6	34	62	90	118		
28	33	6	26	50	74	98	122	
29	33	6	30	54	78	102	126	
30	33	6	26	52	78	104	130	
31	33	6	30	56	82	108	134	
32	33	6	34	60	86	112	138	
33	33	6	30	58	86	114	142	
34	33	6	34	62	90	118	146	
35	46	6	30	54	78	102	126	150
36	46	6	24	50	76	102	128	154
37	46	6	28	54	80	106	132	158
38	46	6	32	58	84	110	136	162
39	46	6	26	54	82	110	138	166
40	46	6	30	58	86	114	142	170

備考 例えば、7型は、表で6, 22, 及び38という値である。したがって、位置合せパターンは(行, 列)すなわち座標(6, 22), (22, 6), (22, 22), (22, 38), (38, 22)及び(38, 38)を中心とする。座標(6, 6), (6, 38)及び(38, 6)は、位置検出要素パターンで占められるため、位置合せパターンとしては使用しない。

## 附属書 F (規定) データキャリア識別子

QR コードに割り当てられているデータキャリア識別子 (適切にプログラムされた復号器によって、復号されたデータの頭に付加される。) を附属書 F 表 1 に示す。

$JQm$

ここに、  
**J**: データキャリア識別子フラグ (JIS X 0201 値 93)。  
**Q**: QR コードシンボルのシンボル識別子。  
**m**: 附属書 F 表 1 で定義するいずれかの値をもつ変更子。

附属書 F 表 1 データキャリア識別子任意機能及び変更子値

変更子値	任意機能
0	モデル 1 シンボル
1	モデル 2 シンボル, ECI プロトコル未適用
2	モデル 2 シンボル, ECI プロトコル適用
3	モデル 2 シンボル, ECI プロトコル未適用, FNC1 適用 (1 番目の位置)
4	モデル 2 シンボル, ECI プロトコル適用, FNC1 適用 (1 番目の位置)
5	モデル 2 シンボル, ECI プロトコル未適用, FNC1 適用 (2 番目の位置)
6	モデル 2 シンボル, ECI プロトコル適用, FNC1 適用 (2 番目の位置)

備考1.  $m$  の許容値は, 0, 1, 2, 3, 4, 5, 6 とする。

2. モデル 1 シンボルについての変更子値は, 0 だけとする。

## 附属書 G (参考) シンボルの符号化例

**序文** この附属書は、本体及び附属書 (規定) に関連する事柄を補足するもので、規定の一部ではない。

この附属書では、**本体の 8.4.2** に従う数字モードで、データ文字列 01234567 の 1-M 型シンボルへの符号化を記述する。

### 手順 1 データの符号化

3 けたのグループに分割し、各グループを 10 又は 7 ビットの 2 進値に変換する。

012 = 0000001100

345 = 0101011001

67 = 1000011

文字数指示子を 2 進 (1-M 型は 10 ビット) に変換する。

文字数指示子 (8) = 0000001000

数字モード (0001) に対するモード指示子, 文字数指示子, 2 進データ及び終端パターン (0000) を接続する。

0001 0000001000 0000001100 0101011001 1000011 0000

8 ビットコード語に分割し、必要な埋め草ビット (下線で示す。) を付加する。

00010000 00100000 00001100 01010110 01100001 10000000

シンボルのデータコード語容量 (1-M 型は 16 データコード語であるので、10 埋め草コード語が必要) を満たすために埋め草コード語 (下線で示す。) を付加する。結果は、次のとおりとなる。

00010000 00100000 00001100 01010110 01100001 10000000 11101100 00010001 11101100  
00010001 11101100 00010001 11101100 00010001 11101100 00010001

### 手順 2 誤り訂正コード語の生成

リードソロモンのアルゴリズムを使用し、必要数の誤り訂正コード語 (1-M 型は 10 個必要) を生成する。これら (下線で示す。) は、ビット列に付加される。結果は次のとおりとなる。

00010000 00100000 00001100 01010110 01100001 10000000 11101100 00010001 11101100  
 00010001 11101100 00010001 11101100 00010001 11101100 00010001 10100101 00100100  
11010100 11000001 11101101 00110110 11000111 10000111 00101100 01010101

### 手順 3 マトリックスにモジュールを配置

1-M 型シンボルでは一つのリードソロモンブロックしか存在しないので、この例ではインタリーブ配置は、必要ではない。21×21 の空マトリックス内に位置検出要素パターン及びタイミングパターンを配置し、形式情報のモジュール位置は、一時的に空とする。手順 2 で得られたコード語を**本体の 8.7.3** に従い、マトリックス内に配置する。



附属書 G 図 1 モジュール配置後のシンボル

#### 手順 4 マスク処理パターンの選択

本体の 8.8.1 で定義されているマスク処理パターンを順次、適用し、本体の 8.8.2 に従い、結果を評価する。選択されたマスク処理パターンは参照 010 となる。

#### 手順 5 形式情報

誤り訂正レベルは M であり、マスク処理パターンは 010 となる。したがって、形式情報のデータビットは、00 010 となる。

BCH 誤り訂正の計算によって、データに付加されるビット列として 1001101110 が生成され、マスク前の形式情報として 000101001101110 を得る。

マスク 101010000010010 で上のビット列を XOR 演算する。

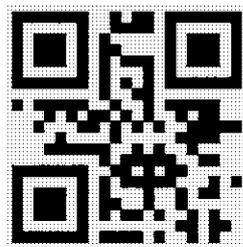
000101001101110 (マスク前のビット列)

101010000010010 (マスク)

101111001111100 (シンボルに配置される形式情報)

#### 手順 6 最終的なシンボル構築

本体の 8.8 で規定するようにシンボルの符号化領域に選択されたマスク処理パターンを適用し、手順 3 で保留した位置に形式情報モジュールを付加する。



附属書 G 図 2 最終的なシンボル

## 附属書 H (参考) ビット列の長さの最適化

**序文** この附属書は、本体及び附属書（規定）に関連する事柄を補足するもので、規定の一部ではない。

本体で規定するとおり、QR コードは様々な符号化モードがあり、その各モードは与えられたデータ列を表すのに必要なビット数がそれぞれ異なる。各モードの文字集合間では重複があり、例えば、数字データは数字、英数字及び 8 ビットバイトのモードで、そしてラテン英数字データは英数字及び 8 ビットバイトのモードで符号化できる。したがって、シンボル生成ソフトウェアは、複数のモードで表されるデータ文字を符号化する場合、最も適切なモードを選択する必要がある。

この選択は最初に行わなければならないが、データ列の途中においても可能な場合がある。

ビット列長を最短にするために幾つかの他の手法を適用する。アルゴリズムは、モード変更時に要求される付加情報を考慮し、直後の文字列だけでなく、次のデータ列も考慮する必要がある。

本体の 8.4.2～8.4.5 で与えられる圧縮効率を注意深く解釈する必要がある。与えられたデータ集合に対する最良の方法は、データごとに最小ビットとなる方法をとることではない。最高次の圧縮が要求される場合、モード変更に必要な追加ビット（モード指示子及び文字数指示子）が考慮されなければならない。また、コード語数が最小でも、シンボルを満たすためにコード語列は、拡張される場合がある。この拡張には、埋め草キャラクタを使用する。

次の手引きは、任意の与えられた入力データに対する最小のビット列を決定する一つのアルゴリズムの基本を形成するものとする。括弧で示される数字、例えば、[5, 7, 9] の場合、第 1 要素“5”は 1 型～9 型、第 2 要素“7”は 10 型～26 型、第 3 要素“9”は 27 型～40 型に適用される文字数を示す。この手引きにおいて、“排他的部分集合”は、あるモードの文字集合において別のモードのより限定された文字集合と共用されない文字集合を参照する。例えば、8 ビットバイト文字集合の排他的部分集合は、16 進数 20, 24, 25, 2A, 2B, 2D～3A, 及び 41～5A を除く 8 ビット符号値 00<sub>HEX</sub>～FF<sub>HEX</sub> で構成される。また、英数字の排他的部分集合は、{A～Z, space, \$, %, \*, +, -, ., /, :} とする。

### H.1 初期モードの選択

- a) 初期入力データが漢字文字集合の場合、漢字モードを選択する。
- b) 初期入力データが 8 ビットバイト文字の排他的部分集合の場合、8 ビットバイトモードを選択する。
- c) 初期入力データが英数字の排他的文字集合であり、それらが [6, 7, 8] より少ない文字数でかつ 8 ビットバイト文字集合が後続する場合は 8 ビットバイトモードを選択、その他の場合は英数字モードを選択する。
- d) 初期データが数字で、かつ、それらが [4, 4, 5] より少ない文字数で 8 ビットバイトの排他的サブセットのデータが後続する場合、8 ビットバイトモードを選択、[7, 8, 9] より少ない文字数で英数字の排他的部分集合が後続する場合、英数字モードを選択、その他の場合は数字モードを選択する。

### H.2 ビットバイトモード中において

- a) 一つ以上の漢字が発生する場合、漢字モードに変更する。
- b) 8 ビットバイト文字集合の排他的部分集合のデータの前に少なくとも [6, 8, 9] 個の数字の列が発生

する場合、数字モードへ変更する。

- c) 8 ビットバイト文字集合の排他的部分集合のデータの前に少なくとも [11, 15, 16] 個の英数字の排他的部分集合の列が発生する場合、英数字モードへ変更する。

### H.3 英数字モード中において

- a) 一つ以上の漢字が発生する場合、漢字モードに変更する。
- b) 一つ以上の 8 ビットバイト文字集合の排他的部分集合が発生する場合、8 ビットバイトモードへ変更する。
- c) 英数字文字集合の排他的部分集合のデータの前に少なくとも [13, 15, 17] 個の数字の列が発生する場合、数字モードへ変更する。

### H.4 数字モード中において

- a) 一つ以上の漢字が発生する場合、漢字モードに変更する。
- b) 一つ以上の 8 ビットバイト文字集合の排他的部分集合が発生する場合、8 ビットバイトモードへ変更する。
- c) 一つ以上の英数字集合の排他的部分集合が発生する場合、英数字モードへ変更する。

## 附属書 I (参考) QR コードシンボルの印刷 及び読取りのための利用者手引き

序文 この附属書は、本体及び附属書（規定）に関連する事柄を補足するもので、規定の一部ではない。

I.1 一般 いかなる QR コードのアプリケーションも総合的なシステムとしてとらえなければならない。シンボル体系の符号化・復号を構成する要素（マーカ又は印字装置、ラベル、読取り装置）が 1 個のシステムとして機能することが重要である。いずれかの要素が動作不良を起こしたり、要素間の相性が悪いと、システム全体の性能低下を招くことになる。

仕様を遵守することはシステム全体の成功を確実にする一つのかぎとなるが、それ以外にも性能に影響する要因が幾つか存在する。次の手引きは、バーコードシステム又はマトリックスコードシステムの仕様を定めたり、機器の選定をする際に考慮すべき点を示している。

- a) 使用するマーキング技術又は印刷技術によって、許容範囲内の印字が可能である印刷密度を選択する。モジュール寸法が確実に印刷ヘッドピクセル寸法（印刷方向に対して水平方向及び垂直方向の両方）の整数倍となるようにする。個々の暗（又は明）モジュールのビットマップは、全体的に均衡となる寸法で調整されるが、モジュールの中心間隔が一定となることを確実にするために、ピクセルの整数相当を個々又は隣接する暗モジュールのグループにおける明暗境界上で明を暗（又は暗を明）へ変更することで印刷による増加（又は減少）の任意の調整が実行されることを確実にする。
- b) マーキング技術又は印刷技術によって制作されるシンボル密度及び品質に適した分解能をもつ読取り装置を選択する。
- c) 印字されたシンボルの光学的な特性が読取り装置の光源又はセンサの波長に確実に適合するようにする。全般的にひずみの影響が少なくなるように、モデル 2 シンボルを指定することが望ましい。
- d) 最終ラベル状態又は包装形態において、シンボルが仕様に合致していることを検証する。ラベルのかぶり、透過、湾曲又は凹凸面いずれもシンボルの可読性に影響する。これらの設計がひずみの影響を受けにくくするために、モデル 2 シンボルを指定することが望ましい。

光沢のあるシンボル面の場合、鏡面反射の影響について考慮する必要がある。また、読取り装置は、明暗の拡散反射の変化も考慮しなければならない。読取り角度によっては、反射光の鏡面反射成分が望ましい拡散反射成分を大幅に上回ることも考えられ、その場合は、読取り能力が低下することになる。材質又は部品の表面を変えることができる場合は、つや消し又は無光沢面に変えることで鏡面反射の影響を最小限に抑えることができる。これが不可能な場合には、読み取るシンボルの照度が望ましいコントラストの構成になるように、特に注意を払う必要がある。

参考 モジュールの形状は、通常正方形が望ましいが、ダイレクトマーキングなどに用いられる印字装置では、円形などの形状となることがある。このような場合には、アプリケーションで使用する読取装置での読取評価を十分に行うことが望ましい。

**I.2 モデルの利用者選択** すべての新規アプリケーションには、モデル 2 のシンボルを推奨する。位置合せパターンがモジュールグリッドの決定又はその正確性の確保における読取り過程を大きく助け、また、40 型までのシンボルが利用できるので、大データ容量を提供することができる。モデル 1 のシンボルは、既存のアプリケーションに限定する。

**I.3 誤り訂正レベルの利用者選択** 利用者は、アプリケーションで要求される適切な誤り訂正のレベルを定義する。本体の表 14 に示すとおり、L~H の 4 レベルで誤りを検出し訂正する能力があるが、レベルを上げると与えられたメッセージの長さに対してシンボルの寸法が大きくなることがある。例えば、20-Q 型のモデル 2 シンボルの場合、485 個のデータコード語をもつが、より低レベルの誤り訂正が容認される場合は、この同じデータを 15-L 型シンボルで表すこともできる（正確には 523 個のデータコード語容量）。誤り訂正レベルは、次の関係において決定する。

- a) 予想されるシンボルの品質：予想される品質レベルが低ければ、より高いレベルを採用する。
- b) 初回読取り率の高さ。
- c) 読取りができなかった場合に再読取りの機会があるか。
- d) シンボルの印刷領域の制限によって、使用できる誤り訂正レベルはどのレベルまでか。

誤り訂正レベル L は、高いシンボル品質及び／又は与えられたデータに対してシンボルをできるだけ小さくする必要のある場合に適している。レベル M は、“標準”レベルで、サイズの縮小化及び信頼性向上の両方を実現する。レベル Q は、“高い信頼性”をもつレベルで、より厳しい読取り又は印刷品質が低いアプリケーションに適している。レベル H は、実現し得る最高の信頼性を提供する。

## 附属書 J (参考) 自動識別能力

**序文** この附属書は、本体及び附属書（規定）に関連する事柄を補足するもので、規定の一部ではない。

QR コードは、他のシンボル体系と自動識別するように設計された復号器を適切にプログラムすることで読み込むことができる。適切にプログラムされた QR コード読取り装置は、別のシンボル体系のシンボルを、正当な QR コードとして復号することはない。しかし、QR コードを含む任意のマトリックスシンボル内に短いバーコードシンボルの表現が見つかることもある。

QR コードのモデル 1 シンボル及びモデル 2 シンボルも、互いに自動識別することを可能とする。

読取りの正確性を最大限にするために、復号器の有効シンボル体系セットを、使用するアプリケーションで必要とされるシンボル体系に限定することが望ましい。

## 附属書 K (規定) マトリックスコードの印刷品質の手引き

この附属書は、どのマトリックス式シンボル体系にも適用することができる、二次元マトリックスコードシンボルの印刷品質評価の枠組みを示す。本体の 11. では、QR コードへの適用について規定している。ここに示す方法は、バーコードシンボル品質の評価に用いられる JIS X 0520 と多くの点で類似している。この方法は、制御された照明及び観察条件の下で、シンボルの高解像度のグレースケール画像を得ることから始める。次に得られた画像は、復号、シンボルコントラスト、印刷時の太り、細り、軸の非均一性及び未使用誤り訂正のパラメタについて解析される。これらの 5 個のパラメタに対する等級、与えられたシンボル体系及びアプリケーションで規定される等級の中で、一番低い等級値がシンボルの最終評価となる。

ここで示す手順は、参照復号アルゴリズム及び関連するシンボル体系の仕様における他の測定項目によって必然的に増える。また、シンボル体系又はアプリケーション仕様の統括によって、適切に変更又は無効にしてもよい。

