PDF417

Specification for Barcode Symbology

1 Introduction

PDF417 is a multi-row, variable-length symbology offering high data capacity and error-correction capability. PDF417 can be scanned by linear scanners, rastering laser scanners, or two-dimensional imaging devices. One PDF417 symbol is capable of encoding more than 1100 bytes, 1800 ASCII characters, or 2700 digits, depending on the selected data compaction mode.

Every PDF417 symbol is composed of a stack of rows, from a minimum of 3 to a maximum of 90 rows. Each PDF417 row contains start and stop patterns, left and right row indicators, and from one to thirty data symbol characters. Since both the number of rows and their length are selectable when printed, the aspect ratio of a PDF417 symbol can be varied to suit spatial requirements for printing.

A PDF417 symbol character consists of seventeen modules arranged into four bars and four spaces. The entire set of symbol characters is divided into three mutually exclusive encodation sets, or "clusters." Each cluster encodes the 929 available PDF417 symbol character values, or codewords, with distinct bar-space patterns so that one cluster cannot be confused with another. Because any two adjacent rows use different clusters, the decoder can utilize data from scans that cross rows while decoding a PDF417 symbol.

By adding error correction codewords to the data message, PDF417 supports correction of lost or missing data. Each PDF417 symbol requires two error correction codewords for error detection. In addition, up to 510 codewords of error correction can be added when printing the symbol. This allows for decoding security that mathematically is many orders of magnitude stronger than that of bar codes with simple check characters.

PDF417 offers three data compaction modes. Each mode defines a conversion or mapping between codeword sequences and byte sequences. The three modes are Text Compaction mode, Byte Compaction mode, and Numeric Compaction mode.

The interpretation of the byte sequences encoded by a compaction mode is determined by the Global Label Identifier (GLI). A GLI is a special codeword sequence which activates a set of interpretations. The implementation of GLIs enables PDF417 to encode international character sets, and industry- and user-defined character sets, as well as ASCII.

Macro PDF417 provides a means for creating a distributed representation of files too large to be represented by a single PDF417 symbol. Macro PDF417 symbols differ from ordinary PDF417 symbols in that they contain additional control information used to support this distributed representation. This allows a decoder to make use of this information to correctly reconstruct and verify the file. independent of the symbol scanning order.

In a relatively clean environment in which damage to the label is unlikely, Truncated PDF417 can be used. This version omits the right row indicator and simplifies the stop pattern into a single module width bar. This reduces the non-data overhead, with some trade-off in robustness, or the ability to withstand degradation. Truncated PDF417 is fully reader-compatible with standard PDF417.

Table 1 summarizes PDF417's characteristics.

Encodable Character Set	All 128 ASCII Characters
	All 128 Extended ASCII Characters
	8-Bit Binary Data
	Up to 811,800 Different Character Sets and/or Interpretations
Code Type	Continuous, Multi-Row
Character Self Checking	Yes
Symbol Size:	
Height	Variable (3 to 90 Rows)
Width	Variable (90X to 583X)
Bidirectional Decoding	Yes
Error Correction Characters	2 to 512
Maximum Data Characters per Symbol	1850 text characters
(at error correction level 0)	2710 digits
	1108 bytes
Additional Features	Selectable Levels of Error Correction
	Can Utilize Scans that Cross Rows
Additional Options	Macro PDF417
	Global Label Identifiers
	Truncated PDF417

Table 1: Characteristics of PDF417

2 Symbol Description

2.1 Symbol Structure

Each PDF417 symbol consists of a stack of vertically aligned rows framed by clear areas called quiet zones.

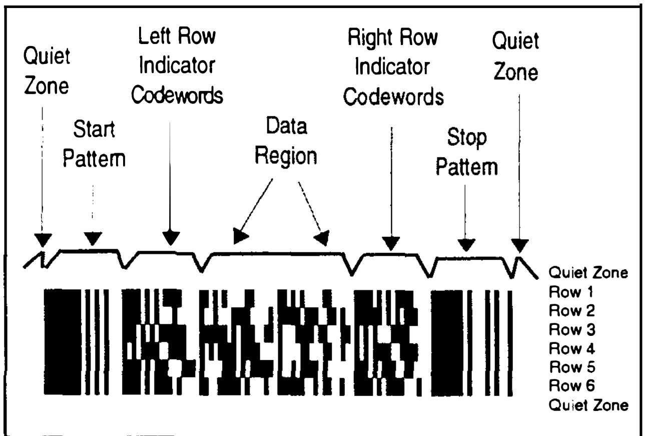

Figure 1: PDF417 Symbol Structure

Figure 1 shows a typical PDF417 symbol and its structure.

Every PDF417 symbol contains a minimum of 3 to a maximum of 90 rows. Each row consists of:

a. Leading quiet zone
b. Start pattern
c. Left row indicator symbol character
d One to thirty data symbol characters
e. Right row indicator symbol character
f. Stop pattern.
g. Trailing quiet zone

A symbol character consists of seventeen modules arranged into four bars and four spaces. Each symbol character represents a value in the range of 0 to 928; within this document, these symbol character values are referred to as "codewords" (not to be confused with the term codeword as used in error correction literature).

Because the number of rows is variable, and rows are variable in length (i.e., in the number of symbol character "columns"), the height/width proportion, or aspect ratio, of a PDF417 symbol can be varied to suit spatial requirements for printing. However, the number of symbol characters in all rows of a given symbol must be the same.

The data region of a PDF417 symbol is the central area of codeword columns between the left row indicator column and the right row indicator column. The first (upper left) codeword of the data region is the symbol length descriptor. Its value indicates the total number of codewords in the data region, including the symbol length descriptor itself, but excluding the error correction codewords. The remaining codewords in the data region (including the data codewords, pad codewords, and error correction codewords, in that order) are arranged with the most significant codeword adjacent to the symbol length descriptor, and are read from left to right, top row to bottom. The total number of codewords in the data region of a single PDF417 symbol cannot exceed 928.

2.2 Symbol Character Encodation

2.2.1 Symbol Character Structure

A symbol character consists of four bars and four spaces; each bar or space contains one to six modules. In all cases, the four bars and four spaces of any symbol character measure 17 modules in total. The width of one module is the X dimension of that symbol.

2.2.2 Clusters and Symbol Character Definitions

The entire set of PDF417 symbol characters is divided into three mutually exclusive encodation sets, or "clusters." Each cluster encodes all 929 defined PDF417 codewords with distinct bar and space patterns. Within each cluster, each symbol character is associated with a unique value in the range of 0 to 928; this value is called the symbol character value or codeword. Table H1 in Appendix H presents, for each cluster, the symbol character bar and space sequences and their corresponding symbol character values.

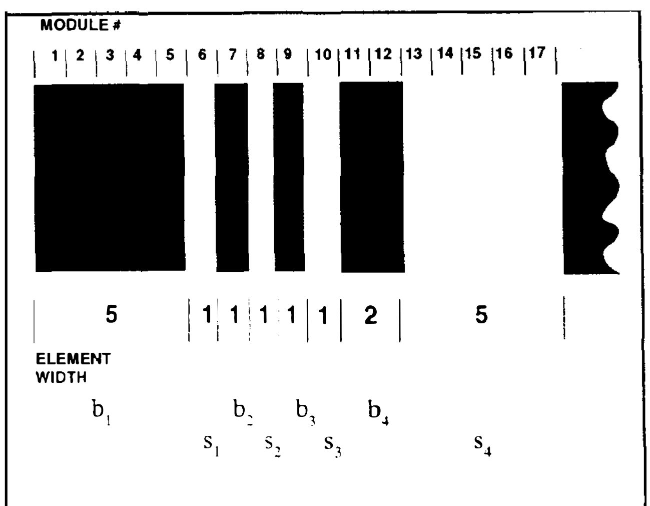

Figure 2: Symbol Character Structure

Referring to Figure 2, the cluster number of a particular symbol character is defined in the following formula:

$$\text{cluster number} = (b_1 - b_2 + b_3 - b_4 + 9) \bmod 9$$

where b_1, b_2, b_3, and b_4 represent bar widths.

For the character in Figure 2 the cluster number calculation is:

$$\text{cluster number} = (5 - 1 + 1 - 2 + 9) \bmod 9 = 3$$

PDF417 uses only cluster numbers 0, 3, and 6. This cluster definition applies to all PDF417 symbol characters. The start and stop patterns, which are described in Section 2.3, do not have symbol character values.

2.2.3 Row Encoding

A PDF417 symbol may have as few as 3 or as many as 90 rows. To encode data, each row uses the symbol characters from only one cluster. The same cluster usage repeats every third row, in sequence. Row 1 uses cluster 0 symbol characters, row 2 uses cluster 3 symbol characters, row 3 uses cluster 6 symbol characters, row 4 uses cluster 0 symbol characters and so forth, in a repeating sequence:

$$\text{cluster number} = ((\text{row number} - 1) \bmod 3) * 3$$

Because any two adjacent rows use different clusters, the decoder can utilize scans that cross rows while decoding a PDF417 symbol. See Reference Decode Algorithm, Section 4.

Figure 3 presents the general PDF417 row structure. Row i begins with a left-row indicator (L_i), followed by data region characters (the most significant adjacent to the left-row indicator), then a right-row indicator (R_i).