**K.1 試験画像の取得** シンボルの試験画像は、そのシンボルに対する典型的な読取り環境を模倣する状況において得られるものでなければならないが、これは、より高い解像度、均一照明及び最適な焦点という条件を伴う。特殊なアプリケーションは、要求される画像の解像度とともにシンボルへの照明の色及び角度を明確に指定しなければならないが、次に示す一般的な試験設定は、多くの一般的なアプリケーションに合うようにするのがよい。

標準モノクロビデオカメラで、試験シンボルの中心でかつ、その平面に垂直な軸上に直接、試験シンボルの画像を得る。適切な焦点で、かつ、光学的ひずみを最小限にするために、十分に小さな視野で、シンボル全体（要求されるクワイエットゾーンを含む。）をフレームに入れるのに適切なレンズを使用する。光照明は、少なくとも二つの側面から 45 度の入射角でシンボル領域を均一に照らすことが必要となる。試験画像は、標準フレーム取込み機器によって 8 ビットのグレースケールで取り入れられることが可能であり、グレースケールは、既知の拡散反射の目標物を用いて測定する。

正確な光学設定にかかわらず、二つの原則がその選択に関係する。第 1 に、試験画像のグレースケールは、直線状で、いかなる場合にもコントラストの向上又は外観の改善に対して調整されない。第 2 は、画像の解像度が安定した読取りに適切とすることで、一般にはモジュール幅及び高さが少なくとも 5 個の画像ピクセルをもつことを要求する。

### K.2 シンボルパラメタの評価

**K.2.1 復号** シンボル体系の参照復号アルゴリズムを試験画像に適用する。有効な復号が成立すれば等級は A (4.0)、成立しなければ F (0.0) とする。

復号パラメタは、適切な画像が取得されたときに、シンボルが正しく読み取れる特徴をもっているかどうかについて、合格又は不合格を基準に試験する。さらに、参照復号の初期段階では、後続する他のシンボル品質パラメタの測定に必要な三つのタスクを追加して行う。第 1 のタスクは、画像内で試験シンボルによって網羅される領域の位置を定め、定義する。第 2 は、モジュールの中心をサンプリングするためにそのグリッドマップを作成する。第 3 は、シンボルの損傷が誤り訂正機能を使用する場合、誤り訂正及び検出を実行する。こうした画像、画像座標及び誤り訂正は、次の一つ以上の測定を容易にする。

**K.2.2 シンボルコントラスト** グレースケールの画像において、要求されるクワイエットゾーンの範囲にまで広がる試験シンボルの領域内に入るすべての画像ピクセル値を反射率によって分類し、ピクセルの最も暗い部分 10 %及び最も明るい部分 10 %を抽出する。ピクセルの最も暗い部分 10 %と最も明るい部分 10 %との反射率の平均を計算する。両者の平均値の違いをシンボルのコントラスト (SC) とする。

シンボルコントラストの等級は次のように定義する。

SC $\geq$ 70 %の場合, A (4.0)

SC $\geq$ 55 %の場合, B (3.0)

SC $\geq$ 40 %の場合, C (2.0)

SC $\geq$ 20 %の場合, D (1.0)

SC $<$ 20 %の場合, F (0.0)

シンボルコントラストは、シンボル内の二つの反射状態、つまり、明及び暗がシンボル全体を通して十分に、かつ、一定して識別できるかを試験する。

**K.2.3 モジュールの伸縮** K.2.2 で計算される明暗間の平均の参照反射率しきい値を計算する。しきい値を用いて、明及び暗の領域を区別する 2 次的な 2 値化画像を作る。

モジュールの伸縮のパラメタ (明部又は暗部がそのモジュール境界を占める範囲) は、読取り性能に影響する印刷過程の品質の重要な指針である。公称寸法に対するモジュールの伸縮を最もよく表す特有のグラフィック構造は、シンボル体系によって大きく異なり、その仕様書において定義されるが、通常、固定構造又は寸法  $D$  が 2 値化画像内のピクセル数によって決定される独立モジュールのいずれかで構成する。複数の寸法 (例えば、水平及び垂直双方への伸縮) は、それぞれ個別に規定し、検査する。

検査した各寸法は、公称値  $D_{\text{NOM}}$  並びに許容される最大値  $D_{\text{MAX}}$  及び最小値  $D_{\text{MIN}}$  で規定する。測定した  $D$  は、それぞれに対応する公称値及び限界値で評価する。

$D > D_{\text{NOM}}$  の場合,  $D' = (D - D_{\text{NOM}}) / (D_{\text{MAX}} - D_{\text{NOM}})$

それ以外の場合,  $D' = (D - D_{\text{NOM}}) / (D_{\text{NOM}} - D_{\text{MIN}})$

したがって、モジュールの伸縮は、次のように各等級に分ける。

$-0.50 \leq D' \leq 0.50$  の場合, A (4.0)

$-0.70 \leq D' \leq 0.70$  の場合, B (3.0)

$-0.85 \leq D' \leq 0.85$  の場合, C (2.0)

$-1.00 \leq D' \leq 1.00$  の場合, D (1.0)

$D' < -1.00$  又は  $D' > 1.00$  の場合, F (0.0)

モジュールの伸縮は、シンボルを構成するグラフィック特性が、公称値から伸縮 (試験条件より劣るが最適な画像条件での読取り性能を妨げるような) していないことを試験する。

**K.2.4 軸の非均一性** 二次元マトリックスコードシンボルは、多角形グリッド上に配置されたモジュールからなるデータフィールドを含み、いかなる参照復号アルゴリズムもそれらのモジュールの中心位置を適切にとらえてデータを引き出さなければならない。軸の非均一性は、グリッドの各主要軸方向におけるマッピングの中心 (つまり、標本化点) 間隔を測定し、等級に分ける。

隣接する標本化点の間隔は、各多角形軸に対して個々に分類し、各軸に沿った平均間隔  $X_{\text{AVG}}$  を計算する。軸の非均一性とは、軸間で標本化点間隔がどれだけ異なるかを測定することである。つまり、

$$AN = \text{abs} (X_{\text{AVG}} - Y_{\text{AVG}}) / ((X_{\text{AVG}} + Y_{\text{AVG}}) / 2)$$

ここに、 $\text{abs} ( )$  は、絶対値を与える関数。

シンボル体系が 2 個以上の主要軸をもつ場合は、その中で最も大きく異なる二つの平均間隔値を用いて

AN を算出する。

軸の非均一性は、次のとおりに等級分けする。

AN ≤ 0.06 の場合、A (4.0)

AN ≤ 0.08 の場合、B (3.0)

AN ≤ 0.10 の場合、C (2.0)

AN ≤ 0.12 の場合、D (1.0)

AN > 0.12 の場合、F (0.0)

軸の非均一性は、正常でない視野角度で、読取り性能を妨げるシンボルの非均一的な度合いを検査する。

**K.2.5 未使用誤り訂正** リードソロモンを使用した復号の訂正能力は、次の式で表す。

$$e + 2t \leq d - p$$

ここに、  
 $e$  : 棄却誤り数  
 $t$  : 代入誤り数  
 $d$  : 誤り訂正コード語数  
 $p$  : 誤り検出のために予約されたコード語数

$d$  及び  $p$  の値は、シンボル体系仕様（シンボルサイズに依存するが）によって決定し、 $e$  及び  $t$  の値は参照復号中に決定する。未使用の誤り訂正量は、次のように計算する。

$$UEC = 1.0 - (e + 2t) / (d - p)$$

複数（例えば、インタリーブ）のリードソロモンブロックをもつシンボル体系の場合、UEC は各ブロックについてそれぞれ計算され、各ブロックの等級（次に示す）の中で、一番低い等級値をシンボルの等級とする。

UEC ≥ 0.62 の場合、A (4.0)

UEC ≥ 0.50 の場合、B (3.0)

UEC ≥ 0.37 の場合、C (2.0)

UEC ≥ 0.25 の場合、D (1.0)

UEC < 0.25 の場合、F (0.0)

未使用誤り訂正パラメタは、シンボルの部分的又はスポット的な損傷が、誤り訂正機能による読取り安全性のクワイエットゾーンをどれだけ侵すかを試験する。

**K.3 シンボル等級の総合評価** 上で得られたパラメタの等級の中で一番低い値をシンボルの総合的な等級とする。附属書 K 表 1 に試験パラメタ及び等級の一覧を示す。

附属書 K 表 1 マトリックスコード記号の品質パラメタの一覧

等級	参照復号	シンボル コントラスト	モジュールの伸縮	軸の非均一性	未使用誤り訂正
A (4.0)	合格	SC ≥ 0.70	-0.50 ≤ D' ≤ 0.50	AN ≤ 0.06	UEC ≥ 0.62
B (3.0)		SC ≥ 0.55	-0.70 ≤ D' ≤ 0.70	AN ≤ 0.08	UEC ≥ 0.50
C (2.0)		SC ≥ 0.40	-0.85 ≤ D' ≤ 0.85	AN ≤ 0.10	UEC ≥ 0.37
D (1.0)		SC ≥ 0.20	-1.00 ≤ D' ≤ 1.00	AN ≤ 0.12	UEC ≥ 0.25
F (0.0)	不合格	SC < 0.20	D' < -1.00 or D' > 1.00	AN > 0.12	UEC < 0.25

品質等級は、英字又はそれに相当する数字のいずれかで表す。前者は、JIS X 0520 で参照されるように、不合格等級を F として、質の高いものから順に A~D の英字で表し、後者は、EN 1635 (Bar Coding—Test Specifications for Bar Code Symbols) に従って 4.0~0.0 の小数点以下 1 けたの数字で表す。

附属書 K 表 2 は、英字及び数字等級の相関関係を示す。

附属書 K 表 2 英字品質等級と数字品質等級との関係

$3.5 \leq A \leq 4.0$
$2.5 \leq B < 3.5$
$1.5 \leq C < 2.5$
$0.5 \leq D < 1.5$
$F < 0.5$

## 附属書 L (参考) プロセス制御技術

**序文** この附属書は、本体及び附属書 (規定) に関連する事柄を補足するもので、規定の一部ではない。

この附属書では、読取り可能な QR コードシンボルの作成プロセスを監視及び制御するために有効なツール及び手順を示す。これらの技術は、作成されるシンボルの印刷品質検査を規定するものではない (本体の 11. 及び附属書 K で定義する手法はシンボルの品質評価に要求される手法) が、これらの技術は個別に、かつ、総合的に、シンボル生成の過程で使用可能なシンボルが作成されるかどうかのよい指針となる。

**L.1 シンボルコントラスト** ほとんどの一次元バーコードシンボル用検証器は、反射計測モード、走査反射率をプロット及び/又は復号不能な走査のシンボルコントラスト (JIS X 0520 で定義される) を出力するモードのいずれかを備えている。特殊な照明構成が要求されるシンボルを除き、660 nm 波長で 0.150 mm 又は 0.250 mm の口径で得られるシンボルコントラスト (又は最大走査反射率から最小走査反射率を通して得られるシンボルコントラスト値、若しくは反射計測で得られる最大値及び最小値の差のいずれか) は、画像に起因するシンボルコントラスト値と相互関係があることが分かっている。特にこのような読取りは、シンボルコントラストが、意図するシンボル品質等級で許容される最小値を十分に超えることを検査するのに用いることができる。

**L.2 軸の非均一性の評価** どのシンボルにおいても、左上の位置検出要素パターンの左端から右上の位置検出要素パターンの右端までの距離、及び左上の位置検出要素パターンの上端から左下の位置検出要素パターンの下端までの距離を測定する。各距離をその型番のモジュール数で除算する。例えば、2 型のシンボルは、除数として 25 をもつ。附属書 K の K.2.4 の式における  $X_{AVG}$  及び  $Y_{AVG}$  の結果を代入し、軸の非均一性の評価に対する結果を等級化する。

**L.3 シンボルのひずみ及び欠陥の目視検査** サンプルシンボルの位置検出要素パターン及びタイミングパターンの目視検査中に、シンボル作成の過程の重要な様相を監視することができる。

マトリックスコードシンボルは、マトリックスグリッドの部分的なひずみを生じやすい。そのようなひずみは、どれも位置検出要素パターン上の湾曲した縁、又は位置検出要素パターン間のタイミングパターン及び位置検出要素パターン内部境界線に並行する非均一な間隔として、視覚的に現れる。

位置検出要素パターン及びそれに隣接するクワイエットゾーン領域は、常に一様に暗及び明とする。シンボル全体を通して明又は暗の線による不具合を生じる印刷機構の欠陥は、位置検出パターン又はクワイエットゾーンを横切る箇所で見られる。このような印刷の過程のシステム的な欠陥は、修正しなければならない。

**L.4 モジュールの伸縮の評価** バー及びスペースパターンの直接測定値が出力可能なバーコード検証器を、水平方向及び垂直方向の印刷の伸縮評価に用いてもよい。その場合、直交する2本の走査線（一本は上方の二つの位置検出要素パターンの3×3モジュールブロックの中央を横切り、もう一本は同様に左側の二つの位置検出要素パターンを横切る。）に沿って測定する。出力はそれぞれの走査線の終端が、明白な暗-明-暗-明-暗のパターンを示すことが望ましい。印刷の伸び（又は縮み）は、五つのエレメント測定値と、理想的な1:1:3:1:1の幅の比率とを比較することで評価できる。

## 附属書 M (規定) QR コードモデル 1 シンボルの特性

QR コードモデル 1 シンボルは、幾つかの又はクローズドシステム用途で用いられるシンボル体系の形式であるが、新たな用途、オープンシステム用途又はデータ容量が多くなりそうな用途には推奨しない。ほとんどの事項でモデル 2 と同じ仕様であるが、いくつかの重要な事項で異なる。それらの事項をこの附属書で詳細に記載する。

**M.1 QR コードモデル 1 シンボルの特徴** モデル 1 シンボルとモデル 2 シンボルとの相違は、次のとおりとする。

a) シンボルサイズ (クワイエットゾーンを除く。)

21×21 モジュール～73×73 モジュール (1 型から 14 型、型番が一つ上がるごとに一辺につき 4 モジュールずつ増加)

b) 最大データ容量 [最も低い誤り訂正レベルをもつ最大シンボルサイズ (14-L 型) に対する]

- |                |          |
|----------------|----------|
| 1) 数字データ       | 1 167 文字 |
| 2) 英数字データ      | 707 文字   |
| 3) 8 ビットバイトデータ | 486 文字   |
| 4) 漢字データ       | 299 文字   |