Row indicators are symbol characters, adjacent to the start and stop patterns. whose values indicate the row number (i), the number of rows (3 to 90), the number of columns in the data region (1 to 30), and the error correction level (0 to 8).

The symbol character value of the left row indicator (L_i) is given by:

$$L_i = \begin{cases} 30\,x_i + y & \text{if } c_i = 0 \\ 30\,x_i + z & \text{if } c_i = 3 \\ 30\,x_i + v & \text{if } c_i = 6 \end{cases}$$

The symbol character value of the right row indicator (R_i) is given by:

$$R_i = \begin{cases} 30\,x_i + v & \text{if } c_i = 0 \\ 30\,x_i + y & \text{if } c_i = 3 \\ 30\,x_i + z & \text{if } c_i = 6 \end{cases}$$

Where:

x_i = (row number -1) div 3 for i = 1 to 90
y = (number of rows - 1) div 3
z = (error correction level) $*$ 3 + (number of rows - 1) mod 3
v = number of columns in the data region - 1
c_i = cluster number of the i th row

[div is the integer division operator, rounding down, and mod is the remainder after division.]

Figure 3 depicts this arrangement. For example, if a symbol has 3 rows, 3 columns, and error correction level 1, the (L_1, L_2, L_3) and (R_1, R_2, R_3) are (0, 5, 2) and (2, 0, 5) respectively.

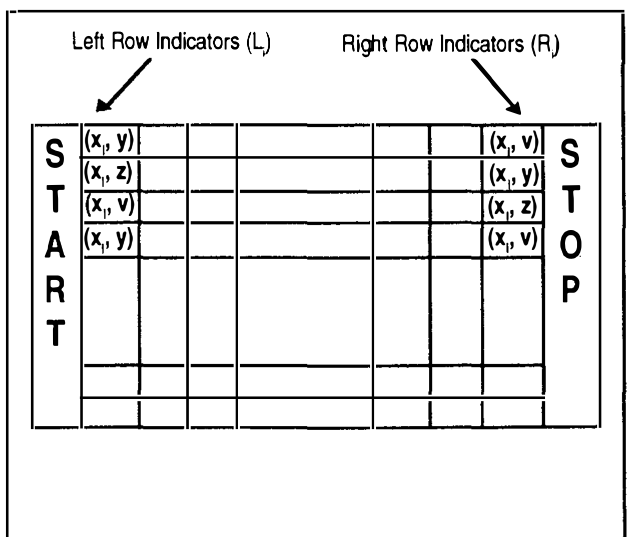

Figure 3: Left/Right Row Indicators

2.2. 4 Mode Structure

2.2.4.1 Mode Definitions

The data compaction mode structure is designed for efficiently packing a stream of bytes into a stream of codewords.

PDF417 has three data compaction modes. These are the Text Compaction (TC) mode, the Byte Compaction (BC) mode, and the Numeric Compaction (NC) mode. Through use of mode latch/shift identifiers, these modes can be intermixed within a symbol.

2.2.4.2 Codeword Set

PDF417 contains 929 defined codewords within its codeword set. A codeword (i.e., symbol character value) ranging from 0 to 928 is assigned to each symbol character in each of the three clusters.

Within the codeword set, codeword use is governed by the following rules:

> Symbol character values 0 - 899: These encode data according to the current compaction mode and GLI interpretation.
>
> Symbol character values 900 - 928: In each mode, these are special-purpose symbol characters as follows:
>
> > Symbol character values 900, 901, 902, 913, 924 are used as mode identifiers (see Table 2).
> >
> > Symbol character values 925, 926, 927 are used for Global Label Identifiers (GLIs)(see Section 2.2.5).
> >
> > Symbol character values 922, 923, 928 are used for Macro PDF417 (see Appendix G).
> >
> > Symbol character value 921 is used for reader initialization (see Section 2.2.7).
> >
> > Symbol character values 903 - 912, 914 - 920 are reserved for future use.

Mode	Mode Latch	Mode Shift
Text Compaction	900	
Byte Compaction	901/ 924*	913
Numeric Compaction	902	

* See Section 2.2.4.5

Table 2: Mode Definition

2.2.4.3 Mode Latch and Mode Shift Codewords

For switching among modes, there are Mode Latch and Mode Shift codewords, as defined in Table 2.

A Mode Latch codeword switches from the current mode to the indicated destination mode. This mode switch stays in effect until another mode switch is explicitly indicated.

The Mode Shift codeword 913 causes a temporary switch from Text Compaction mode to Byte Compaction mode. This switch is in effect only for the next codeword, after which the mode reverts the current sub-mode of the Text Compaction mode.

Note that a Mode Latch may be to any mode, including the current mode, whereas a mode shift to the current mode is invalid (i.e. 913 from within Byte Compaction).

Figure 4 depicts the mode switching structure.

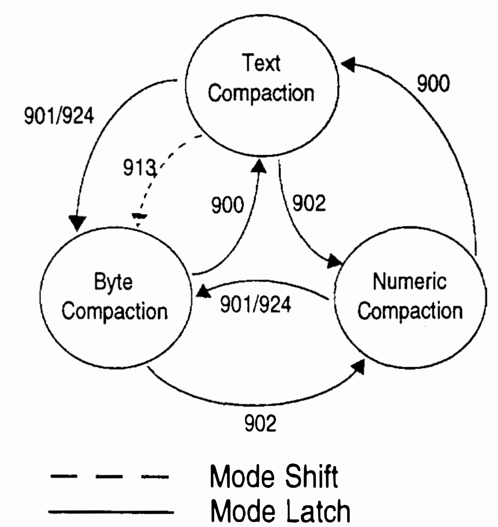

Figure 4: Available Mode Switching

2.2.4.4 Text Compaction Mode (TC)

2.2.4.4.1 Sub-Modes

The Text Compaction mode contains four sub-modes:

- Alpha
- Lower Case
- Mixed
- Punctuation

Value	--Alpha-- ASCII Value	GLI 0 Char	--Lower-- ASCII Value	GLI 0 Char	--Mixed-- ASCII Value	GLI 0 Char	-Punctuation- ASCII Value	GLI 0 Char
0	65	A	97	a	48	0	59	;
1	66	B	98	b	49	1	60	<
2	67	C	99	c	50	2	62	>
3	68	D	100	d	51	3	64	@
4	69	E	101	e	52	4	91	[
5	70	F	102	f	53	5	92	\
6	71	G	103	g	54	6	93	]
7	72	H	104	h	55	7	95	_
8	73	I	105	i	56	8	96	`
9	74	J	106	j	57	9	126	~
10	75	K	107	k	38	&	33	!
11	76	L	108	l	13	CR	13	CR
12	77	M	109	m	09	HT	09	HT
13	78	N	110	n	44	,	44	,
14	79	O	111	o	58	:	58	:
15	80	P	112	p	35	#	10	LF
16	81	Q	113	q	45	-	45	-
17	82	R	114	r	46	.	46	.
18	83	S	115	s	36	$	36	$
19	84	T	116	t	47	/	47	/
20	85	U	117	u	43	+	34	"
21	86	V	118	v	37	%	124	|
22	87	W	119	w	42	*	42	*
23	88	X	120	x	61	=	40	(
24	89	Y	121	y	94	^	41	)
25	90	Z	122	z	pl		63	?
26	32	SP	32	SP	32	SP	123	{
27	ll		as		ll		125	}
28	ml		ml		al		39	'
29	ps		ps		ps		al	

Note: Except for sub-mode latch/shifts (ll, ml, pl, ps, as, al; see Figure 5) decimal ASCII values and their equivalent default character interpretations are listed in this table.

Table 3: Text Compaction Sub-Mode Definition with Mapped Values and ASCII Interpretation in GLI 0

This sub-mode structure allows for efficient data compaction with a selected set of characters which frequently occur in typical texts to be encoded. Each of these characters is assigned a value between 0 and 29 in one or more of the sub-modes; this allows two characters to be encoded in a single symbol character, as shown in Table 3. The symbol character value which encodes the pair is given by:

symbol character value = $30*H + L$.

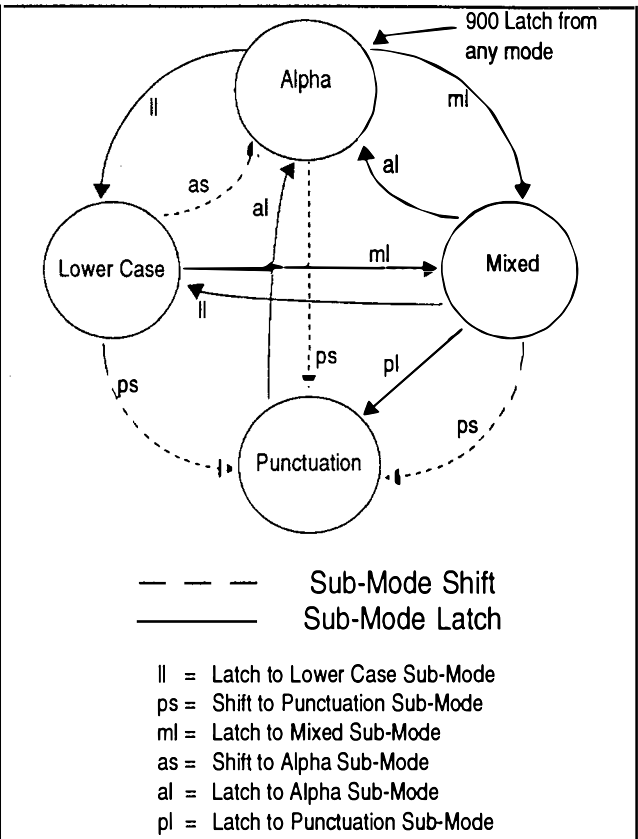

Figure 5: Text Compaction Sub-Mode Switching

where H is the value corresponding to the leading character of the pair and L is the value corresponding to the trailing character.