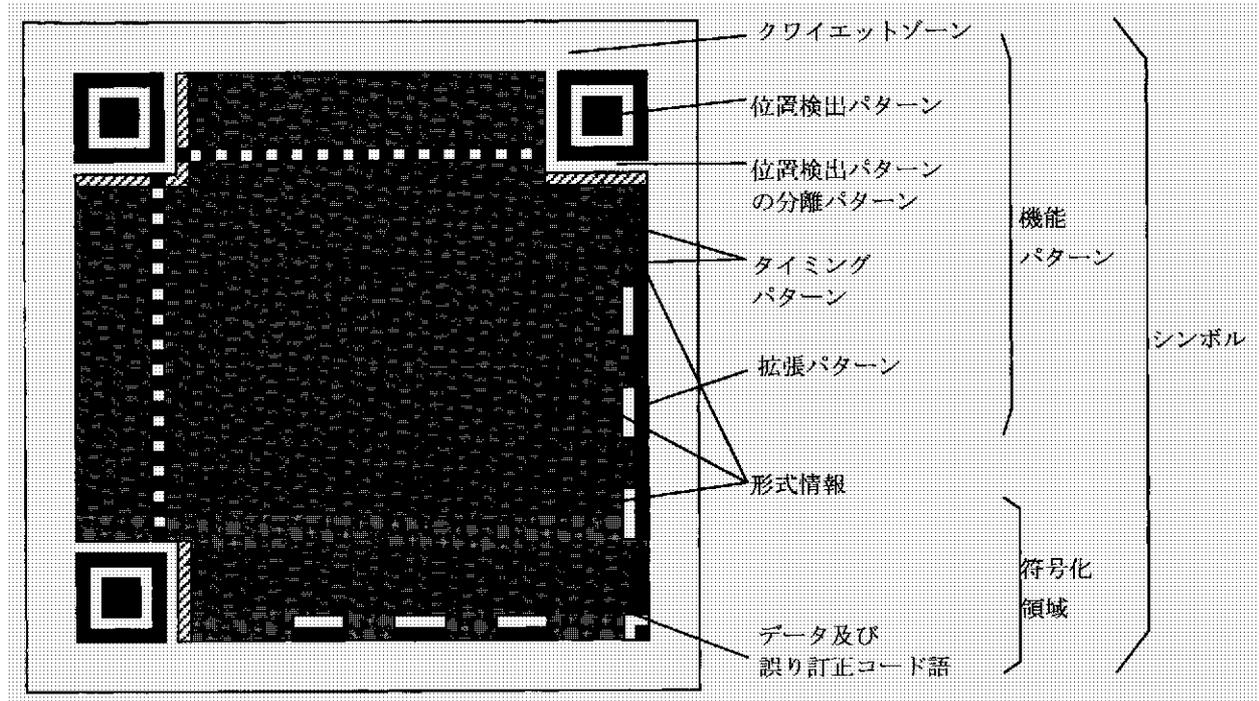
c) シンボルの構造

- |                 |                                |
|-----------------|--------------------------------|
| 1) 位置補正パターン：    | モデル 1 シンボルは、位置補正パターンをもたない。     |
| 2) 拡張パターン：      | モデル 1 シンボルは、右側及び下側に拡張パターンをもつ。  |
| 3) 型番情報：        | モデル 1 シンボルは、型番情報をもたない。         |
| 4) シンボルキャラクタの配置 | 上記の結果、シンボルキャラクタの配置は、異なった規則に従う。 |

d) 誤り訂正

誤り検出及び誤り訂正コード語はモデル 2 と同様に計算するが、幾つかの型番に対する誤り訂正ブロックの数及び大きさは異なる。

附属書 M 図 1 に QR コードモデル 1 シンボルの 7 型の構造を示す。



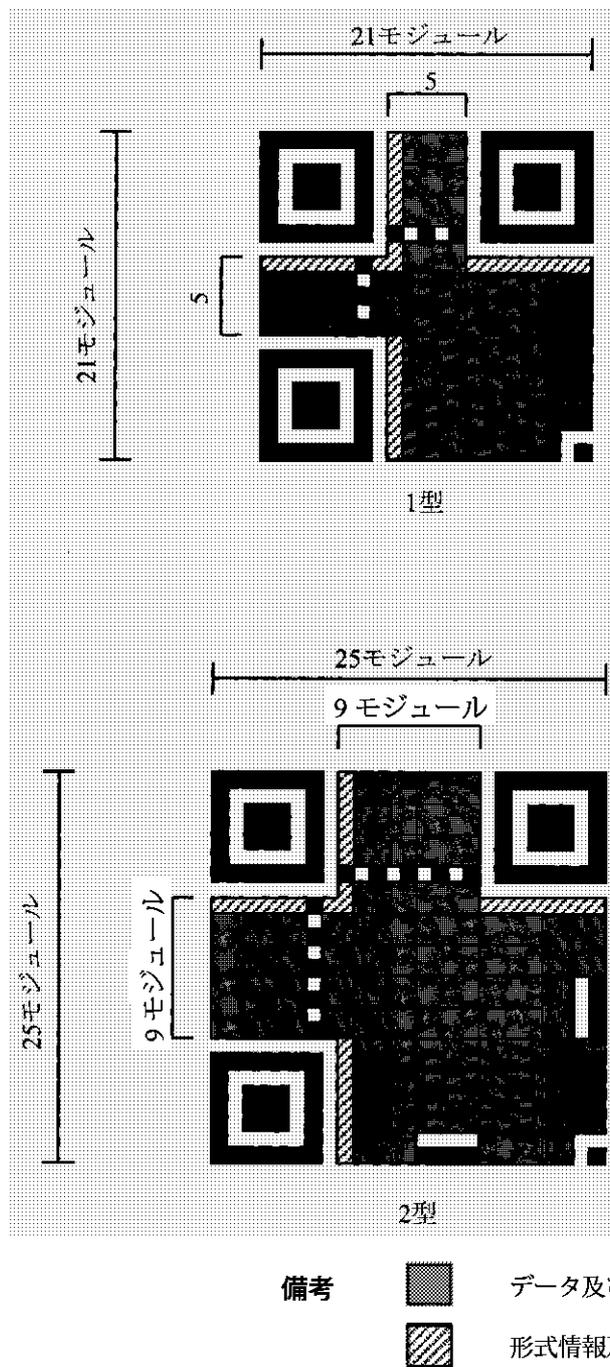
附属書 M 図 1 QR コードモデル 1 シンボルの構造

**M.2 シンボルの型番及び大きさ** QR コードモデル 1 シンボルには、1 型～14 型までの 14 種類の型番しかない。その大きさは、本体の 7.3.1 で既定のとおり、同じ型番であればモデル 2 シンボルの大きさと等しい。したがって、1 型は 21×21 モジュール、14 型は 73×73 モジュールとなる。附属書 M 図 2 及び附属書 M 図 3 は、モデル 1 の 1 型、2 型、13 型及び 14 型の構造を示す。附属書 M 表 1 に、モデル 1 シンボルの、異なる誤り訂正レベルにおけるすべての型番のデータ容量を示す。

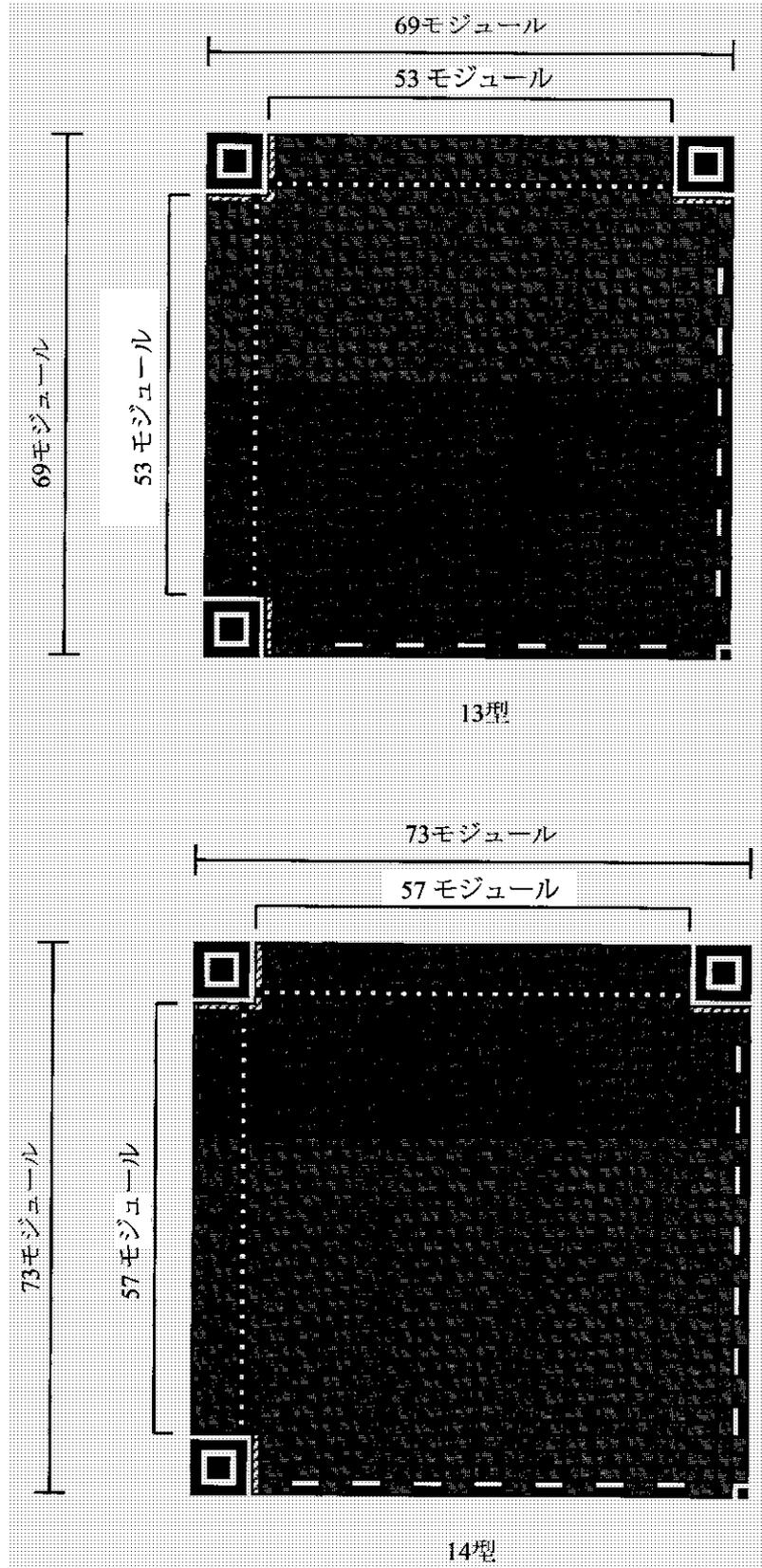
附属書 M 表 1 QR コードモデル 1 の全型番のデータ容量

型番	モジュール数/ 辺(A)	機能パターン モジュール(B)	形式情報 モジュール(C)	(C)以外のデータモジュール (D=A <sup>2</sup> -B-C)	データ容量 [コード語 <sup>(1)</sup> (E)]
1	21	206	31	204	26
2	25	230	31	364	46
3	29	238	31	572	72
4	33	262	31	796	100
5	37	270	31	1068	134
6	41	294	31	1356	170
7	45	302	31	1692	212
8	41	326	31	2044	256
9	53	334	31	2444	306
10	57	358	31	2860	358
11	61	366	31	3324	416
12	65	390	31	3804	476
13	69	398	31	4332	542
14	73	422	31	4876	610

注<sup>(1)</sup> 最初のコード語は、4 ビット長とする。それ以降のすべてのコード語は、8 ビット長とする。最初のデータコード語 (4 ビット長) は、誤り訂正コード語生成時には 0000 を前に付け、8 ビット長とする。



附属書 M 図 2 モデル 1 シンボルの 1 型及び 2 型



附属書 M 図 3 モデル 1 シンボルの 13 型及び 14 型

### M.3 シンボルの構造

**M.3.1 位置補正パターン** QRコードモデル1シンボルは、位置補正パターンをもたない。符号化領域は、形式情報と合わせて、**附属書 M 図 1** で影をほどこしたすべての範囲である。

**M.3.2 拡張パターン** 拡張パターンは、QRコード機能の将来的な拡張を意図していた。また、データは符号化しない。拡張パターンは、シンボルの右下に位置する1個の4モジュールブロックとシンボルの右辺及び下辺に沿って位置する幾つかの8モジュールブロックとで構成される。8モジュールブロック数は、シンボルの型番によって決まり、次の公式からN型に対するブロック数が得られる。

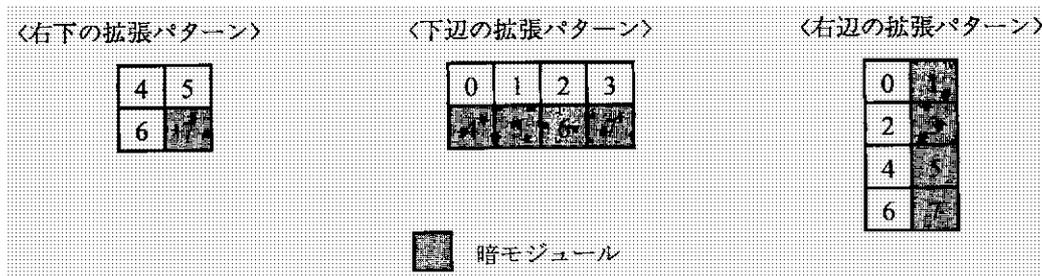
$$8 \text{ モジュール拡張ブロック数} = 2 (N \text{ DIV } 2)$$

このことは、1型のシンボルは4モジュールの拡張パターンだけをもち、2型及び3型のシンボルはこれに加えて2個の8モジュールブロックをもち、4型及び5型のシンボルは四つの拡張パターンをもつことを示す。**附属書 M 図 2** 及び**附属書 M 図 3** はモデル1の1型、2型、13型及び14型の拡張パターンの位置を示す。

**附属書 M 図 4** はシンボルの右下、右辺及び下辺に位置する拡張パターンの明並びに暗のモジュールパターンを示す。

奇数のシンボル型番では、最初の8モジュールブロックは、右辺の17行目から20行目まで及び下辺の17列目から20列目までに位置する。後続するブロックは、右辺(下辺)の25行(列)目~28行(列)目、33行(列)目~36行(列)目というように位置し、拡張パターンと拡張パターンとの間にある8モジュールブロックは符号化領域として使われる。

この原理は、13行(列)目~16行(列)目で始まり、21行(列)目~24行(列)目、29行(列)目~32行(列)目と続く偶数の型番にも適用される。



**附属書 M 図 4 拡張パターン**

**備考** QRコードの初期運用では上と異なる拡張パターンが使用されていた。それらは**附属書 M 図 4** での暗色のビットに加え、ビット0と3(下辺)及びビット0と6(右辺)が暗である。両パターンともに有効でシンボル内で情報をもたない。

**M.3.3 型番情報** このシンボルは、符号化領域に型番情報をもたない。

**M.4 モデル1シンボルの符号化** 符号化手順は、本体の8.1で規定のとおりとするが、次に従う。

手順1の拡張チャネル解釈についての言及は無視する。

手順4を次のとおり置き換える。“**手順4 最終的なメッセージの構築** 各ブロックのデータコード語を接続し、各ブロックの誤り訂正コード語、必要に応じて残余コード語を後続する。”

手順5の位置補正パターンを拡張パターンに置き換える。

**M.4.1 データからビット列への変換** モデル1におけるデータからビット列への変換は、**本体の 8.4** に記述の、モデル2に対して規定された手順に従う（**8.4.1** 及びその細分した箇条を除く。）。モデル1は、ECIプロトコルに対応していないことに注意し、データ分析の段階で、データを検証し、**8.3.2～8.3.5** で規定の、数字、英数、8ビットバイト及び漢字だけを含むことを確認しなければならない。

**M.4.2 ビット列からコード語への変換** モデル1シンボルでは、ビット列をまず、4ビットコード語に分割、残りを8ビットコード語に分割する。最後のコード語の長さが8ビットでないビット列の場合、最後のコード語に2進値0の埋め草ビットを付加し、8ビット長にする。埋め草ビットは、データ列の最終ビット(最下位けた)の後に付加する。メッセージビット列は、**8.4.9** に記述のとおり、埋め草コード語 **11101100** 及び **00010001** を交互に付加することによって、**附属書 M 表 2** 及び**附属書 M 表 3** に示すシンボルのデータ容量を満たす長さにする。

附属書 M 表 2 モデル 1 の 1 型～10 型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8 ビット バイト	漢字
1	L	19	148	40	24	17	10
	M	16	124	33	20	14	8
	Q	13	100	25	15	11	6
	H	9	68	16	10	7	4
2	L	36	284	81	49	34	20
	M	30	236	66	40	28	17
	Q	24	188	52	31	22	13
	H	16	124	33	20	14	8
3	L	57	452	131	79	55	33
	M	44	348	100	60	42	25
	Q	36	284	81	49	34	20
	H	24	188	52	31	22	13
4	L	80	636	186	113	78	48
	M	60	476	138	84	58	35
	Q	50	396	114	69	48	29
	H	34	268	76	46	32	19
5	L	108	860	253	154	106	65
	M	82	652	191	116	80	49
	Q	68	540	157	95	66	40
	H	46	364	105	63	44	27
6	L	136	1 084	321	194	134	82
	M	106	844	249	151	104	64
	Q	86	684	201	122	84	51
	H	58	460	133	81	56	34
7	L	170	1 356	402	244	168	103
	M	132	1 052	311	188	130	80
	Q	108	860	253	154	106	65
	H	72	572	167	101	70	43
8	L	208	1 660	493	299	206	126
	M	160	1 276	378	229	158	97
	Q	128	1 020	301	183	126	77
	H	87	692	203	123	85	52
9	L	246	1 964	585	354	244	150
	M	186	1 484	441	267	184	113
	Q	156	1 244	369	223	154	94
	H	102	812	239	145	100	61
10	L	290	2 316	690	418	287	177
	M	222	1 772	526	319	219	135
	Q	183	1 460	433	262	180	111
	H	124	988	291	176	121	74