Table 3 defines the Text Compaction sub-mode structure.

Because there are limits to the character set within the sub-modes, a full set of sub-mode latch and shift indicators is not available for each sub-mode. For instance, Alpha sub-mode has a Lower Latch (ll), Mixed Latch (ml), and Punctuation Shift (ps), but there is no Lower Case Shift (ls), Mixed Shift (ms), or Punctuation Latch (pl).

2.2.4.4.2 Sub-Mode to Sub-Mode Switching

Figure 5 depicts the sub-mode switching structure.

A latch from any mode to the Text Compaction mode is a latch to the Alpha sub-mode. In Text Compaction mode, each codeword encodes two base-30 values (range 0 - 29). If at the end of a string of characters an odd number of base-30 values have been used, a pad value of 29 is needed to complete the last codeword. If ps is used as a pad immediately preceding a Byte shift (codeword 913), then the ps is ignored. Following a sub-mode shift another sub-mode shift or latch is prohibited.

2.2.4.5 Byte Compaction Mode (BC)

The Byte Compaction mode encodes a sequence of bytes into a sequence of codewords through base 256 to base 900 conversion.

For the Byte Compaction mode, there are two Mode Latches (901, 924). Mode Latch 901 is used when the total number of bytes to be encoded is not a multiple of 6. Mode Latch 924 is used when the total number of bytes to be encoded is an integer multiple of 6.

In cases using Mode Latch 924, 6 bytes are encoded into 5 codewords by using a base 256 to base 900 transformation, from left to right, the most to least significant digits. For example, a sequence of binary data (01H, 02H, 03H, 04H, 05H, 06H), where H stands for hexadecimal notation, is encoded into a codeword sequence (924, 1, 620, 89, 74, 846). Because there are six data units, the first codeword is a Byte Compaction Mode Latch for a multiple of six, which is 924. The following five codewords encode the six data bytes which satisfy the basis transformation equation:

$$1*256^5 + 2*256^4 + 3*256^3 + 4*256^2 + 5*256 + 6 =$$
$$1*900^4 + 620*900^3 + 89*900^2 + 74*900 + 846$$

A somewhat different process occurs when the number of encoded bytes is not an integer multiple of 6. Except for the remaining portion of bytes not divisible by 6, cases using Mode Latch 901 work the same as in the multiple of 6 example above, encoding 6 bytes in 5 codewords. For example, for a data sequence that is a length of 9 (01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 04H), the encoding result will be (901, 1, 620, 89, 74, 846, 7, 8, 4).

The first codeword, 901, is the Byte Compaction Mode Latch for a data size not a multiple of six. The first six bytes are encoded in the same manner as above, using base 256 to base 900 transformation; the remaining data (07H, 08H, 04H) are encoded directly, 1 byte per codeword.

Mode Shift 913 is used for a temporary shift from Text Compaction mode to Byte Compaction mode.

A Mode Latch (900 or 902) can be used to exit the Byte Compaction mode.

2.2.4.6 Numeric Compaction Mode (NC)

The Numeric Compaction mode is a method for base 10 to base 900 data compaction. In GLI 0, Numeric Compaction is dedicated to the compaction of numeric digits. The byte value mapping is defined as in Table 4.

Digit	ASCII Value	GLI 0 Character
0	48	0
1	49	1
2	50	2
3	51	3
4	52	4
5	53	5
6	54	6
7	55	7
8	56	8
9	57	9

Table 4: Byte Value Mapping in the Numeric Compaction Mode

The Numeric Compaction mode can pack almost 3 digits (2.93) into a symbol character. Though Numeric Compaction mode can be invoked at any digit length, it is recommended to use Numeric Compaction mode when there are more than 13 consecutive digits. Otherwise, use Text Compaction mode.

The digits are encoded according to the following algorithm:

1. Partition the string of digits into groups of 44 digits, except for the last group, which may contain fewer.
2. For each group:
   2a. Prefix the digit sequence with a leading 1.
   2b. Perform a base 10 to base 900 conversion.

For example, to encode 000213298174000, first subdivide it into groups. There are only 15 digits in this case, so there is only one group. Next, add a 1 as the left most digit. This gives 1000213298174000. Finally, convert to base 900, which results in the codeword sequence (1, 624, 434, 632, 282, 200).

The decode algorithm is the reverse of the encode algorithm:

1. Get groups of 15 codewords (15 codewords translate into 44 digits), except for the last group, which may contain fewer.
2. For each group:
   2a. Perform a base 900 to base 10 conversion.
   2b. Strip the leading 1.

Following the example above, there are only 6 codewords, hence only one group. The conversion yields $1*900^5 + 624*900^4 + 434*900^3 + 632*900^2 + 282*900 + 200 = 1000213298174000$. Removing the leftmost 1 leaves 000213298174000, the original number.

A mode latch (900, 901, or 924) can be used to exit the Numeric Compaction mode.

2.2.5 High-Level Encoding

The first codeword within the data region is the symbol length descriptor. The mode structure applies from the second codeword onward. The Alpha sub-mode of the Text Compaction mode, and GLI 0 interpretation, are always in effect at the start of each symbol. Within the symbol, the mode may be changed by Mode Latch or Shift codewords, as explained earlier and the GLI may be changed, as explained in Section 2.2.6.

In the Text Compaction mode, each symbol character encodes two base 30 values from Table 3 as follows:

$$\text{symbol character value} = H * 30 + L$$

For example, a string "Ad:102" can be encoded as a sequence of characters (A, ll, d, ml, :, 1, 0, 2), in which ll and ml correspond to Lower Latch and Mixed Latch respectively.

From Table 3, the values corresponding to the characters are (0, 27, 3, 28, 14, 1, 0, 2). These values are paired into ((0, 27), (3, 28), (14, 1), (0, 2)). According to the equation above, the symbol character values can be calculated as follows:

$$(0 \cdot 30 + 27, 3 \cdot 30 + 28, 14 \cdot 30 + 1, 0 \cdot 30 + 2) = (27, 118, 421, 2)$$

As a result, six characters are packed into four codewords by using a sub-mode switching mechanism.

A given string of data characters may be represented by different codeword sequences, through the use of different combinations of latches and shifts. The following illustrates one sequence using a shift from Lower Case sub-mode to Byte Compaction mode and back to Lower Case sub-mode, and for comparison a second sequence using Byte Compaction mode only:

Input (a four-character ASCII string): <j> <ACK> <p> <q>

Output (Sequence 1): (<ll><j>) (<913>) (<ACK>) (<p><q>)
(Sequence 2): (<901>) (<j>) (<ACK>) (<p>) (<q>)

The corresponding symbol character values are (Sequence 1): (819, 913, 6, 466) and (Sequence 2): (901, 106, 6, 112, 113).

There may not always be exactly the right number of symbol characters to fill up the symbol. Codeword 900 should be used to fill up, or "pad" the remaining codewords. Any pads must appear before the optional Macro PDF417 Control Block and before the error correction codewords.

2.2.6 Global Label Identifier

The Global Label Identifier (GLI) is a special codeword sequence which activates a set of interpretations assigning meaning to the stream of bytes encoded by the data compaction modes. The interpretations remain in effect until the next GLI or the end of the symbol.

2.2.6.1 GLI Codeword Sequence and GLI Value

There are three different GLI codeword sequences led by the codewords 927, 926, and 925. The sequences are the following:

927, G1
or
926, G2, G3
or
925, G4

where G1, G2, G3, and G4 are codewords which range from 0 to 899. The GLI value is defined as:

$$GLI = G1$$
or
$$GLI = (G2 + 1) * 900 + G3$$
or

GLI = 810,900 + G4

There are 811,800 possible GLI values available.

2.2.6.2 GLI Usage

The three Global Label Identifier codeword sequences defined above correspond to the following conventions for GLI usage:

International Character Sets:

> Codeword 927 followed by a single codeword with a value ranging from 0 to 899. The GLI value of 0 is the default interpretation (see Section 2.2.6.3).

General Purpose GLIs:

> Codeword 926 followed by two codewords representing GLI values from 900 to 810,899.

User-Defined GLIs:

> Codeword 925 followed by one codeword (GLI values from 810,900 to 811,799). These GLIs should be used for closed-system applications.

Each GLI has its corresponding interpretation. The interpretation need not always be a one-to-one mapping between byte values and characters. For example, if a specific GLI value were to be assigned to Chinese characters, then the interpretation might map pairs of byte values to those characters. Even more complex interpretations, such as data encryption, could be defined via the GLI mechanism.