注<sup>(2)</sup> 最初のコード語は 4 ビット長で、後続するすべてのコード語は 8 ビット長とする。

<sup>(3)</sup> データビット数には、モード指示子及び文字数指示子も含む。

附属書 M 表 3 モデル 1 の 11 型～14 型のデータコード語数及び入力データ容量

型番	誤り訂正 レベル	データ コード語数 <sup>(2)</sup>	データ ビット数 <sup>(3)</sup>	データ容量			
				数字	英数字	8 ビット バイト	漢字
11	L	336	2 684	800	485	333	205
	M	256	2 044	608	368	253	156
	Q	208	1 660	493	299	205	126
	H	145	1 156	342	207	142	87
12	L	384	3 068	915	555	381	234
	M	292	2 332	694	421	289	178
	Q	244	1 948	579	351	241	148
	H	165	1 316	390	236	162	100
13	L	432	3 452	1 030	624	429	264
	M	332	2 652	790	479	329	202
	Q	276	2 204	656	398	273	168
	H	192	1 532	454	275	189	116
14	L	489	3 908	1 167	707	486	299
	M	368	2 940	877	531	365	225
	Q	310	2 476	738	447	307	189
	H	210	1 676	498	302	207	127

注<sup>(2)</sup>及び注<sup>(3)</sup>は、附属書 M 表 2 の注参照。

**M.5 誤り訂正符号化** 誤り訂正符号化手順及び誤り訂正レベルは、本体の 8.5 で規定のとおりとする。

モデル 1 では、最初のデータコード語は 4 ビットだけで構成しているの、4 個のゼロビットを前に付加し、リードソロモンの計算時には 8 ビットのコード語として処理する。附属書 M 表 4 及び附属書 M 表 5 に、各型番及び誤り訂正レベルに対する、コード語の総数（残余コード語を含む）、誤り訂正コード語の総数、RS ブロックの構造及び数を示す。

モデル 1 において、残余コード語は、シンボル容量を満たすために誤り訂正コード語の最終ブロックの後に付加する埋め草コード語とする。残余コード語は、他の目的では機能しない。例えば、14-H 型シンボルでは、101 のデータ及び誤り訂正コード語が 6 ブロックあり、総計で 606 のコード語となる。シンボルは 610 コード語を含んでいるので、4 個の残余コード語を最後に付加する。埋め草コード語 **11101100** 及び **00010001** は、残余コード語として交互に使用する。

誤り訂正アルゴリズムを続けるために、データコード語は、附属書 M 表 4 及び附属書 M 表 5 で示す RS ブロック数に分割する。誤り訂正コード語は、各 RS ブロックに対して計算し、データコード語に付加する。

附属書 M 表 4 モデル 1 の 1 型~8 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注 <sup>4</sup> )	残余コード語数
1	26	L	7	1	(26, 19, 2) (注 <sup>5</sup> )	0
		M	10	1	(26, 16, 4) (注 <sup>5</sup> )	0
		Q	13	1	(26, 13, 6) (注 <sup>5</sup> )	0
		H	17	1	(26, 9, 8) (注 <sup>5</sup> )	0
2	46	L	10	1	(46, 36, 4) (注 <sup>5</sup> )	0
		M	16	1	(46, 30, 8)	0
		Q	22	1	(46, 24, 11)	0
		H	30	1	(46, 16, 15)	0
3	72	L	15	1	(72, 57, 7) (注 <sup>5</sup> )	0
		M	28	1	(72, 44, 14)	0
		Q	36	1	(72, 36, 18)	0
		H	48	1	(72, 24, 24)	0
4	100	L	20	1	(100, 80, 10)	0
		M	40	1	(100, 60, 20)	0
		Q	50	1	(100, 50, 25)	0
		H	66	1	(100, 34, 33)	0
5	134	L	26	1	(134, 108, 13)	0
		M	52	1	(134, 82, 26)	0
		Q	66	1	(134, 68, 33)	0
		H	88	2	(67, 23, 22)	0
6	170	L	34	1	(170, 136, 17)	0
		M	64	2	(85, 53, 16)	0
		Q	84	2	(85, 43, 21)	0
		H	112	2	(85, 29, 28)	0
7	212	L	42	1	(212, 170, 21)	0
		M	80	2	(106, 66, 20)	0
		Q	104	2	(106, 54, 26)	0
		H	138	3	(70, 24, 23)	2
8	256	L	48	2	(128, 104, 12)	0
		M	96	2	(128, 80, 24)	0
		Q	128	2	(128, 64, 32)	0
		H	168	3	(85, 29, 28)	1

注<sup>4</sup>)  $(c, k, r)$   $c$ =総コード語数 [本体の表 1 の (E) 参照],  $k$ =データコード語数,  $r$ =誤り訂正数

(注<sup>5</sup>) 誤り訂正数は, 復号誤りの可能性を低減するために, 誤り訂正コード語数の半分未満となっている。

附属書 M 表 5 モデル 1 の 9 型～14 型における誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック数	RS ブロック (注 <sup>4</sup> )	残余コード語数
9	306	L	60	2	(153, 123, 15)	0
		M	120	2	(153, 93, 30)	0
		Q	150	3	(102, 52, 25)	0
		H	204	3	(102, 34, 34)	0
10	358	L	68	2	(179, 145, 17)	0
		M	136	2	(179, 111, 34)	0
		Q	174	3	(119, 61, 29)	1
		H	232	4	(89, 31, 29)	2
11	416	L	80	2	(208, 168, 20)	0
		M	160	4	(104, 64, 20)	0
		Q	208	4	(104, 52, 26)	0
		H	270	5	(83, 29, 27)	1
12	476	L	92	2	(238, 192, 23)	0
		M	184	4	(119, 73, 23)	0
		Q	232	4	(119, 61, 29)	0
		H	310	5	(95, 33, 31)	1
13	542	L	108	3	(180, 144, 18)	2
		M	208	4	(135, 83, 26)	2
		Q	264	4	(135, 69, 33)	2
		H	348	6	(90, 32, 29)	2
14	610	L	120	3	(203, 163, 20)	1
		M	240	4	(152, 92, 30)	2
		Q	300	5	(122, 62, 30)	0
		H	396	6	(101, 35, 33)	4

注<sup>4</sup>)は、附属書 M 表 4 の注参照。

**M.6 最終的なメッセージコード語列の構成** メッセージのコード語の総数は、附属書 M 表 4 及び附属書 M 表 5 に示すとおり、シンボル表示が可能なコード語の総数と常に等しい。

モデル 1 シンボルについては、附属書 M 表 4 及び附属書 M 表 5 で定義されている残余コード語数を付加し、次のとおり、 $n$  ブロックのデータ及び  $n$  ブロックの誤り訂正コード語からなる最終列を構成する。

データブロック 1, データブロック 2, ..., データブロック  $n$ , 誤り訂正ブロック 1, 誤り訂正ブロック 2, ..., 誤り訂正ブロック  $n$ , 残余コード語

例 モデル 1 の 10-H 型シンボル

全容量： 358 コード語

データコード語： 124 (31 データコード語が 4 ブロック)

誤り訂正コード語： 232 (58 誤り訂正コード語が 4 ブロック)

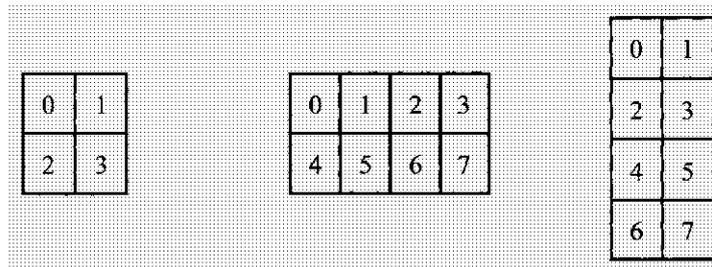
必要な残余コード語： 2

1～58 の誤り訂正コード語は 1～31 のデータコード語で計算し、59～116 の誤り訂正コード語は、32～62 のデータコード語で計算する。同様に残りの 117～174 の誤り訂正コード語は、63～93 のデータコード語で、したがって、最終的なメッセージコード語列は、次のようになる。

データコード語 1, 2, ..., 31, 32, ..., 62, 63, ..., 93, 94, ..., 124, 誤り訂正コード語 1, 2, ..., 58, 59, ..., 116, 117, ..., 174, 175, ..., 232, 残余コード語 1, 2

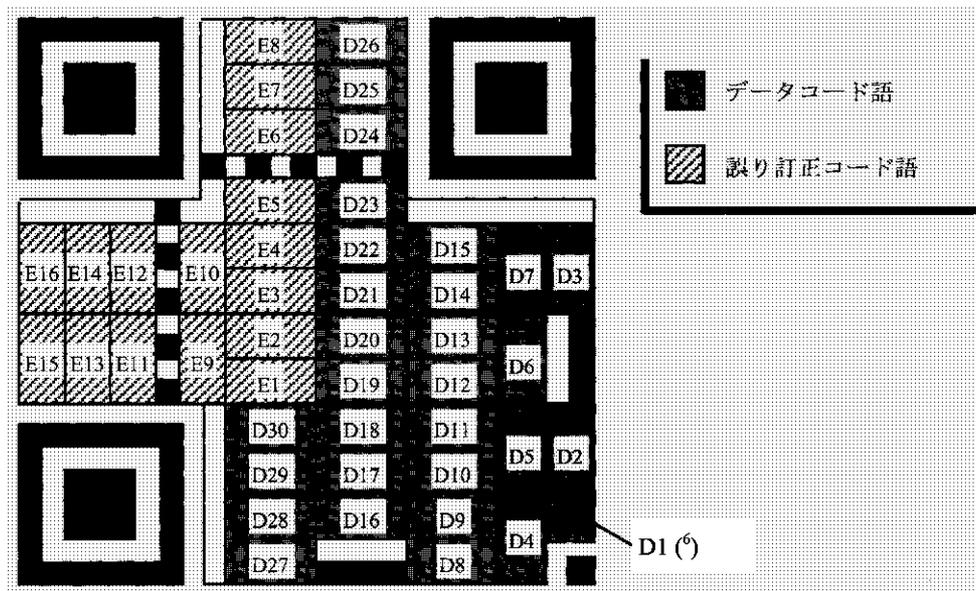
**M.7 コード語の配置**

**M.7.1 シンボルキャラクタ表示** モデル1のQRコードシンボルには、2種類のシンボルキャラクタがある。最初のコード語（4ビット）は、2×2モジュールブロックのシンボルキャラクタとして表示する。その他のコード語は、シンボルにおいて2×4モジュールブロックとして表示する。これらブロックの配置方法も、垂直配列（幅が2モジュール、高さが4モジュール）及び水平配列（幅が4モジュール、高さが2モジュール）の2種類がある。附属書M図5に1個のシンボルキャラクタのモジュール配列をそれぞれの配置方法ごとに示す。同図において、“0”は最下位ビットに対応し、“7”は最上位ビットに対応する。最下位ビットは、常にシンボルキャラクタの左上のモジュールに位置し、左から右、上から下へとビットが続き、右下のモジュールに位置する最上位ビットで終わる。“0”ビットは、明のモジュールを示し、“1”ビットは暗のモジュールを示す。



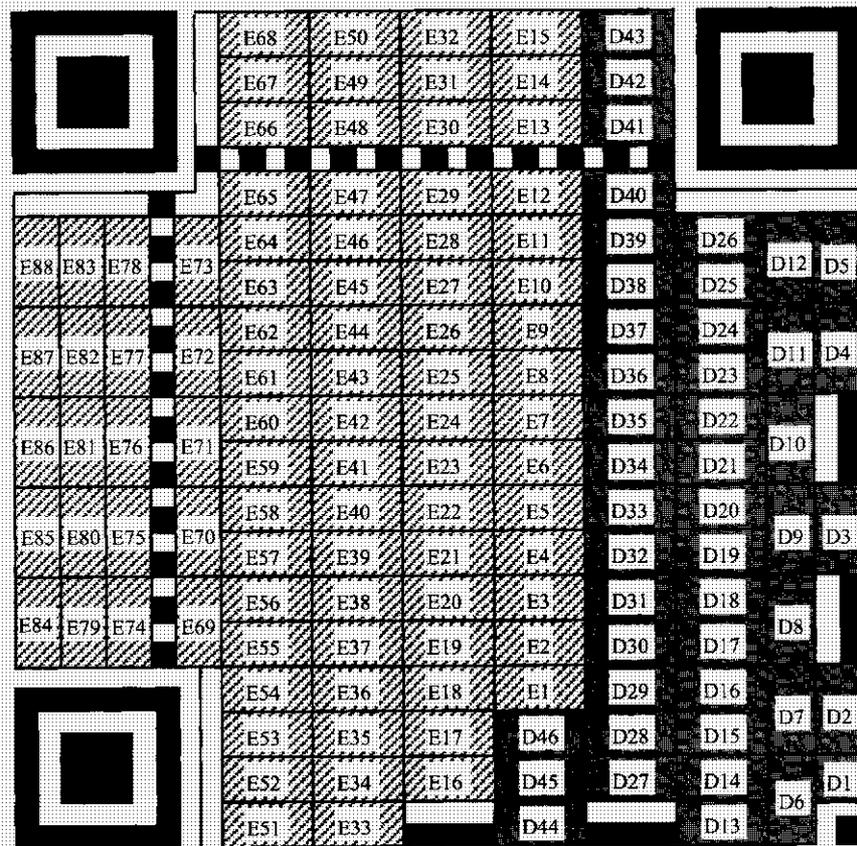
附属書 M 図 5 1個のキャラクタのモジュール配列

**M.7.2 機能パターンの配置** 正方形の空マトリックスは、使用する型番に従って水平及び垂直に配置されるモジュールから構成される。位置検出パターン、分離パターン、タイミングパターン及び拡張キャラクタに対応する位置は、特定の暗モジュール又は明モジュールで占められる。形式情報のモジュール位置は、一時的に空とする。これらの位置は、すべての型番に共通し、附属書M図6及び附属書M図7に示す。



注(6) 最初のデータコード語 D1 は 4 ビットで構成する。リードソロモン誤り訂正符号の計算時は、4 個のゼロビットを頭に付けて 8 ビットのコード語として処理する。