It is permissible to use multiple GLIs within one symbol.

AIM® USA plans to publish a separate document, *Global Label Identifier (GLI) Assignments*, for the allocation, assignment, and maintenance of the character set GLIs and the general purpose GLIs.

2.2.6.3 GLI 0 - The Default GLI

The default GLI value is zero. No GLI codeword sequence is required to activate this default interpretation in the beginning of the symbol. If the current GLI value is not zero, and the GLI 0 interpretation is desired, then the codeword sequence 927 0 is used to return to the default interpretation.

When the GLI value is zero, the interpretation of the 8-bit character values is defined by Table 5. The first half of the table is the same as the ASCII character set; the second half of the table is the same as the Extended Character set of the PC 437 English table.

2.2.6.4 GLI 1 - ISO 8859-1

GLI 1 is assigned to ISO 8859-1 which is the 8-bit character set used by other AIM® USA standard bar code symbologies.

2.2.7 Reader Initialization Symbols

The codeword 921 instructs the reader to interpret the data contained within the symbol for reader initialization or programming. The codeword 921 must appear as the first codeword after the symbol length descriptor. In the case of a Macro PDF417 initialization sequence, the codeword 921 must appear in every symbol.

The data contained in an initialization symbol, or sequence of symbols, will not be transmitted by the reader.

2.3 Start and Stop Patterns

Unique start and stop patterns are defined for PDF417. The start pattern is defined as the bar-space sequence of 81111113, while the stop pattern is defined as the nine element pattern 711311121 which ends with a bar.

2.4 Quiet Zones

The quiet zones are areas free of all printing surrounding the symbol, preceding the start symbol character, following the stop symbol character, above the first row, and below the bottom row.

All quiet zones extend a minimum of 2X from the symbol.

2.5 Symbol Dimensions: Width and Height

PDF417 may be printed at various densities to accommodate a variety of printing and scanning processes. The significant dimensional parameter is X, the nominal width of each narrow element. The X dimension must be constant throughout a given symbol.

To calculate the width and height of a PDF417 symbol to be printed. first encode the data message

Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.
0	NUL	37	%	74	J	111	o	148	ö	185	╣	222	▐
1	SOH	38	&	75	K	112	p	149	ò	186	║	223	▀
2	STX	39	’	76	L	113	q	150	û	187	╗	224	α
3	ETX	40	(	77	M	114	r	151	ù	188	╝	225	β
4	EOT	41	)	78	N	115	s	152	ÿ	189	╜	226	Γ
5	ENQ	42	*	79	O	116	t	153	Ö	190	╛	227	π
6	ACK	43	+	80	P	117	u	154	Ü	191	┐	228	Σ
7	BEL	44	,	81	Q	118	v	155	¢	192	└	229	σ
8	BS	45	-	82	R	119	w	156	£	193	┴	230	µ
9	HT	46	.	83	S	120	x	157	¥	194	┬	231	τ
10	LF	47	/	84	T	121	y	158	Pt	195	├	232	Φ
11	VT	48	0	85	U	122	z	159	ƒ	196	─	233	Θ
12	FF	49	1	86	V	123	{	160	á	197	┼	234	Ω
13	CR	50	2	87	W	124	|	161	í	198	╞	235	δ
14	SO	51	3	89	X	125	}	162	ó	199	╟	236	∞
15	SI	52	4	89	Y	126	~	163	ú	200	╚	237	ø
16	DLE	53	5	90	Z	127	DEL	164	ñ	201	╔	238	∈
17	DC1	54	6	91	[	128	Ç	165	Ñ	202	╩	239	∩
18	DC2	55	7	92	\	129	ü	166	ª	203	╦	240	≡
19	DC3	56	8	93	]	130	é	167	º	204	╠	241	±
20	DC4	57	9	94	^	131	â	168	¿	205	═	242	≥
21	NAK	58	:	95	_	132	ä	169	⌐	206	╬	243	≤
22	SYN	59	;	96	`	133	à	170	¬	207	╧	244	⌠
23	ETB	60	<	97	a	134	å	171	1/2	208	╨	245	⌡
24	CAN	61	=	98	b	135	ç	172	1/4	209	╤	246	÷
25	EM	62	>	99	c	136	ê	173	¡	210	╥	247	≈
26	SUB	63	?	100	d	137	ë	174	«	211	╙	248	°
27	ESC	64	@	101	e	138	è	175	»	212	╘	249	•
28	FS	65	A	102	f	139	ï	176	░	213	╒	250	·
29	GS	66	B	103	g	140	î	177	▒	214	╓	251	√
30	RS	67	C	104	h	141	ì	178	▓	215	╫	252	^{n}
31	US	68	D	105	i	142	Ä	179	│	216	╪	253	^{2}
32	SP	69	E	106	j	143	Å	180	┤	217	┘	254	■
33	!	70	F	107	k	144	É	181	╡	218	┌	255	
34	"	71	G	108	l	145	æ	182	╢	219	█		
35	#	72	H	109	m	146	Æ	183	╖	220	▄		
36	$	73	I	110	n	147	ô	184	╕	221	▌		

Table 5: Decimal Values of GLI 0 Character Set

into a sequence of data codewords. Having thus found the number of data codewords required, and further having selected a suitable level of error correction, the minimum number of required codewords is now known. The next step is to select the number of columns in the data region, C, which will determine the number of rows the symbol will have.

The width and height of the resulting symbol may be calculated as:

$$W = (17C + 69)X + 2Q$$

$$H = RY + 2Q$$

where:

H = Height of symbol
W = Width of symbol
C = Number of columns in the data region
R = Number of rows
X = X-dimension of the symbol
Y = Row height
Q = Size of Quiet Zone (minimum 2X)

With at least the recommended minimum level of error correction, the recommended minimum row height is 3X. With less than the minimum recommended level. the recommended minimum row height is 4X. The minimum recommended error correction levels are shown in Table 7.

2.6 Error Detection and Correction

Each PDF417 symbol contains at least two error correction codewords. The Error Correction codewords provide capability for both error detection and correction.

2.6.1 Error Correction Level

The error correction level for a PDF417 symbol is selectable at the time of symbol creation. Table 6 shows the number of error correction codewords for each error correction level.

Error Correction Level	Number of Error Correction Codewords
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256
8	512

Table 6: Error Correction Level

2.6.2 Error Correction Capacity

The error correction level is concerned with two types of errors, rejection errors (erasures) and substitution errors (errors). An erasure is a missing, unscanned or undecodable symbol character; the symbol character's position is known but not its value. An error is a misdecoded or mislocated symbol character; both the position and value of the symbol character are unknown.

Certain values must be defined to express the correcting power of a specified error correction level:

e = Number of erasures
t = Number of errors
s = Error correction level
d = Number of error correction codewords = 2^{s+1}

In these terms the error correction capacity of a given error correction level is as follows:

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

2^{s+1} is the total number of error correction codewords. Two of these are needed for error detection. Any additional error correction codewords provide error correction capability, with one such codeword needed to recover each erasure and two needed to correct each error.

However, if most of the error correction capacity is used to correct erasures, the possibility of undetected errors is increased. To adjust, whenever there are fewer than four errors corrected (except for s = 0) the error correction capacity becomes :

$$e + 2t \leq d - 3$$

To illustrate, a PDF417 symbol with error correction level 3 can correct up to 13 erasures or 7 errors, or any combination of e erasures and t errors subject to the applicable equation above. For instance, Error Correction Level 3 can correct 9 erasures and 2 errors.

2.6.3 Foundations of Error Detection and Correction

For a given string of data codewords, the error correction codewords are computed using the Reed-Solomon Error Control Code algorithm. The coding theory behind this scheme can be found in many coding theory texts, such as ***Error Control Coding, Fundamentals and Applications***, by Shu Lin and D. J. Costello, Jr. (Prentice-Hall, 1983).

The coefficients of the symbol data polynomial are the symbol character values in the data codeword area (see Figure 6), which include the symbol length descriptor, the data codewords, the pad codewords and optional Macro PDF417 Control Block. The symbol data polynomial is:

$$d(x) = d_{n-1}x^{n-1} + d_{n-2}x^{n-2} + ... + d_1x + d_0$$

The row indicators are not included in the error correction. The symbol length descriptor helps to detect row indicator misdecodes.

For a PDF417 symbol, Figure 6 illustrates the position of each symbol character value d_i where i = 0 ... n - 2, n - 1.

In Figure 6, c_i represents an error correction codeword, where i = 0... k - 2. k - 1. There are k error correction codewords. From the Error Correction Codeword column in Table 6 there are 9 possible numbers for k.

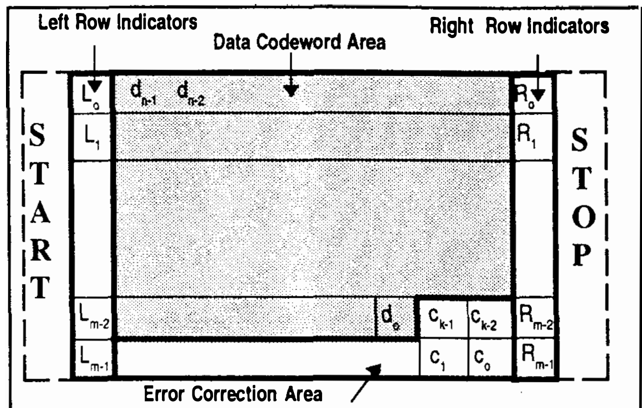

Figure 6: Data, Row Indicators and Error Correction

The following describes mathematically how to compute the error correction codewords for a given stream of data codewords and a selected error correction level. All of the arithmetic is done modulo 929.

The error correction codewords are the complement of coefficients of the remainder resulting from dividing the symbol data polynomial d(x), multiplied by x^k, by the generator polynomial g(x). The negative values in the Galois Field GF(929) are equal to the complement of the value if $c_i > -929$ or equal to the complement of the remainder (c_i /929) if $c_i \leq -929$. The generator polynomial for k error correction codewords is:

$$g(x) = (x - 3)(x - 3^2) ... (x - 3^k)$$

$$= x^k + g_{k-1}x^{k-1} + ... + g_1x + g_0$$

Tables of the resulting coefficients are included on the PDF417 developers diskette available from AIM USA. See Reference Documents section. This can be implemented by using a division circuit as shown in Figure 7. The registers b_0 through b_{k-1} are initialized as zeros. For our purposes, the modulo addition ⊕, modulo multiplication ⊗, and modulo complement © are defined as:

$$x \oplus y \equiv (x + y) \bmod 929$$
$$x \otimes y \equiv (x \times y) \bmod 929$$
$$© x \equiv (929 - x) \bmod 929$$

where x and y are numbers from 0 to 928.

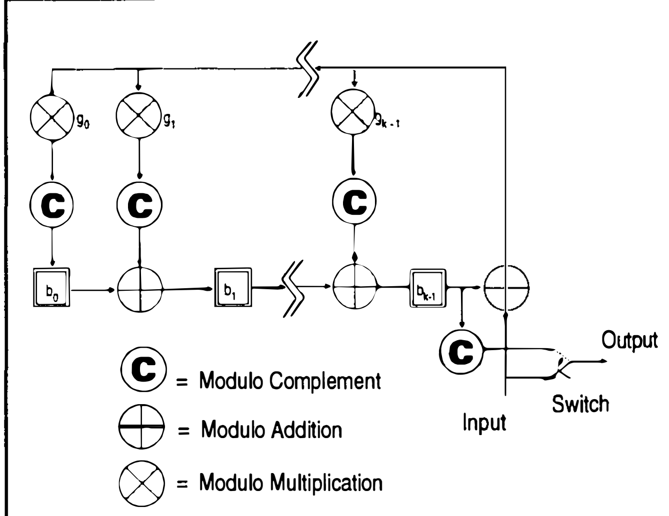

Figure 7: Error Correction Codeword Encoding Circuit

There are two phases to generate the encoding. In the first phase, with the switch in the down position the symbol data is passed both to the output and the circuit. The first phase is complete after n clock pulses. In the second phase (n + 1 ... n + k clock pulses), with the switch in the up position, the error correction codewords c_{k-1} ... c_0 are generated by flushing the registers in order and complementing the output while keeping the data input at 0.

2.6.4 Recommended Error Correction Level

For open systems, the recommended error correction levels for different amounts of encoded data are given in Table 7.

Number of Data Codewords	Error Correction Level
1 - 40	2
40 - 160	3
161 - 320	4
321 - 863	5

Table 7: Recommended Error Correction Levels

Lower than recommended error correction levels may be used in closed systems applications while higher levels are recommended for applications where significant symbol damage or degradation is anticipated.

As a guide to understanding Table 7, note that each "data codeword" encodes roughly 1.8 text characters or, in long numeric strings, 2.9 digits.

2.7 Transmitted Data

In the default interpretation, all data symbol characters are translated according to the compaction modes in effect and are included in the data transmission as a sequence of 8-bit bytes. Start and stop symbol characters, row indicators, the symbol length descriptor, mode switching symbol characters, and error correction symbol characters are not transmitted.

Certain special codewords with values greater than 902, which are not mode identifier codewords, should be communicated to the application system. These special codewords will be identified by an escape character, followed by three digits representing the symbol character's decimal value.

Following some special codewords, additional codewords should be encoded into decimal escape sequences to form a unique identification field. Each of these codewords is represented by its own escape sequence. Note that codewords 923 and 928 also redefine the compaction mode for subsequent characters. Table 8 lists the attributes of the special codewords.

The default escape character is "\", a backslash. If the escape character needs to be transmitted as

Special Codewords	Description	Additional Codewords Requiring Escape	Subsequent Mode
903 - 912	Reserved	0	Unchanged
914 - 920	Reserved	0	Unchanged
922	Macro PDF417 Terminator	0	Unchanged
923	Macro PDF417 Optional Field	1	*TC or NC
925	User-Defined GLI	1	Unchanged
926	General Purpose GLI	2	Unchanged
927	Intl. Char. Sets GLI	1	Unchanged
928	Macro PDF417 Control Header	Variable**	TC

* If the additional codeword value is 0, 3, or 4, then the subsequent mode is TC; otherwise, the mode is NC.
** See Appendix G.

Table 8: Transmission of Special Codewords

data, two escape characters are transmitted, e.g, "A\\B\" would be transmitted as "A\\\\B\\". The default escape character may be changed in the reader but not to any digit.

For example, GLI sequence "927 001" using the default escape character would be transmitted as "\927\001".

As an option, readers may have an operating mode where no escape character is defined; they do not transmit escape sequences or double any data characters.

2.7.1 Symbology Identifier

The AIM® USA Symbology Identifier Guideline provides a uniform methodology for reporting the symbology read, options set in the reader and any special features of the symbology encountered.

The symbology identifier for PDF417 is:

]L*m*

where:

] is the symbology identifier flag character (ASCII 93)

L is the symbology identification character for PDF417

m is the modifier character

Option values for PDF417 are listed in Table 9. The sum of all appropriate option values is the modifier character *m*.

Option Value	Option
0	No options, always transmit 0

(Permissible values of m: 0)

Table 9: Symbology Identifier Option Values for PDF417

3 Symbol Quality

3.1 Start and Stop Pattern Grade

The document ANSI X3.182 "Bar Code Print Quality - Guideline" presents a standardized methodology for measuring and grading bar code symbols. The PDF417 start and stop patterns shall be evaluated according to that document using the aperture size that is defined in the appropriate application standard. Note that this method does not provide a complete symbol quality grade with PDF417. (see Section 3.2)

Section 4 describes the reference decode algorithm for PDF417. This algorithm shall be used for evaluating the parameters "decode" and "decodability" in the start and stop patterns. PDF417 test scans shall be graded using these algorithms.

When performing a measurement, the scan lines should be made perpendicular to the start and stop patterns. This measurement of the start and stop pattern may be used for process control purposes. Note that this method will not be sensitive to printing variations parallel to the start and stop patterns. If a full analysis of the printing process is desired, PDF417 symbols should be printed and tested in both orientations.

3.2 Symbol Character Based Grade

This grade shall be based on an analysis of the entire symbol as scanned with an effective aperture defined in the appropriate application specification. Test scans shall be made slightly off the perpendicular of the start and stop patterns such that cross row scanning is utilized in all scans.

1. Determine a global threshold value for each scan equal in reflectance to $(R_{max} + R_{min})/2$ where R_{max} is the highest reflectance value in the scan and R_{min} is the lowest. All elements above the global threshold are spaces and those below are bars. Determine the t measurements in Figure 8 using an edge location equal to the mid-point reflectance of the adjacent bar and space pairs. Continue processing scans until the number of decoded codewords stabilizes.

2. Decode the symbol and note the percentage of unused error correction, then compare the percentage to the table below to determine the symbol character based grade.

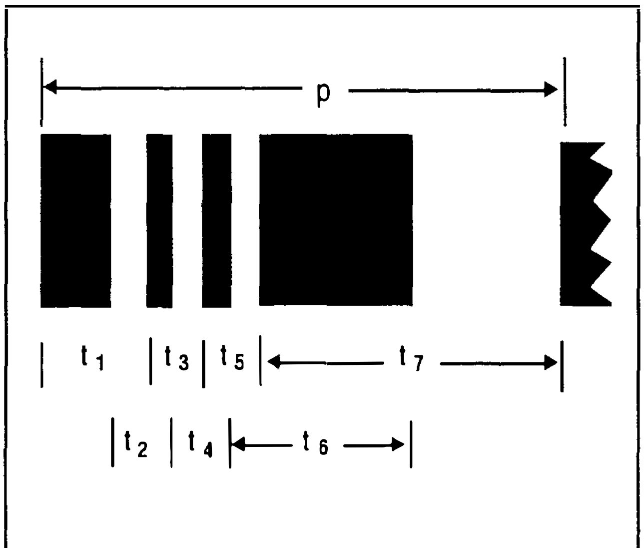

Figure 8: Decode Measurements

Percentage of Unused EC	Grade
≥ 62%	A
≥ 50%	B
≥ 37%	C
≥ 25%	D
< 25%	F

3.3 Overall Symbol Grade

The overall symbol grade is the lesser of the ANSI X3.182 start and stop pattern grade and the symbol character based grade.