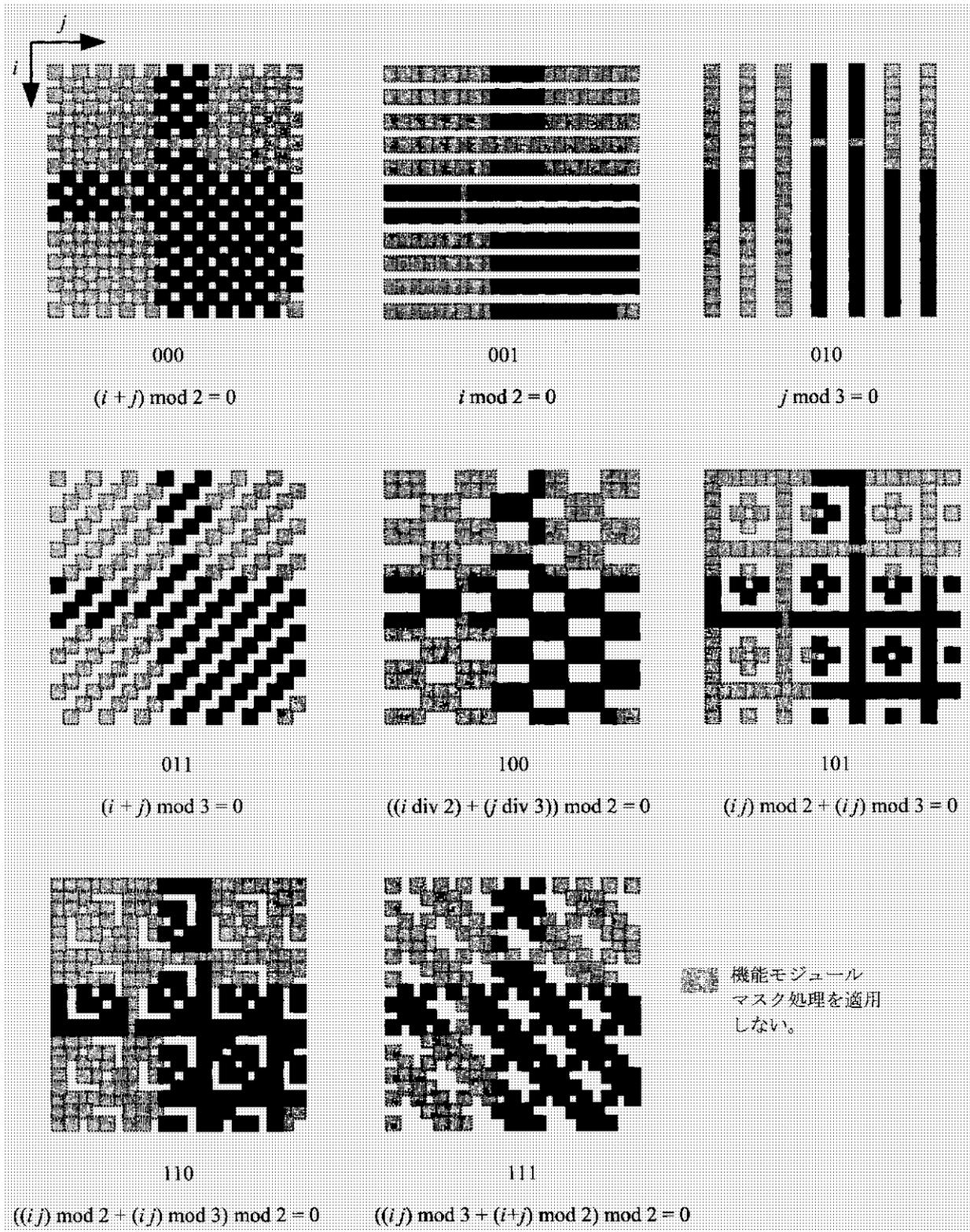
附属書 M 図 6 モデル1シンボルの2-M型におけるシンボルキャラクタ配列



D1～D23 :	データブロック 1
D24～D46 :	データブロック 2
E1～E44 :	誤り訂正ブロック 1
E45～E88 :	誤り訂正ブロック 2

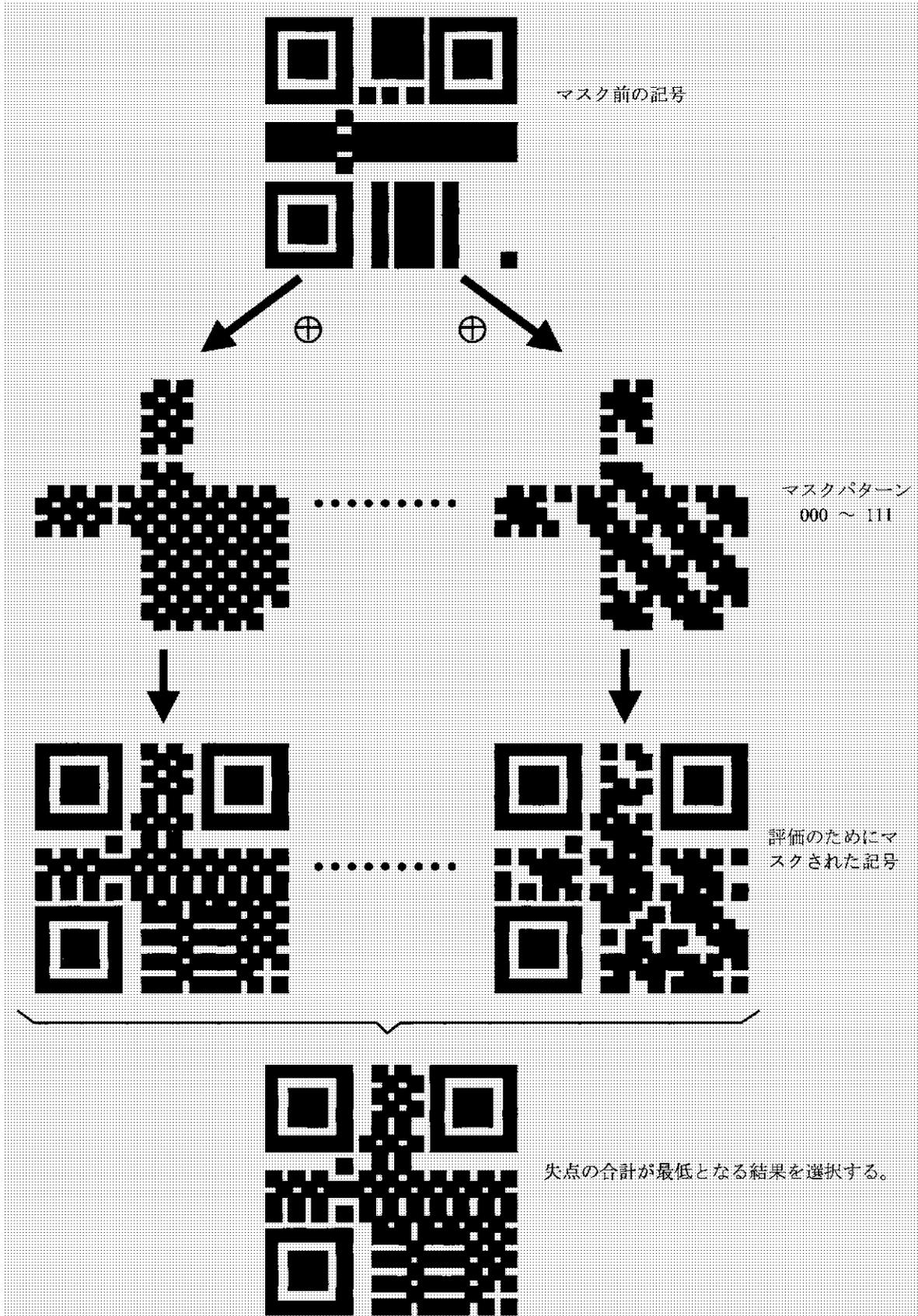
附属書 M 図 7 モデル 1 シンボルの 5-H 型におけるシンボルキャラクタ配列

**M.8 マスク処理** マスク処理は本体の 8.8 に規定するとおり実行する。附属書 M 図 8 及び附属書 M 図 9 はそれぞれ、モデル 1 の 1 型シンボルに対する全マスクパターン及びそのシミュレーションを示す。



- 備考1. 各パターン下の3ビットがマスクパターン情報となる。
2. 3ビットの下の式は、マスクパターン生成条件を示し、式を満たすモジュールが暗モジュールに対応する。

附属書 M 図 8 モデル 1 の 1 型シンボルに対する全マスクパターン



附属書 M 図 9 モデル 1 シンボルにおけるマスク処理のシミュレーション

**M.9 形式情報** 形式情報は、**本体の 8.9**に規定するとおり計算し、マスク処理後、シンボルの予約された領域に配置しなければならない。

モデル 1 シンボルの形式情報に対するマスクパターンは、010100000100101 とする。二つの異なるマスクパターンを用いることで、読取り時のシンボルモデルの自動識別を確実にする。

**M.10 構造的接続** モデル 1 の構造的接続は、正確にモデル 2 と同じ規則 (**本体の 9.**に規定) に従う。モデル 1 のシンボルにおいては、構造的接続モード指示子 0011 を最初のシンボルキャラクタ (4 モジュール) に置く。次に 2 個の構造的接続コード語が後続する。モデル 1 では、これらの 2 個のコード語は 2 番目及び 3 番目のシンボルキャラクタとなる。

**M.11 シンボルの印刷及びマーキング** **本体の 10.**を完全に適用する。

**M.12 シンボルの品質** **本体の 11.**を完全に適用する。

**M.13 復号手順の概要** **本体の 12.**を完全に適用する。

**M.14 参照復号アルゴリズム** **本体の 13.**に記述された参照復号アルゴリズム **g)**, **h)** 及び **l)** を、次の方法に置き換えて適用する。参照復号アルゴリズム **m)** は適用しない。

**g)** タイミングパターンにおける暗及び明のモジュールの中心点の平均間隔を  $X$  として再定義する。同様に左側のタイミングパターンにおける暗及び明のモジュールの中心点の平均間隔を  $Y$  寸法と計算する。

**h)** 次に基づいてサンプリンググリッドを設定する。

1) 上側のタイミングパターンを通り、 $y$  軸の垂直方向でそのタイミングパターンと平行な水平線、その水平基準線から上に、6 本の線、及びその下にシンボルの型番に必要とされる数の線 (それぞれの水平線の垂直方向の間隔は  $Y$ )。

2) 左側のタイミングパターンを通り、 $x$  軸の水平方向でそのタイミングパターンと平行な垂直線、その垂直基準線から左に 6 本の線、及びその右にシンボルの型番に必要とされる数の線 (それぞれの垂直線の水平方向の間隔は  $X$ )。

**i)** 附属書 **M** の配置規則に従い、シンボルのコード語を求める。

**M.15 自動識別能力** **本体の 14.**を完全に適用する。

**M.16 送信データ** **本体の 15.2** 及び **本体の 15.3** を除き、**本体の 15.**を適用する (ECI に関する支持は無視する。)

#### **M.17 附属書**

— 附属書 **A**, **B**, **F**, **H**, **I**, **J**, **K**, 及び **L** は、完全に適用する。

— 附属書 **D** 及び **E** は、モデル 1 シンボルには適用しない。

— 附属書 **C**

1) **C.1** の最初の段落の最後の文を、次のとおり置き換えて適用する。

最後に、どのようなマスクパターン及び誤り訂正レベルの組合せでも形式情報のビットパターン

がすべてゼロにならないように、また、モデル 1 シンボルがモデル 2 シンボルと自動識別できるように、ビット文字列を 010100000100101 と XOR 演算し、マスク処理をする。

- 2) C.1 の例における最後の 2 行を次のとおり置き換える。

マスクと XOR 演算 010100000100101

結果： 011110011111001

- 3) C.2 の最初の段落の最初の文を、次のとおり置き換えて適用する。

マスクパターン 010100000100101 でビット列を XOR 演算し、形式情報のマスク処理を解除する。

- 附属書 G は、手順 3 を次のとおり置き換えて適用する。

21×21 の空マトリックス内に位置検出要素パターン、タイミングパターン及び拡張パターンを配置し、形式情報のモジュール位置は、一時的に空とする。手順 2 で得られたコード語を、この附属書に従い、マトリックス内に配置する。

## 附属書 1 (規定) マイクロ QR コード

**序文** マイクロ QR コードは、シンボルの左上コーナーに独自の位置検出パターンをもち、シンボルの位置及び大きさが容易に検索できるようになっているマトリックス式シンボル体系である。印字面積が小さく大量のデータを必要としないアプリケーションを対象に標準の QR コードよりシンボルサイズが小さくなるよう設計している。四種類のシンボルサイズを規定しており、各シンボルサイズに応じて誤り訂正レベルを設定できる。様々な技術でシンボルが生成できるよう、モジュール寸法は、利用者が指定できる。

マイクロ QR コードのデータ文字の符号化方法、誤り訂正方法など、多くのシンボルの特徴は QR コードと同じである。

### 1. マイクロ QR コードの仕様

**1.1 基本的特性** マイクロ QR コードは、次の特性をもつマトリックス式シンボル体系とする。

#### a) 符号化文字集合

- 1) 数字データ (数字 0~9)
- 2) 英数字データ (数字 0~9, 大文字 A~Z, 9 個の特殊文字: スペース, \$, %, \*, +, -, ., /, :)
- 3) 8 ビットバイトデータ (JIS X 0201 に基づくラテン文字, 片仮名用 8 ビット符号の文字集合)
- 4) 漢字 (JIS X 0208 附属書 1 のシフト符号化表現で規定された文字)

**備考** マイクロ QR コードで符号化する漢字データは、13 ビットで圧縮が可能な 8140<sub>HEX</sub>~9FFC<sub>HEX</sub> 及び E040<sub>HEX</sub>~EBBF<sub>HEX</sub> までの範囲とする。

**b) データ表示** 暗のモジュールは 2 進法の 1, 明のモジュールは 2 進法の 0 を意味する。

**c) シンボルサイズ (クワイエットゾーンを除く)** 11×11 モジュール, 13×13 モジュール, 15×15 モジュール及び 17×17 モジュールの 4 サイズ。

**d) シンボル当たりの文字数 (最大シンボルサイズに対して)**

- |                |       |
|----------------|-------|
| 1) 数字データ       | 35 文字 |
| 2) 英数字データ      | 21 文字 |
| 3) 8 ビットバイトデータ | 15 文字 |
| 4) 漢字データ       | 9 文字  |

**e) 選択可能な誤り訂正** シンボルサイズに応じて、次のシンボルコード語の復元が可能なリードソロモン誤り訂正を選択できる。

11×11 モジュール: 誤り検出だけ

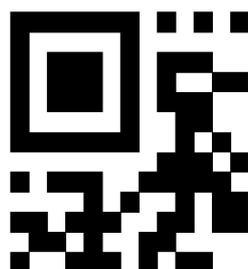
13×13 モジュール: L (7%), M (15%) の 2 段階

15×15 モジュール: L (7%), M (15%) の 2 段階

17×17 モジュール: L (7%), M (15%), Q (25%) の 3 段階

**f) コード形式** マトリックス

**g) 方向の独立性** 独立している。

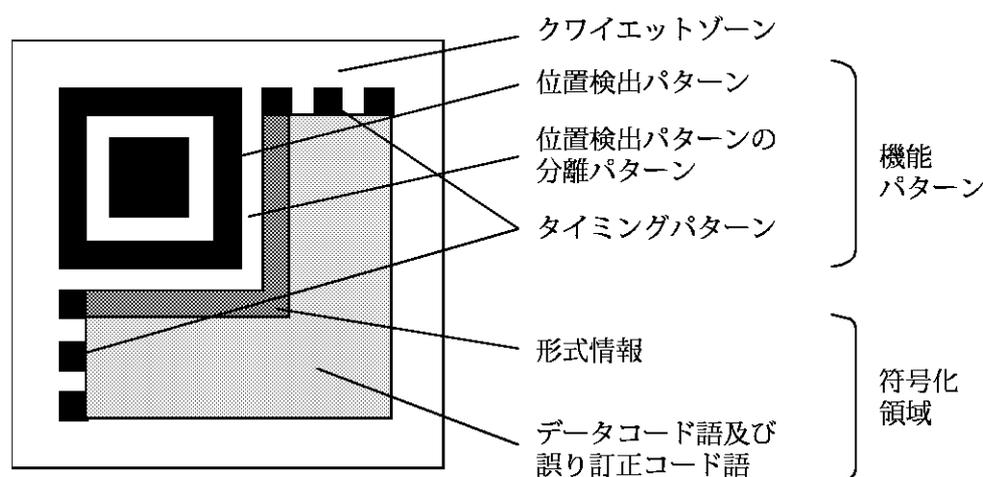


附属書 1 図 1 マイクロ QR コードシンボルの例

1.2 追加機能 マイクロ QR コードは、次に示す追加機能を備える。

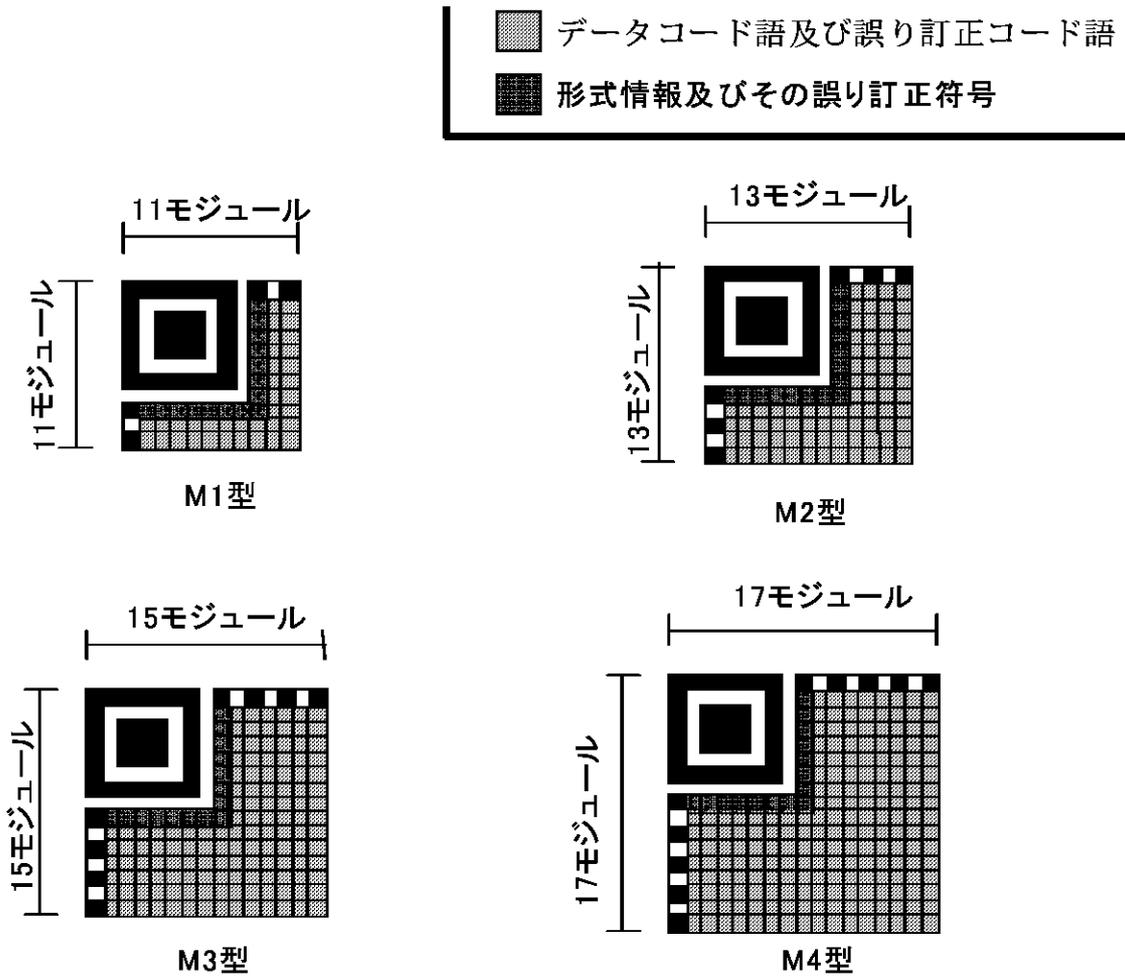
a) マスク処理 復号効率を妨げるようなモジュール配列の出現を最小限に抑え、シンボルの明及び暗のモジュール比率を 1:1 に近づける。