4 Reference Decode Algorithm for PDF417

This section describes the reference decode algorithm used in the computation of decodability when assessing the symbol quality using the method described in the ANSI document X3.182, "Bar Code Print Quality – Guideline."

In this algorithm the symbol is decoded using "edge to similar edge'' measurements. The algorithm contains the following steps to decode each bar coded codeword:

1. Calculate seven width measurements p, t_1, t_2, t_3, t_4, t_5, t_6 (Figure 8).
2. Convert measurements $t_1, t_2, \ldots, t_6$ to normalized values T_1, T_2, T_3, T_4, T_5, and T_6 called T-sequence, which will represent the integral module width of these measurements. The following method is used for the i th value.

   If $1.5p / 17 \leq t_i < 2.5\,p/17$, then $T_i = 2$;
   If $2.5p / 17 \leq t_i < 3.5\,p/17$, then $T_i = 3$;
   If $3.5p / 17 \leq t_i < 4.5\,p/17$, then $T_i = 4$;
   If $4.5p / 17 \leq t_i < 5.5\,p/17$, then $T_i = 5$;
   If $5.5p / 17 \leq t_i < 6.5\,p/17$, then $T_i = 6$;
   If $6.5p / 17 \leq t_i < 7.5\,p/17$, then $T_i = 7$;
   If $7.5p / 17 \leq t_i < 8.5\,p/17$, then $T_i = 8$;
   If $8.5p / 17 \leq t_i < 9.5\,p/17$, then $T_i = 9$;

   Otherwise the character is in error.

3. Compute the codeword cluster number by:

cluster number = $(T_1 - T_2 + T_5 - T_6 + 9)$ mod 9

The cluster number is equal to 0, 3 or 6; otherwise the symbol character is in error.

4. Look up the symbol character using the cluster number, T_1, T_2, T_3, T_4, T_5, and T_6, as the key.

The calculation implicitly uses the cluster number to detect all decode errors caused by single non-systematic one-module edge errors.

Look for a start or stop pattern in the scanned data. Using the first two steps, decode the pattern by matching the T-sequence derived from the start and stop patterns (see Section 2.3). Attempt to decode all subsequent characters in the direction derived from the start or stop pattern decoded.

At first, the symbol structure parameters (number of rows, number of columns and error correction level) are unknown. These parameters must be initialized by decoding the row indicators, which are the symbol characters adjacent to the start and stop patterns.

After the symbol structure parameters are initialized, a matrix which reflects the size of the symbol to be decoded can be initialized. The number of columns of the matrix excludes the start and stop patterns and row indicators.

After finding a start or stop pattern, attempt to decode as many symbol characters as the number of columns in the matrix. Then the decode results are placed in the matrix at the appropriate positions, according to the row number, from one of the left or right row indicators. If row crossing happens, the cluster numbers will indicate this and can be used to deduce the correct row number for each individual symbol character. For example, a decoded scan has a left row indicator with row number 2, and the following data cluster numbers are 6,0,0,?,3 where the "?" denotes an illegal character pattern which occurred during row crossing. Then these symbol characters are filled in the positions of (1,2), (2,3), (3,3) and (5,4), where (c, r) stands for column c and row r. While the matrix is being filled, the erasure count, e, which is the number of unknown symbol characters, is also updated.

Error recovery is attempted when the number of erasures, e, satisfies the equations in Section 2.6.2. If error recovery fails then more codewords must be collected. For more details on error detection and correction see Appendix E.

Finally, the data codewords are interpreted according to the compaction modes and GLIs in effect.

Appendix A (Informative)

Auto-Discrimination Compatibility

PDF417 may be read by suitably programmed bar code readers that are designed to autodiscriminate this code from other symbologies. PDF417, in particular, is compatible with:

- Codabar
- Code 128
- Code 16K
- Code 39
- Code 49
- Code 93
- Code One
- EAN
- Interleaved 2 of 5
- U.P.C.

It is advisable to limit the reader's valid set of symbologies to those needed by a given application to maximize reading security.

Appendix B (Informative)

Human Readable Interpretation

Because PDF417 is capable of encoding thousands of characters, a human readable interpretation of the data characters may not be practical. As an alternative, descriptive text rather than literal text may accompany the symbol. If it is printed, the character size and font are not specified, and the message may be printed anywhere in the area surrounding the symbol. The human readable interpretation should not interfere with the symbol itself nor the quiet zones.

Figure B1 illustrates a typical PDF417 symbol incorporating human readable text.

AIM USA
634 Alpha Drive
Pittsburgh, PA 15238

Figure B1: The Human Readable Interpretation

Appendix C (Informative)

Systems Considerations

C.1 System Components

It is important that the various components (printers, labels, readers) making up a bar code installation operate together as a system. A failure in any component, or a mismatch between them, can compromise the performance of the overall system.

C.2 Specular Reflection

The effects of specular (mirror-like) reflections from shiny symbol surfaces must be considered. Standard reading systems are designed to detect variations in diffuse reflection between bars and spaces. At some scanning angles, the specular component of the reflected light can greatly exceed the desired diffuse component, which reduces reading performance. Matte (non-glossy) finishes minimize this effect.

In cases where specular reflection effects are used to achieve the desired contrasts (as in some forms of printing or etching directly onto metal), extreme care must be exercised to assure that the optical properties are within specifications over the entire range of scanning angles and distances required by the particular application.

C.3 Printing Considerations

Graphics software used to create bar codes on pixel-based printers must scale each bar and space exactly to the pixel pitch of the printer being used.

For two-width symbologies, the number of pixels can be chosen to come the closest to achieving the nominal widths.

For edge-to-edge decodable symbologies, the number of pixels comprising each symbol character must be an integer multiple of the number of modules for each symbol character. Any compensation for uniform bar growth or shrinkage must be in equal offsetting amounts on all bars and spaces in the symbol.

Failure to follow these procedures often results in unreadable symbols.

C.4 Other Considerations

Compliance with specifications is one key to assuring overall system success, but other considerations come into play, which can influence performance as well. The following guidelines suggest some factors to keep in mind when specifying or implementing bar code systems:

1. Choose a symbology and print density which yield tolerance values achievable by the printing technology to be used.

2. Choose a reader with resolution suitable for the symbol density and quality produced by the printing technology.

3. Be certain that the printed symbol's optical properties are compatible with the wavelength of the scanner's light source.

4. Be sure to verify symbol specification compliance in the final label or package configuration. Overlays, showthrough, and curved or irregular surfaces can all affect symbol readability.

5. The symbol should generally be printed with the largest X dimension that is practical, given label, package, printing, and scanning technology constraints.

6. The system should be programmed to accept only the bar code symbol message lengths and formats expected in that application.

7. When the row height is less than 4X, choose an error correction level that is at least as high as that shown in Table 7.

Appendix D (Informative)

Encoding Data

The same data may be represented by different PDF417 symbols through the use of different code sets. The following algorithm will tend to minimize the number of codewords.

1. Let P point to the start of data stream
2. Set current encoding mode to TC
3. Let N be the number of consecutive digits starting at P
4. If N is ≥ 13 then
    5. Latch to NC mode
    6. Encode the N characters using numeric compaction.
    7. Advance P by N
    8. Go to Step 3
9. Else if N < 13 then
    10. Let T be the length of a TC mode character sequence starting at P. The sequence is terminated when either a non-TC mode character is detected or a numeric sequence of ≥ 13 digits is detected.
    11. If T is ≥ 5 then
        12. Latch into TC mode
        13. Encode the T characters using the TC mode
        14. Advance P by T
        15. Go to Step 3
    16. Else if T < 5 then
        17. Let B be the length of the binary encodable sequence starting at P. The sequence is terminated when either a TC sequence of length ≥ 5 is found or a numeric sequence of length ≥ 13 is found.
        18. If B is equal to 1 AND the current mode is TC, then
            19. Shift into BC mode
            20. Encode the single byte value using BC mode
            21. Advance P by one
            22. Go to Step 3.
        23. Else
            24. Latch into BC mode
            25. Encode the B bytes using BC mode
            26. Advance P by B
            27. Go to Step 3

Appendix E (Informative)

Error Detection and Error Correction

When the total number of unknown codewords, υ, is less than or equal to the error correction capacity, the recovery scheme is invoked. The unknown codewords are substituted by zeros and the position of the l'th unknown codeword is j_l, for $l = 1,2,...,\upsilon$.

Construct the symbol character polynomial C(x) $= C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + ... + C_1x^1 + C_0$ where the n coefficients are the symbol character values read with C_{n-1} being the first symbol character and where n is the total number of symbol characters. Calculate i syndrome values S_0 through S_{i-1} by evaluating C(x) at $x = 3^k$ for k = 1 through i and where i is the number of error correction characters in the symbol. A circuit to generate the syndromes is shown in Figure E1.