1.3 シンボルの構造 マイクロ QR コードシンボルは、一般的には正方形に配置された正方形のモジュールからなり、符号化領域及び機能パターン（すなわち、位置検出パターン、分離パターン及びタイミングパターン）で構成する。機能パターンはデータの符号化領域としては使用しない。シンボルは、その 4 辺の周囲にクワイエットゾーンをもつ。附属書 1 図 2 は、マイクロ QR コードシンボルの構造を示す。



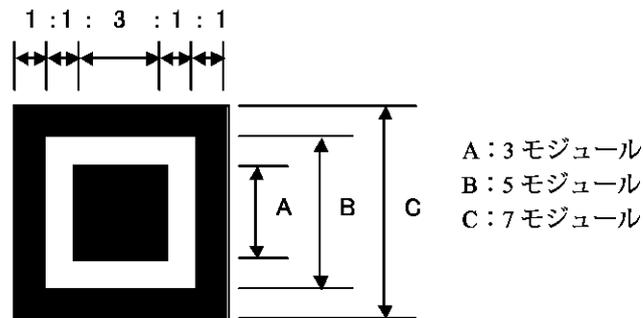
附属書 1 図 2 マイクロ QR コードシンボルの構造

1.4 シンボルの型番及び大きさ マイクロ QR コードシンボルは、M1 型～M4 型までの 4 種類の大きさをもつ。M1 型は 11×11 モジュール、M2 型は 13×13 モジュール、M3 型は 15×15 モジュール、M4 型は 17×17 モジュールであり、型番が一つ上がるごとに一辺につき 2 モジュールずつ増加する。附属書 1 図 3 はマイクロ QR コードシンボルすべての型番の構造を示す。



附属書 1 図 3 マイクロ QR コードシンボルの全型番

**1.5 位置検出パターン** 位置検出パターンは3個の同心正方形が重なった形状で、暗の7×7モジュール、明の5×5モジュール及び暗の3×3モジュールで構成し、附属書 1 図 2 に示すようにシンボルの左上に配置する。位置検出パターンにおけるモジュール幅の比率は附属書 1 図 4 で示すように1:1:3:1:1とする。これに類似するパターンがシンボルの任意の位置に出現する可能性を抑え、視野内でマイクロ QR コードシンボルの認識を容易にすることができるようにシンボルを符号化する。位置検出パターンを識別することによって、視野内でのシンボルの位置及び方向を明確に認識できる。



附属書 1 図 4 位置検出パターンの構造

**1.6 分離パターン** 1モジュール幅の分離パターンは、**附属書 1 図 2** で示すように位置検出パターンと符号化領域との間に配置し、すべて明のモジュールで構成する。

**1.7 タイミングパターン** 水平及び垂直タイミングパターンは、それぞれ 1 モジュール幅で暗と明とが交互になっているモジュールの行又は列で構成し、暗のモジュールで始まり、暗のモジュールで終了する。水平タイミングパターンは、シンボルの 0 行目、位置検出パターンの右側に配置する。垂直タイミングパターンは同様にシンボルの 0 列目、位置検出パターンの下側に配置する。これらにより、シンボル密度を決めることができ、またモジュール座標を決めるデータ位置が得られる。

**1.8 符号化領域** この領域には、データを示すシンボルキャラクタ、それらに対する誤り訂正コード語及び形式情報を含む。シンボルキャラクタの詳細については、QR コードモデル 2 に準じる。また、形式情報の詳細については **2.6** を参照。

**1.9 クワイエットゾーン** シンボルの四辺の周囲を囲む、何も表示されない 2 モジュール幅の領域。その公称反射値は、明のモジュール値と等しい。

## 2. 要求事項

**2.1 符号化手順の概要** ここでは、入力データをマイクロ QR コードシンボルに変換する際に要求される手順の概要を規定する。

**手順 1 データ分析** 符号化する種々の異なる文字を識別するため、入力データ列を分析する。マイクロ QR コードには、異なる文字集合を効果的にシンボルキャラクタに変換するための幾つかのモード (**2.2** 参照) がある。最も効果的にデータを 2 進文字列へ変換するため、必要に応じてモードを切り替える。必要な誤り訂正レベルを選択する。利用者が使用するシンボル型番を指定していない場合は、データが収容できる最小型番を選択する。シンボル型番及びそのデータ容量の一覧を **附属書 1 表 1** に示す。

**手順 2 データの符号化** **2.3.1**~**2.3.4** で規定するように、適用中のモードの規則に従ってデータ文字をビット列に変換する。新しいモードセグメントの開始時にモード変更に必要なモード指示子を挿入し、データ列の最後に終端パターンを挿入する。得られたビット列を 8 ビットコード語に分割していき、その型番に要求されるデータコード語数を満たすために必要に応じて埋め草コード語を付加する。

**手順 3 誤り訂正符号データ分析** 誤り訂正アルゴリズムを実行するため、誤り訂正コード語を生成し、データコード語列の後に付加する。

**手順 4 マトリックスにおけるモジュールの配置** マトリックスに位置検出パターン、分離パターン及びタイミングパターンとともにコード語モジュールを配置する。

**手順 5 マスク処理** シンボルの符号化領域にマスク処理パターンを適用する。明と暗のモジュールバランスを最適にし、また望ましくないパターンの出現を最小限に抑えるパターンを評価し、選択する。

**手順 6 形式情報** 形式情報を生成し、シンボルを完成させる。

附属書 1 表 1 マイクロ QR コードの全型番のデータ容量

型番	モジュール 数/辺 (A)	機能パターン モジュール (B)	形式情報 モジュール (C)	(C)以外のデータ モジュール (D=A <sup>2</sup> -B-C)	データ容量 [コード語] <sup>(1)</sup> (E)
M1	11	70	15	36	5
M2	13	74	15	80	10
M3	15	78	15	132	17
M4	17	82	15	192	24

注<sup>(1)</sup> M1 型及び M3 型の場合、最後のデータコード語は 4 ビット長とし、誤り訂正コード語生成時には 0000 を後に付け、8 ビット長とする。

**2.2 データ分析** 内容を決定するために入力データ文字列を分析し、2.3 に記述するように、各文字列の符号化に適切なモードを選択する。数字モードから漢字モードに進むにつれ、文字あたりに要求されるビット数が増加する。混在モードでは、データのビット列の長さを最小にするため、シンボル内でモードを切り替えることができ、部分的には、一つのモードで、他の部分より効率的に符号化できる（例えば、数字列の後に英数字列が続く場合）。一般的には、文字あたりの最少ビットを要求するモードでデータを符号化することが最も効果的であるが、モード変更に伴ってモード指示子及び文字数指示子のオーバーヘッドがあるため、文字数の少ない場合にもモードを変更することが常に最短のビット列となるわけではない。また、シンボルの容量は、ある型番から次の型番へと段階的に増加するため、すべての事例で最大の変換効率を達成することは、必ずしも必要ではない。

**2.2.1 数字モード** 数字モードは、十進数集合 (0~9) のデータを、通常 3 文字を 10 ビットで符号化する。

**2.2.2 英数字モード** 英数字モードは、45 文字 [10 個の数字 (0~9) 2.2.1 参照]、26 個のアルファベット文字 (A~Z) 及び 9 個の記号 (スペース, \$, %, \*, +, -, ., /, :) のデータを符号化する。通常、2 文字を 11 ビットで符号化する。

**2.2.3 8ビットバイトモード** 8ビットバイトモードは、JIS X 0201 に基づく 8 ビットのラテン文字・片仮名用 8 ビット符号に規定された文字を扱う。このモードにおいて、データは、1 文字を 8 ビットで符号化する。

**参考** このモードで使用する符号化文字集合は、QR コードを使用するアプリケーションで指定する符号化文字集合とする。ただし、アプリケーションごとに使用する符号化文字集合が異なると、複数のアプリケーションを扱うシステムでは符号化文字集合の切替えが必要となり、システムが複雑になる。このため、国際的なアプリケーションとしては JIS X 0201、ISO 8859-15 及び JIS X 0221-1 (UTF-8) の使用を推奨する。各国国内に限られたアプリケーションでは、ISO 8859 の各パートの使用を推奨する。日本では JIS X 0201 の使用を推奨する。UTF-8 は UCS-2 (Universal Multiple-Octet Coded Character Set-2) で定義される文字集合を用いて記述された文字列をバイト列に変換する方式の一つで、1 文字を 1~6 バイトの可変長マルチバイトに変換する。

**2.2.4 漢字モード** 漢字モードは、JIS X 0208 の附属書 1 に規定された漢字を扱う。各 2 バイトの文字値は 13 ビット 2 進コード語に圧縮する。

**2.2.5 混在モード** マイクロ QR コードシンボルは、2.2.1~2.2.4 に規定する任意のモードの組合せによるデータ列を扱うことができる。

**2.3 データの符号化** 入力データを、それぞれのモードで、1 個又は複数のセグメントで構成するビット列に変換する。完全なビット列は、次をもつセグメントで構成する。

- モード指示子
- 文字数指示子
- データビット列

各セグメントは、モード指示子の最初（最上位けた）のビットで始まり、データビット列の最後（最下位けた）のビットで終了する。セグメントの長さは、適用中のモードの規則及び入力データの文字数によって明確に定義するので、セグメントとセグメントとの間には明確な分離を示すパターンは存在しない。

規定のモードで入力データ列を符号化するには、2.3.1～2.3.4 に定義する手順に従う。附属書 1 表 2 に各型番におけるモード指示子のビット列を示す。附属書 1 表 3 に各シンボル型番における文字数指示子の長さを示す。

附属書 1 表 2 各型番におけるモード指示子のビット列

型番	モード指示子長	数字モード	英数字モード	8ビットバイトモード	漢字モード
M1	0 ビット	-	-	-	-
M2	1 ビット	0	1	-	-
M3	2 ビット	00	01	10	11
M4	3 ビット	000	001	010	011

附属書 1 表 3 文字数指示子のビット数

型番	文字数指示子			
	数字モード	英数字モード	8ビットバイトモード	漢字モード
M1	3	-	-	-
M2	4	3	-	-
M3	5	4	4	3
M4	6	5	5	4

**2.3.1 数字モード** 入力データ文字列を 3 けたのグループに分割し、各グループを 10 ビットの 2 進値に変換する。入力けた数が 3 の倍数でない場合、最終の 1 けた又は 2 けたは、それぞれ 4 又は 7 ビットに変換する。次に 2 進データを接続し、モード指示子及び文字数指示子を最初の部分に付加する。

附属書 1 表 3 で定義するとおり、数字モードのモード指示子は 0, 1, 2 又は 3 ビット長であり、文字数指示子は 3, 4, 5 又は 6 ビット長である。モード指示子及び入力データ文字数を 2 進値に変換し、モード指示子、文字数指示子、2 進データ列の順で付加する。

**例1.** (M4-M型シンボル)

入力データ： **01234567**

1. 3 けたのグループに分割 **012 345 67**

2. 各グループを 2 進値に変換 **012 = 0000001100**

**345 = 0101011001**

**67 = 1000011**

3. 2 進データを順に接続 **0000001100 0101011001 1000011**

4. 文字数指示子を 2 進 (M4-M 型は 6 ビット) に変換

入力データ文字数：**8 = 001000**

5. モード指示子 (M4-M 型は **000**) 及び文字数指示子を 2 進データに付加

**000 001000 0000001100 0101011001 1000011**

**例2.** (M4-M型シンボル)

- 入力データ： **0123456789012345**
1. 3けたのグループに分割 **012 345 678 901 234 5**
  2. 各グループを2進値に変換
    - 012 = 0000001100**
    - 345 = 0101011001**
    - 678 = 1010100110**
    - 901 = 1110000101**
    - 234 = 0011101010**
    - 5 = 0101**
  3. 2進データを順に接続
    - 0000001100 0101011001 1010100110 1110000101 0011101010 0101**
  4. 文字数指示子を2進値 (M4-M型は6ビット) に変換
    - 入力データ文字数：**16 = 010000**
  5. モード指示子 (M4-M型は**000**) と文字数指示子を2進データに付加
    - 000 010000 0000001100 0101011001 1010100110 1110000101 0011101010 0101**

数字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B = M + C + 10(D \text{ DIV } 3) + R$$

- ここに、
- B** : ビット列のビット数
  - M** : モード指示子のビット数 (附属書1表3参照)
  - C** : 文字数指示子のビット数 (附属書1表3参照)
  - D** : 入力データ文字数
  - R** : (D MOD 3)=0ならば0, (D MOD 3)=1ならば4, (D MOD 3)=2ならば7

**2.3.2 英数字モード** 各入力文字は、附属書1表4に基づいて0~44の文字値を割り当てる。

附属書1表4 英数字モードでの符号化・復号表

0	0	6	6	C	12	I	18	O	24	U	30	SP	36	.	42
1	1	7	7	D	13	J	19	P	25	V	31	\$	37	/	43
2	2	8	8	E	14	K	20	Q	26	W	32	%	38	:	44
3	3	9	9	F	15	L	21	R	27	X	33	*	39		
4	4	A	10	G	16	M	22	S	28	Y	34	+	40		
5	5	B	11	H	17	N	23	T	29	Z	35	-	41		

入力文字を、2けたのグループに分け、11ビットの2進コードに符号化する。最初の文字の文字値は45で乗算し、2番目の文字値をそれに加算する。合計値を11ビットの2進値に変換する。入力文字数が2の倍数でない場合、最終文字の文字値は6ビットの2進値に変換する。附属書1表3で定義しているように、英数字モードのモード指示子は1, 2又は3ビット長であり、文字数指示子は3, 4又は5ビット長である。モード指示子及び入力データ文字数は2進値に変換し、モード指示子、文字数指示子、2進データ列の順に付加する。

**例** (M4-M型シンボル)

入力データ：**AC-42**

1. 附属書1表5に従って文字値を割り当て **AC-42 (10, 12, 41, 4, 2)**

- |   |          |                |                                       |   |
|---|----------|----------------|---------------------------------------|---|
| 2. 結果を2けたの十進数のグループに分割                         | (10, 12) | (41, 4)        | (2)                                   |   |
| 3. 各グループを11ビットの2進値に変換                         | (10, 12) | $10*45+12=462$ | <b>00111001110</b>                    |   |
|   | (41, 4)  | $41*45+4=1849$ | <b>11100111001</b>                    |   |
|   | (2)      | 2              | <b>000010</b>                         |   |
| 4. 2進データを順に接続                                 |          |                | <b>00111001110 11100111001 000010</b> |   |
| 5. 文字数指示子を2進(M4-M型は5ビット)に変換                   |          |                |                                       | 入力データ文字数: <b>5 = 00101</b>                      |
| 6. モード指示子(M4-M型は <b>001</b> )と文字数指示子を2進データに付加 |          |                |                                       | <b>001 00101 00111001110 11100111001 000010</b> |

英数字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B = M + C + 11(D \text{ DIV } 2) + 6(D \text{ MOD } 2)$$

ここに、  
*B*: ビット列のビット数  
*M*: モード指示子のビット数 (附属書 1 表 3 参照)  
*C*: 文字数指示子のビット数 (附属書 1 表 3 参照)  
*D*: 入力データ文字数

**2.3.3 8ビットバイトモード** このモードでは、1個の8ビットコード語は、附属書 1 表 5 で示す入力データ文字に対する文字値を直接割り当てる (密度は、1文字当たり8ビット)。

附属書 1 表 5 ラテン文字, 片仮名用 8 ビット符号の符号化/復号表

文字 16 進													
NUL	00	SP	20	@	40	'	60	80	A0	タ	C0	E0	
SOH	01	!	21	A	41	a	61	81	。	A1	チ	C1	E1
STX	02	"	22	B	42	b	62	82	「	A2	ツ	C2	E2
ETX	03	#	23	C	43	c	63	83	」	A3	テ	C3	E3
EOT	04	\$	24	D	44	d	64	84	、	A4	ト	C4	E4
ENQ	05	%	25	E	45	e	65	85	・	A5	ナ	C5	E5
ACK	06	&	26	F	46	f	66	86	ヲ	A6	ニ	C6	E6
BEL	07	'	27	G	47	g	67	87	ア	A7	ヌ	C7	E7
BS	08	(	28	H	48	h	68	88	イ	A8	ネ	C8	E8
HT	09	)	29	I	49	i	69	89	ウ	A9	ノ	C9	E9
LF	0A	*	2A	J	4A	j	6A	8A	エ	AA	ハ	CA	EA
VT	0B	+	2B	K	4B	k	6B	8B	オ	AB	ヒ	CB	EB
FF	0C	,	2C	L	4C	l	6C	8C	ヤ	AC	フ	CC	EC
CR	0D	-	2D	M	4D	m	6D	8D	ユ	AD	ヘ	CD	ED
SO	0E	.	2E	N	4E	n	6E	8E	ヨ	AE	ホ	CE	EE
SI	0F	/	2F	O	4F	o	6F	8F	ツ	AF	マ	CF	EF
DLE	10	0	30	P	50	p	70	90	ー	B0	ミ	D0	F0
DC1	11	1	31	Q	51	q	71	91	ア	B1	ム	D1	F1
DC2	12	2	32	R	52	r	72	92	イ	B2	メ	D2	F2
DC3	13	3	33	S	53	s	73	93	ウ	B3	モ	D3	F3
DC4	14	4	34	T	54	t	74	94	エ	B4	ヤ	D4	F4
NAK	15	5	35	U	55	u	75	95	オ	B5	ユ	D5	F5
SYN	16	6	36	V	56	v	76	96	カ	B6	ヨ	D6	F6
ETB	17	7	37	W	57	w	77	97	キ	B7	ラ	D7	F7
CAN	18	8	38	X	58	x	78	98	ク	B8	リ	D8	F8
EM	19	9	39	Y	59	y	79	99	ケ	B9	ル	D9	F9
SUB	1A	:	3A	Z	5A	z	7A	9A	コ	BA	レ	DA	FA
ESC	1B	;	3B	[	5B	{	7B	9B	サ	BB	ロ	DB	FB
IS4	1C	<	3C	¥	5C		7C	9C	シ	BC	ワ	DC	FC
IS3	1D	=	3D	]	5D	}	7D	9D	ス	BD	ン	DD	FD
IS2	1E	>	3E	^	5E	~	7E	9E	セ	BE	ゝ	DE	FE
IS1	1F	?	3F	_	5F	DEL	7F	9F	ソ	BF	。°	DF	FF

参考1. JIS X 0208 の 8 ビット符号のバイト値 80~9F 及び E0~FF までは未定義で、保留とする。これらの幾つかは、シフト符号化表現では第 1 バイトとして使用し、8 ビット符号及びシフト符号化表現を区別するのにも適用できる。詳細については、JIS X 0208 附属書 1 シフト符号化表現を参照する。

2. JIS X 0208 の 8 ビット符号のバイト値 00<sub>HEX</sub>~FF<sub>HEX</sub> は、5C<sub>HEX</sub> 及び 7E<sub>HEX</sub> を除いて、ISO/IEC 646 に対応している。

次に 2 進データを接続し、モード指示子及び文字数指示子を前に付加する。附属書 1 表 3 で定義するとおり 8 ビットバイトモードのモード指示子は 2 ビット又は 3 ビット長とし、文字数指示子は 4 ビット又は 5 ビット長とする。モード指示子及び入力データ文字数を 2 進値に変換し、モード指示子、文字数指示子及び 2 進データ列の順に付加する。

8 ビットバイトモードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B = M + C + 8D$$

ここに、  
 B: ビット列のビット数  
 M: モード指示子のビット数 (附属書 1 表 3 より)  
 C: 文字数指示子のビット数 (附属書 1 表 3 より)  
 D: 入力データ文字数

**2.3.4 漢字モード JIS X 0208** のシフト符号化表現では、漢字は、2 バイトの組合せによって表現する。漢字モードの入力データ文字は、次に示すように、13 ビット 2 進コード語に圧縮する。2 進データを接続し、モード指示子及び文字数指示子を前に付ける。附属書 1 表 3 で定義するとおり、漢字モードのモード指示子は 2 ビット又は 3 ビット長とし、文字数指示子は 3 ビット又は 4 ビット長とする。モード指示子及び入力データ文字数を 2 進値に変換し、モード指示子、文字数指示子及び 2 進データ列の順に付加する。

a) 8140<sub>HEX</sub>～9FFC<sub>HEX</sub> までのシフト符号化表現値の文字

- 1) シフト符号化表現値から 8140<sub>HEX</sub> を減算する。
- 2) 得られた値の上位バイトに C0<sub>HEX</sub> を乗じる。
- 3) 2) の結果に下位バイトを加算する。
- 4) 結果を 13 ビット 2 進文字列に変換する。

b) E040<sub>HEX</sub>～EBBF<sub>HEX</sub> までのシフト符号化表現値の文字

- 1) シフト符号化表現値から C140<sub>HEX</sub> を減算する。
- 2) 得られた値の上位バイトに C0<sub>HEX</sub> を乗じる。
- 3) 2) の結果に下位バイトを加算する。
- 4) 結果を 13 ビット 2 進文字列に変換する。

例

入力文字 (シフト符号化表現値)	“点”	“茗”
	<b>935F</b>	<b>E4AA</b>
1. 8140 又は C140 を減算する	<b>935F - 8140 = 121F</b>	<b>E4AA - C140 = 36A</b>
2. 上位バイトに C0 を乗じる	<b>12 × C0 = D80</b>	<b>23 × C0 = 1A40</b>
3. 下位バイトを加算する	<b>D80 + 1F = D9F</b>	<b>A40 + 6A = 1AAA</b>
4. 13 ビット 2 進文字列に変換する	<b>0D9F =</b> <b>0 1101 1001 1111</b>	<b>1AAA =</b> <b>1 1010 1010 1010</b>

c) 8140<sub>HEX</sub>～9FFC<sub>HEX</sub> までのシフト符号化表現値の文字及び E040<sub>HEX</sub>～EBBF<sub>HEX</sub> までのシフト符号化表現値の文字すべてに対し、入力文字の 2 進列の前にモード指示子 (2 ビット又は 3 ビット) 及び文字数指示子 2 進値 (3 ビット又は 4 ビット) を付加する。

漢字モードにおいて、任意の数のデータ文字数に対するビット列の長さは、次の式によって得られる。

$$B = M + C + 13D$$

ここに、  
 B: ビット列のビット数  
 M: モード指示子のビット数 (附属書 1 表 7 より)  
 C: 文字数指示子のビット数 (附属書 1 表 7 より)  
 D: 入力文字数

**2.3.5 混在モード** データの内容によって、あるモードにおけるデータ列を別のモードへ変更する場合、又は符号化密度を高める場合、この任意機能を用いる。データの各セグメントは、モード指示子、文字数指示子、データ、及び次のセグメントを開始するモード指示子によって続く基本構造で 2.3.1～2.3.4 に示す適切なモードで符号化する。

モード指示子 1	文字数指示子	データ	モード指示子 2	文字数指示子	データ	モード指示子 3	文字数指示子	データ	...
----------	--------	-----	----------	--------	-----	----------	--------	-----	-----

附属書 1 図 5 混在モードの形式

**2.3.6 終端パターン** シンボルのデータの最後は、附属書 1 表 6 に示した終端パターンを型番に応じて付加する。終端パターンは、データビット列がシンボル容量を完全に満たしている場合は省略し、必要な終端パターン数に満たない場合は短縮する。

附属書 1 表 6 終端パターンのビット数及びビット列

型番	終端パターンビット数	終端パターンビット列
M1	3	000
M2	5	00000
M3	7	0000000
M4	9	000000000

**2.3.7 コード語変換に対するビット列** 各モードセグメントに対応するビット列を、順番に接続する。データビット列が完全にシンボル容量を満たしていないときは、終端パターンをビット列に付加する。得られたメッセージのビット列を、コード語に分割する。すべてのコード語は、8 ビット長とする。最後のコード語の長さが 8 ビットでないビット列の場合、最後のコード語に 2 進値 0 の埋め草ビットを付加し、8 ビット長にする。埋め草ビットは、データ列の最終ビット（最下位けた）の後に付加する。メッセージビット列は、埋め草コード語 11101100 及び 00010001 を交互に付加することにより、附属書 1 表 7 で定義する、型番及び誤り訂正レベルに対応するシンボルのデータ容量を満たす長さにする。ただし、M1 型及び M3 型の場合、最後のコード語は 4 ビット長とし、埋め草コード語として使用する場合は 0000 とする。

附属書 1 表 7 全型番のシンボル文字数及び入力データ容量

型番	誤り訂正レベル	データコード語 <sup>(2)</sup> 数	データビット <sup>(2)</sup> 数	データ容量			
				数字	英数字	8 ビットバイト	漢字
M1	誤り検出	3	20	5	-	-	-
M2	L	5	40	10	6	-	-
	M	4	32	8	5	-	-
M3	L	11	84	23	14	9	6
	M	9	68	18	11	7	4
M4	L	16	128	35	21	15	9
	M	14	112	30	18	13	8
	Q	10	80	21	13	9	5

注<sup>(2)</sup> すべてのコード語は 8 ビット長とする。ただし、型番 M1 及び M3 の最後のデータコード語は 4 ビット長とする。

<sup>(2)</sup> データビット数には、モード指示子及び文字数指示子を含む。

**2.3.8 誤り検出及び／又は訂正** マイクロ QR コードは、RS (リードソロモン) 符号によって、誤りの検出及び／又は訂正ができる。RS 符号の誤り訂正コード語をデータコード語列の後に付加する。

附属書 1 表 8 に各型番及び誤り訂正レベルに関し、総コード語数及び誤り訂正コード語数を示す。

附属書 1 表 8 全型番の誤り訂正特性

型番	総コード語数	誤り訂正レベル	誤り訂正コード語数	RS ブロック (4)
M1	5	誤り検出だけ	2	(5, 3, 0) (5)
M2	10	L	5	(10, 5, 1) (5)
		M	6	(10, 4, 2) (5)
M3	17	L	6	(17, 11, 2) (5)
		M	8	(17, 9, 4)
M4	24	L	8	(24, 16, 3) (5)
		M	10	(24, 14, 5)
		Q	14	(24, 10, 7)

注(4)  $(c, k, r)$  :  $c$ =コード語数 [附属書 1 表 1 (E) を参照],  $k$ =データコード語数,  $r$ =誤り訂正数

(5) 誤り訂正数は復号誤りの可能性を低減するために誤り訂正コード語数の半分未満とする。

**2.4 コード語の配置** コード語の配置については QR コードシンボルモデル 2 に準じる。

## 2.5 マスク

**2.5.1 マスク処理** マイクロ QR コードの読出しを確実にするためには、明及び暗のモジュールをバランス良くシンボル内に配列することが望ましい。シンボルの外周を暗モジュールを多くしてシンボルとクワエットゾーンとの区別を付けやすくする。この条件を満たすため、次の手順に従ってマスク処理を適用する。

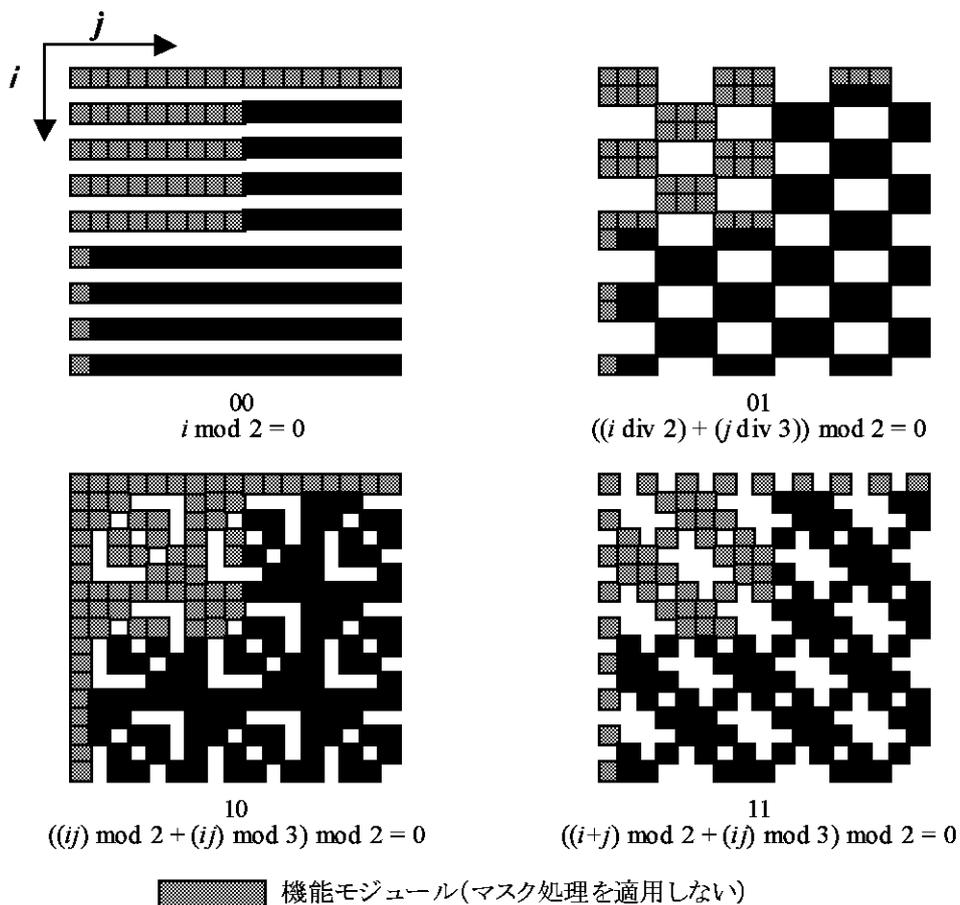
- マスク処理は、機能パターンには適用しない。
- 符号化領域 (形式情報を除く。) で与えられたモジュールパターンと複数のマスク処理マトリックスパターンとで順に XOR 演算による変換をする。XOR 演算とは、各マスク処理マトリックスパターン上にモジュールパターンを重ね合わせ、マスク処理パターンの暗モジュールに対応するモジュールを反転 (明から暗へ、暗から明へ) する。
- それぞれの変換結果において、望ましい項目に対する評価点をすべての変換結果で計算する。
- 評価点の一番高いパターンを選択する。

**2.5.2 マスクパターン** 附属書 1 表 9 は、マスクパターン参照子 (形式情報で使用される 2 進値) 及びマスクパターン生成条件を示す。マスクパターンは、符号化領域 (形式情報を除く。) において条件が真であるモジュールを暗と定義することで、マスクパターンを生成する。条件において、 $i$  はモジュールの行位置を示し、 $j$  は列位置を示す。 $(i, j) = (0, 0)$  はシンボルの左上モジュールを示す。