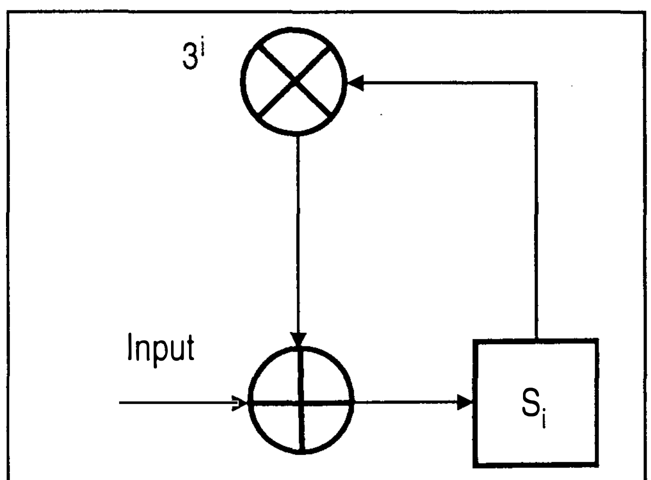

Figure E1: Symbol Syndrome Divider

Since the location of unknown codewords are known, j_l for $l = 1, 2, \ldots, \upsilon$, the error location polynomial for these known positions can be computed:

$$\Lambda(x) = (1 - xX_1)(1 - xX_2)\ldots(1 - xX_\upsilon)$$
$$= 1 + \Lambda_1 x + \ldots + \Lambda_\upsilon x^\upsilon$$

where $X_l = 3^{j_l}$

The error location polynomial, $\Lambda(x)$, can be updated to include the position of errors. This can be done by using the Berlekamp-Massey Algorithm, an error and erasure decoder for BCH codes, as discussed in ***Theory and Practice of Error Control Codes***, by Richard E. Blahut (Addison Wesley, 1983). At this point verify the number of erasures and errors satisfy the appropriate error correction capacity equation in Section 2.6.2.

Solving $\Lambda(x) = 0$ yields the position of the t errors, where $t \geq 0$; if $t = 0$ there is no error. Now, we only need to compute the error value, e_{j_l} for location j_l, $l = 1, \ldots, \upsilon + t$. To compute the error values, one auxiliary polynomial is needed, the Ω-polynomial, defined by:

$$\Omega(x) = 1 + (s_1 + \Lambda_1)x + (s_2 + \Lambda_1 s_1)x^2 + \ldots$$
$$+ (s_\eta + \Lambda_1 s_{\eta-1} + \Lambda_2 s_{\eta-2} + \ldots + \Lambda_{\eta-1} s_1)x^\eta$$

where $\eta = \upsilon + t$.

The error value at location j_l is thus given by:

$$Y_{j_l} = \frac{\Omega(X_l^{-1})}{X_l^{-1} \prod_{i=1, i \neq l}^{\eta} (1 - X_i X_l)}$$

After solving successfully for the error values, the complements of the error values are added to the corresponding locations of erroneous codewords.

Appendix F (Informative)

Truncated PDF417

In a relatively "clean" environment where label damage is unlikely (e.g., an office), the right row indicators can be omitted and the stop pattern can be reduced to one module width bar, as indicated in Figure F1. This truncation reduces the non-data overhead from 4 codewords per row to 2 codewords per row, with a trade-off in decode performance and robustness, or the ability to withstand degradation.

This version is called Truncated PDF417, which is fully reader compatible with standard PDF417.

Figure F1: Truncated PDF417

Appendix G (Informative)

Macro PDF417

G.1 Macro PDF417

Macro PDF417 provides a standard mechanism for creating a distributed representation of files too large to be represented by a single PDF417 symbol. Macro PDF417 symbols differ from ordinary PDF417 symbols in that they contain additional control information in a Macro PDF417 Control Block.

Using Macro PDF417, large files are split into several file segments and encoded into individual symbols. The Control Block defines the file ID, the concatenation sequence and optionally other information about the file. The Macro PDF417 decoder uses the Control Block's information to reconstruct the file correctly and then verify it, independent of symbol scanning order.

G.2 Macro PDF417 Syntax

Compared to standard PDF417, each Macro PDF417 symbol also encodes a Macro PDF417 Control Block containing control information. Macro PDF417 is defined as a trailer protocol, which means that the control block always follows the data block it is associated with. The length of the Control Block is included in the value of the symbol length descriptor.

The Control Block begins with a marker codeword with value 928 and extends to the beginning of the error correction codewords. It contains at least the two mandatory fields: a segment index and file ID. It also may contain a number of optional fields, as described later.

G.3 The Segment Index

In Macro PDF417, each symbol represents a segment of the whole file. To reconstruct the whole file, the segments must be reconstructed in the correct order. Control information in the Control Block facilitates this reassembly process. For a file divided into a set of k Macro PDF417 symbols, the segment index field in each symbol's Control Block contains a value between 0 and k-1, corresponding to the relative position of that symbol's content within the distributed representation.

The segment index field is 2 codewords in length and is encoded using Numeric Compaction mode as defined in GLI 0. Note, the segment index value must be padded with leading zeros to five digits before Numeric Compaction is applied. The largest allowed value in the segment index field is 99,998. Thus, up to 99,999 Macro PDF417 symbols may comprise the distributed representation of a data file. This translates to a capacity limit of nearly 110 million bytes of data in Byte Compaction mode or 184 million characters in Text Compaction mode.

G.4 File ID Layout

For each related Macro PDF417 symbol, the file ID field contains the same value. This ensures that all reassembled symbol data belongs to the same distributed file representation. The file ID is a variable length field which begins with the first codeword following the segment index and extends to the start of the optional fields (if present) or to the end of the Control Block (if not).

Each codeword in file ID can have a value between 0 and 899. This identification scheme's effectiveness is influenced by both the length of the file ID field and the suitability of the algorithm used to generate its value.

G.5 Optional Field Layout

Optional fields may follow the file ID. Each optional field begins with a specific tag sequence and extends until the start of the next optional field (if present) or the end of the Control Block (if not). These fields always represent global file attributes and so need not be present in the Control Block of more than one Macro PDF417 symbol within the distributed file representation, with the exception of the segment count field, as described below. The segment which contains these fields is defined by the specific encoder implementation. If a particular field is to appear in more than one segment, it must appear identically in every segment.

The tag sequence consists of a codeword 923 followed by a single codeword field designator. In each optional field, data following the tag sequence has a field-specific interpretation. Any of the general purpose or mode identifier codewords may be used within these fields. Empty optional fields are not to be used.

Table G1 shows the correspondence between currently defined field designators and optional field contents.

Field Designator	Contents	Compaction Mode	Total Field Length
0	File Name	Text Compaction	Variable
1	Segment Count	Numeric Compaction	4
2	Time Stamp	Numeric Compaction	6
3	Sender	Text Compaction	Variable
4	Addressee	Text Compaction	Variable
5	File Size	Numeric Compaction	Variable
6	Checksum	Numeric Compaction	4

Table G1: Macro PDF417 Field Designators

As shown in the table above, all optional fields use standard PDF417 high-level encoding. At the beginning of each field, the default mode in effect is defined by Table G1, regardless of mode shifts and latches earlier in the symbol.

The segment count field can contain values from 1 to 99,999 and is always encoded as two codewords. If the optional segment count field is used, that field shall appear in every segment.

The time stamp field is interpreted in Numeric Compaction mode. It indicates the time stamp on the source file expressed as the elapsed time in seconds since January 1, 1970 00:00 GMT. Using this format, four codewords can encode any date in the foreseeable future.

The file size field contains the size in bytes of the entire source file.

The checksum field contains the value of the 16-bit (2 bytes) CRC checksum using the CCITT-16 polynomial $x^{16} + x^{12} + x^5 + 1$ computed over the entire source file.

Field designator values greater than 6 are reserved.

G.6 Macro PDF417 Terminator

The Control Block in the symbol representing the last segment of a Macro PDF417 file contains a special marker. It consists of a single codeword with value 922 at the end of the Control Block. The Control Block for every other symbol ends after any optional fields with no special terminator.

G.7 High-Level Encoding Considerations

While Macro PDF417 provides a mechanism for logically associating a set of symbols, it is important to realize that, with respect to high-level encoding, each symbol remains a distinct entity. In other words, the scope of a mode switch or Global Label Identifier is confined to the symbols in which it occurs. Each symbol implicitly begins in the default interpretation of the Text Compaction mode.

In the context of a Control Block optional field, the compaction modes indicated in Table G1 supersede that set by the mode identifier codewords within the data codeword region of the symbol. The scope of the current GLI, however, does extend to those optional fields.

G.8 Macro PDF417 Transmission

The transmission of Macro PDF417 information can either be buffered or unbuffered. A buffered transmission scheme requires the reader to collect the entire symbol set prior to its transmission. Unbuffered, on the other hand, allows the reader to transmit the individual symbols as they are decoded.