附属書 1 表 9 マスクパターン生成条件

マスクパターン参照子	条件
00	$i \bmod 2 = 0$
01	$((i \div 2) + (j \div 3)) \bmod 2 = 0$
10	$((ij) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
11	$((i+j) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$

附属書 1 図 6 は、マイクロ QR シンボルに対するすべてのマスクパターンを示す。



備考1. 各パターン下の2ビットがマスクパターン情報となる。

2. 2ビットの下式は、マスクパターン生成条件を示し、式を満たすモジュールが暗モジュールに対応する。

附属書1図6 マイクロQRシンボルに対する全マスクパターン

**2.5.3 マスク処理結果の評価** 各マスクパターンで順次マスク処理を実行した後、次の特徴を計算し評価点を求めることで結果を評価する。評価点が低いほど、結果が受け入れがたい。タイミングパターンでない2辺の一番外側の暗モジュールの数を各辺で求める。暗モジュールが多いほどクワエットゾーンと符号化領域の区別が付き望ましい。各辺の暗モジュールの数をSUM1, SUM2とすると次の評価式で評価点を求める。

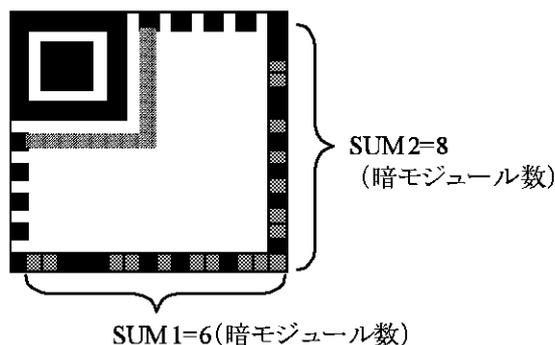
- SUM1  $\leq$  SUM2 の場合

$$\text{評価点} = \text{SUM1} \times 16 + \text{SUM2}$$

- SUM1  $>$  SUM2 の場合

$$\text{評価点} = \text{SUM2} \times 16 + \text{SUM1}$$

評価点が一番高いマスクパターンをシンボルに適應する。



$$\begin{aligned} \text{SUM1} < \text{SUM2}, \text{したがって} \\ \text{評価点} &= \text{SUM1} \times 16 + \text{SUM2} \\ &= 6 \times 16 + 8 \\ &= 104 \end{aligned}$$

附属書 1 図 7 マスク処理の評価点の例

**2.6 形式情報** 形式情報は (15, 5) BCH コードで計算する 10 誤り訂正ビット及び 5 データビットをもつ 15 ビット列とする。形式情報の誤り訂正計算に関する詳細は、附属書 C による。附属書 1 表 10 に示すように、最初の 3 データビットは型番番号である。

附属書 1 表 10 型番番号指示子

型番番号	型番	誤り訂正レベル	2進指示子
0	M1	誤り検出だけ	000
1	M2	L	001
2		M	010
3	M3	L	011
4		M	100
5	M4	L	101
6		M	110
7		Q	111

形式情報の 4 番目及び 5 番目のデータビットは、2.5 に基づき選択されるパターンに対するマスクパターン参照子 (附属書 1 表 9 参照) とする。

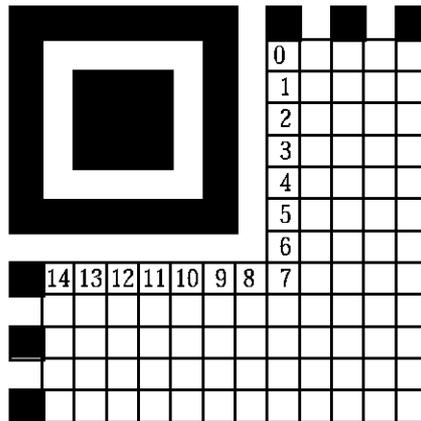
10 誤り訂正ビットは、5. の規定のとおり計算し、5 データビットに付加する。

15 ビットの誤り訂正化した形式情報は、シンボル番号及びマスクパターンとの組合せによってデータ文字列がすべてゼロにならないよう、ビットパターン **100010001000101** とで XOR 演算する。

得られたマスク形式情報は、附属書 1 図 8 に示すシンボルの予約領域に割り当てる。形式情報の最下位けたビットは附属書 1 図 8 のモジュール数 0 に位置し、最上位けたビットはモジュール数 14 に位置する。

#### 例

型番番号 0 :	<b>000</b>
マスクパターン参照子 :	<b>11</b>
データビット (シンボル番号, マスクパターン参照子) :	<b>00011</b>
BCH ビット :	<b>1101011001</b>
マスク前のビット列 :	<b>000111101011001</b>
XOR 演算用マスクパターン :	<b>100010001000101</b>
形式情報モジュール :	<b>100101100011100</b>



附属書 1 図 8 形式情報の配置

3. リードソロモン誤り検出／訂正

3.1 誤り訂正の生成多項式 誤り訂正の生成多項式は、データコード語の多項式の除算に使用する。ここで各コード語は、累乗の降順に除算される多項式の係数となる。この除算の剰余の係数は、誤り訂正コード語の値となる。

附属書 1 表 11 は、各型番及び各誤り訂正レベルで使用する誤り訂正コードに対する生成多項式を示す。同表において、 $\alpha$  は  $GF(2^8)$  上の原始要素 2 の根である。各生成多項式は 1 次多項式  $x-2^0, x-2^1, \dots, x-2^{n-1}$  から生成する。ここに  $n$  は生成多項式の次数とする。

附属書 1 表 11 誤り訂正コードの生成多項式

RS コード語数	生成多項式
2	$x^2 + \alpha^{25}x + \alpha$
5	$x^5 + \alpha^{113}x^4 + \alpha^{164}x^3 + \alpha^{166}x^2 + \alpha^{119}x + \alpha^{10}$
6	$x^6 + \alpha^{116}x^5 + \alpha^{134}x^4 + \alpha^{134}x^3 + \alpha^5x^2 + \alpha^{176}x + \alpha^{15}$
8	$x^8 + \alpha^{175}x^7 + \alpha^{238}x^6 + \alpha^{208}x^5 + \alpha^{249}x^4 + \alpha^{215}x^3 + \alpha^{252}x^2 + \alpha^{196}x + \alpha^{28}$
10	$x^{10} + \alpha^{251}x^9 + \alpha^{67}x^8 + \alpha^{46}x^7 + \alpha^{61}x^6 + \alpha^{118}x^5 + \alpha^{70}x^4 + \alpha^{64}x^3 + \alpha^{94}x^2 + \alpha^{32}x + \alpha^{45}$
14	$x^{14} + \alpha^{199}x^{13} + \alpha^{249}x^{12} + \alpha^{155}x^{11} + \alpha^{48}x^{10} + \alpha^{190}x^9 + \alpha^{124}x^8 + \alpha^{218}x^7 + \alpha^{137}x^6 + \alpha^{216}x^5 + \alpha^{87}x^4 + \alpha^{207}x^3 + \alpha^{59}x^2 + \alpha^{22}x + \alpha^{91}$

4. 誤り訂正復号手順 QR コードに準じる。

5. 形式情報 形式情報は、5 ビットのデータ及び 10 ビットの BCH 誤り訂正からなる 15 ビット列で構成する。ここでは誤り訂正ビットの計算と誤り訂正復号手順について規定する。

5.1 誤り訂正ビット計算 Bose-Chaudhuri-Hocquenghem (15, 5) 符号を誤り訂正として使用する。データビット文字列を係数とする多項式を生成多項式  $G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$  で除算する。剰余多項式の係数文字列をデータビット文字列に付加し、(15, 5) BCH 符号文字列を形成する。最後に、どのようなマスクパターン及び誤り訂正レベルの組合せでも形式情報のビットパターンがすべてゼロにならないように、ビット文字列を 100010001000101 とで XOR 演算し、マスク処理をする。

例

型番番号 0 ; マスクパターン 11

2進文字列：                   **00011**  
 多項式：                        **$x + 1$**   
 (15-5) 乗する：                **$x^{11} + x^{10}$**   
 G(x) で除算する：            **$=(x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1)(x + 1) + (x^9 + x^8 + x^6 + x^4 + x^3 + 1)$**   
 上の多項式の係数文字列を形式情報のデータ文字列に付加する。  
**00011 + 1101011001 → 000111101011001**  
 マスクと XOR 演算           **100010001000101**  
 結果：                         **100101100011100**

2.6 で規定したように、これらのビットを形式情報領域に配置する。

**5.2 誤り訂正復号手順** マスクパターン **100010001000101** でビット列を XOR 演算し、形式情報のマスク処理を解除する。

これによって次のコード語を得る。

$$R = (r_0, r_1, r_2, \dots, r_{14})$$

すなわち、

$$R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{14}x^{14}$$

ここに、 $r_i (i=0 \sim 14) : 0$  又は  $1$

シンδροームを計算する。

シンδροーム  $S_i (i=1, 3, 5)$  を求める。

$$S_1 = R(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{14}\alpha^{14}$$

$$S_3 = R(\alpha^3) = r_0 + r_1\alpha^3 + r_2\alpha^6 + \dots + r_{14}\alpha^{42}$$

$$S_5 = R(\alpha^5) = r_0 + r_1\alpha^5 + r_2\alpha^{10} + \dots + r_{14}\alpha^{70}$$

ここに、 $\alpha$  は  $GF(2^4)$  の原始元とする。

誤り位置を求める。

$$S_1 + \sigma_1 = 0$$

$$S_3 + S_2\sigma_1 + S_1\sigma_2 + S_3 = 0$$

$$S_5 + S_4\sigma_1 + S_3\sigma_2 + S_2\sigma_3 = 0$$

ここに、 $S_2 = (S_1)^2$ ,  $S_4 = (S_2)^2$  である。

上の式を用いてそれぞれの誤り位置変数  $\sigma_i (i=1 \sim 3)$  を求める。

さらに、次の多項式にその変数を代入し、 $GF(2^4)$  の元を順次代入する。

$$\sigma(x) = x^3 + \sigma_1x^2 + \sigma_2x + \sigma_3$$

この結果、 $\sigma(\alpha_j) = 0$  となる元  $\alpha_j$  に対し、 $j$  けた目 ( $0$  けた目から数えて) に誤りがあることがわかる。各誤り位置のビット値を反転し、誤りを訂正する。

**関連規格** JIS X 0211:1994 符号化文字集合用制御機能

ISO/IEC 6429:1992 Information technology—Control functions for coded character sets

CEN EN 1635 Bar Coding—Test Specifications for Bar Code Symbols

ANSI X 3.4 Coded Character Sets—7-bit American National Standard Code for Information Interchange (7-bit ASCII)

## 附属書 2 (参考) JIS と対応する国際規格との対比表

JIS X 0510 : 2004 二次元コードシンボル—QR コード—基本仕様		ISO/IEC 18004 : 2000 情報技術—自動認識及びデータ取得技術—バーコードシンボル体系—QR コード					
(I) JIS の規定		(II) 国際規格番号	(III) 国際規格の規定		(IV) JIS と国際規格との技術的差異の項目ごとの評価及びその内容 表示箇所：本文の左側 表示方法：傍線		(V) JIS と国際規格との技術的差異の理由及び今後の対策
項目番号	内容		項目番号	内容	項目ごとの評価	技術的差異の内容	
1.	適用範囲 QR コードとして知られる二次元シンボル体系の技術的要件を規定。	ISO/IEC 18004	1	JIS とほぼ同じ。	MOD/追加	JIS は、マイクロ QR コードの規定を適用範囲に追加。	ISO 規格改正時に、マイクロ QR コードの追加を提案する。
2.	適合条件 この規格への合致条件について規定。		2	JIS と同じ。	IDT	—	—
3.	引用規格 JIS X 0201, JIS X 0208, JIS X 0530, EN 1556, AIM Inc. Technical Specification		3	JIS X 0201, JIS X 0208, ISO/IEC 15424, EN 1556, AIM Inc. Technical Specification	IDT	—	—
	JIS X 0221-1, JIS X 0520			—	MOD/追加		JIS として必要な引用規格を追加。
	—			ANSI X 3.4	MOD/削除		JIS として不要な引用規格を削除。
4.	定義 26 の用語を定義	4	22 の用語を定義	MOD/追加	4 つの用語の定義を追加。	JIS として必要な用語を追加。	
5.	記号 公式及び数式で使用する記号について規定。	5	JIS と同じ。	IDT	—	—	

(I) JIS の規定		(II) 国際規格番号	(III) 国際規格の規定		(IV) JIS と国際規格との技術的差異の項目ごとの評価及びその内容 表示箇所：本文の左側 表示方法：傍線		(V) JIS と国際規格との技術的差異の理由及び今後の対策
項目番号	内容		項目番号	内容	項目ごとの評価	技術的差異の内容	
6.	記法 モジュール位置、バイト表記及び型番参照について規定。		6	<b>JIS</b> と同じ。	IDT	—	—
7.	QR コードの仕様 QR コードの基本的特性、追加機能のまとめ及びシンボルの構造（9 項目）について規定。		7	<b>JIS</b> とほぼ同じ。	MOD/追加	<b>JIS</b> は、シンボルの構造として、“明暗反転”と“表裏反転”とを追加。	—
8.	要求事項 要求事項 9 項目を規定。		4	<b>JIS</b> とほぼ同じ。	MOD/追加	8.4（データの符号化）で規定している表 2（モード指示子）に、未使用領域の予約用の指示子を追加。	
					MOD/変更	8.4.4（8 ビットバイトモード）の符号化文字集合に関する記述を変更。	
9.	接続 接続の基本原理、シンボル列指示子及びパリティデータについて規定。		9	<b>JIS</b> と同じ。	IDT	—	—
10.	シンボルの印刷及びマーキング 寸法、文字による表記及びマーキングの手引きについて規定。		10	<b>JIS</b> と同じ。	IDT	—	—
11.	シンボルの品質 QR コードシンボルの品質評価について規定。		11	<b>JIS</b> と同じ。	IDT	—	—

(I) JIS の規定		(II) 国際規格番号	(III) 国際規格の規定		(IV) JIS と国際規格との技術的差異の項目ごとの評価及びその内容 表示箇所：本文の左側 表示方法：傍線		(V) JIS と国際規格との技術的差異の理由及び今後の対策	
項目番号	内容		項目番号	内容	項目ごとの評価	技術的差異の内容		
12.	復号手順の概要 QR コードシンボルの読取りからデータ文字出力に至る復号手順の概要について規定。		12	JIS と同じ。	IDT	—	—	
13.	QR コードの参照復号アルゴリズム		13	JIS と同じ。	IDT	—	—	
14.	自動識別能力		14	JIS と同じ。	IDT	—	—	
15.	送信データ		15	JIS と同じ。	IDT	—	—	
附属書	A (規定)		附属書	A (規定)	IDT	IDT	—	—
	B (規定)			B (規定)	IDT	IDT	—	—
	C (規定)			C (規定)	IDT	IDT	—	—
	D (規定)			D (規定)	MOD/変更	JIS は、D.2 (誤り訂正復号手順) の記述を変更。	記述を効率的、かつ、簡易な手順に変更。	
	E (規定)			E (規定)	IDT	—	—	
	F (規定)			F (規定)	IDT	—	—	
	G (参考)			G (参考)	MOD/変更	手順 4 及び手順 5 を変更。	ISO/IEC 規格の誤記を訂正。	
	H (参考)			H (参考)	IDT	—	—	
	I (参考)			I (参考)	IDT	—	—	
	J (参考)			J (参考)	IDT	—	—	
	K (規定)			K (参考)	IDT	—	—	
L (参考)	L (参考)	IDT	—	—				
M (規定)	M (参考)	IDT	—	—				
附属書	1 (規定) マイクロ QR コード	—	—	MOD/追加	マイクロ QR コードの仕様、要求事項などを規定。	ISO/IEC 規格改正時に、追加を提案する。		

**JIS と国際規格との対応の程度の全体評価：MOD**

**備考1.** 項目ごとの評価欄の記号の意味は、次のとおりである。

- IDT…………… 技術的差異がない。
- MOD/削除…………… 国際規格の規定項目又は規定内容を削除している。
- MOD/追加…………… 国際規格にない規定項目又は規定内容を追加している。
- MOD/変更…………… 国際規格の規定内容を変更している。

**2. JIS と国際規格との対応の程度の全体評価欄の記号の意味は、次のとおりである。**

- MOD…………… 国際規格を修正している。