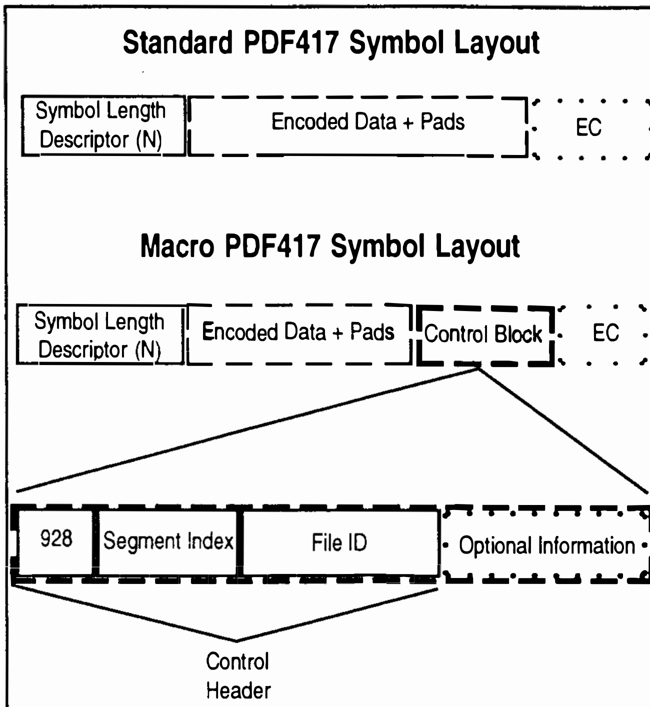

Figure G1: PDF417 Symbol Layouts

Note that under either scheme, the PDF417 decoder, once it has processed a Macro PDF417 symbol with a given File ID. must decode and transmit all of the symbols for that File ID before it can transmit any other symbols. Since the symbols in the unbuffered scheme are not ordered internally by the reader it is recommended that the transmission of the Macro PDF417 Control Header be enabled. This allows the host system to impose the proper ordering on the received data.

Transmission of Macro PDF417 Control Header can be enabled or disabled. The Macro PDF417 Control Header is a portion of the Macro PDF417 Control Block. Figure G1, which consists of the marker codeword 928, the Segment Index (in Numeric Compaction mode), and the File ID codeword sequence. When transmission of the Control Header is enabled, the marker codeword and the File ID codewords should be transmitted using the escape character as defined in Section 2.7. For example. the Macro PDF417 Control Header of the first symbol. Segment Index = 0, with a File ID

(100, 200, 300) would be encoded in the symbol as the codeword sequence:

<928>, <111>, <100>, <100>, <200>, <300>

and would be transmitted (assuming the default escape of "\" has not been changed) as:

\92800000\100\200\300

If enabled, the Macro PDF417 Control Header must be transmitted following the data encoded in the symbol.

When the last GLI sequence transmitted by the reader was other than GLI 0, then the transmitted data from that segment must be terminated with \927\000, as if the symbol's data ended with the sequence of codewords <927>, <0>. This reverts the interpretation of the next block back to GLI 0.

Transmission of optional fields can be individually enabled or disabled in the reader. If the reader is programmed to be in buffered mode, the enabled optional fields are transmitted once, at the end of decoding the Macro PDF417 set; in unbuffered mode, the enabled optional fields are transmitted with each Macro PDF417 symbol in which they have been encoded. Each field begins with the transmission of the corresponding Macro PDF417 optional field tag sequence. The tag sequence consists of the codeword value 923 followed by a tag value as defined in Table G1; this sequence is transmitted using the escape character as defined in Section 2.7. The high level decoded content of the field is transmitted after this tag sequence.

Appendix H (Normative)

Symbol Character Patterns

Table H1 gives the bar and space sequence and symbol character value (val) of each codeword in three clusters. The corresponding T-sequence used in the decoding phase can be derived from the bar and space sequences in Table H1.

A reference diskette containing this table is available from AIM® USA. See Reference Documents section.

Table H1. The Bar-Space Sequence Table. Cluster 0

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
31111136	0	41111144	1	51111152	2	31111235	3	41111243	4	51111251	5
21111326	6	31111334	7	21111425	8	11111516	9	21111524	10	11111615	11
21112136	12	31112144	13	41112152	14	21112235	15	31112243	16	41112251	17
11112326	18	21112334	19	11112425	20	11113136	21	21113144	22	31113152	23
11113235	24	21113243	25	31113251	26	11113334	27	21113342	28	11114144	29
21114152	30	11114243	31	21114251	32	11115152	33	51116111	34	31121135	35
41121143	36	51121151	37	21121226	38	31121234	39	41121242	40	21121325	41
31121333	42	11121416	43	21121424	44	31121432	45	11121515	46	21121523	47
11121614	48	21122135	49	31122143	50	41122151	51	11122226	52	21122234	53
31122242	54	11122325	55	21122333	56	31122341	57	11122424	58	21122432	59
11123135	60	21123143	61	31123151	62	11123234	63	21123242	64	11123333	65
21123341	66	11124143	67	21124151	68	11124242	69	11124341	70	21131126	71
31131134	72	41131142	73	21131225	74	31131233	75	41131241	76	11131316	77
21131324	78	31131332	79	11131415	80	21131423	81	11131514	82	11131613	83
11132126	84	21132134	85	31132142	86	11132225	87	21132233	88	31132241	89
11132324	90	21132332	91	11132423	92	11132522	93	11133134	94	21133142	95
11133233	96	21133241	97	11133332	98	11134142	99	21141125	100	31141133	101
41141141	102	11141216	103	21141224	104	31141232	105	11141315	106	21141323	107
31141331	108	11141414	109	21141422	110	11141513	111	21141521	112	11142125	113
21142133	114	31142141	115	11142224	116	21142232	117	11142323	118	21142331	119
11142422	120	11142521	121	21143141	122	11143331	123	11151116	124	21151124	125
31151132	126	11151215	127	21151223	128	31151231	129	11151314	130	21151322	131
11151413	132	21151421	133	11151512	134	11152124	135	11152223	136	11152322	137
11161115	138	31161131	139	21161222	140	21161321	141	11161511	142	32111135	143
42111143	144	52111151	145	22111226	146	32111234	147	42111242	148	22111325	149
32111333	150	42111341	151	12111416	152	22111424	153	12111515	154	22112135	155
32112143	156	42112151	157	12112226	158	22112234	159	32112242	160	12112325	161
22112333	162	12112424	163	12112523	164	12113135	165	22113143	166	32113151	167
12113234	168	22113242	169	12113333	170	12113432	171	12114143	172	22114151	173
12114242	174	12115151	175	31211126	176	41211134	177	51211142	178	31211225	179
41211233	180	51211241	181	21211316	182	31211324	183	41211332	184	21211415	185
31211423	186	41211431	187	21211514	188	31211522	189	22121126	190	32121134	191
42121142	192	21212126	193	22121225	194	32121233	195	42121241	196	21212225	197
31212233	198	41212241	199	11212316	200	12121415	201	22121423	202	32121431	203
11212415	204	21212423	205	11212514	206	12122126	207	22122134	208	32122142	209
11213126	210	12122225	211	22122233	212	32122241	213	11213225	214	21213233	215
31213241	216	11213324	217	12122423	218	11213423	219	12123134	220	22123142	221
11214134	222	12123233	223	22123241	224	11214233	225	21214241	226	11214332	227
12124142	228	11215142	229	12124241	230	11215241	231	31221125	232	41221133	233
51221141	234	21221216	235	31221224	236	41221232	237	21221315	238	31221323	239
41221331	240	21221414	241	31221422	242	21221513	243	21221612	244	22131125	245
32131133	246	42131141	247	21222125	248	22131224	249	32131232	250	11222216	251
12131315	252	31222232	253	32131331	254	11222315	255	12131414	256	22131422	257
11222414	258	21222422	259	22131521	260	12131612	261	12132125	262	22132133	263
32132141	264	11223125	265	12132224	266	22132232	267	11223224	268	21223232	269
22132331	270	11223323	271	12132422	272	12132521	273	12133133	274	22133141	275
11224133	276	12133232	277	11224232	278	12133331	279	11224331	280	11225141	281
21231116	282	31231124	283	41231132	284	21231215	285	31231223	286	41231231	287
21231314	288	31231322	289	21231413	290	31231421	291	21231512	292	21231611	293
12141116	294	22141124	295	32141132	296	11232116	297	12141215	298	22141223	299
32141231	300	11232215	301	21232223	302	31232231	303	11232314	304	12141413	305
22141421	306	11232413	307	21232421	308	11232512	309	12142124	310	22142132	311
11233124	312	12142223	313	22142231	314	11233223	315	21233231	316	11233322	317
12142421	318	11233421	319	11234132	320	11234231	321	21241115	322	31241123	323
41241131	324	21241214	325	31241222	326	21241313	327	31241321	328	21241412	329

Table H1. The Bar-Space Sequence Table. Cluster 0

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
21241511	330	12151115	331	22151123	332	32151131	333	11242115	334	12151214	335
22151222	336	11242214	337	21242222	338	22151321	339	11242313	340	12151412	341
11242412	342	12151511	343	12152123	344	11243123	345	11243222	346	11243321	347
31251122	348	31251221	349	21251411	350	22161122	351	12161213	352	11252213	353
11252312	354	11252411	355	23111126	356	33111134	357	43111142	358	23111225	359
33111233	360	13111316	361	23111324	362	33111332	363	13111415	364	23111423	365
13111514	366	13111613	367	13112126	368	23112134	369	33112142	370	13112225	371
23112233	372	33112241	373	13112324	374	23112332	375	13112423	376	13112522	377
13113134	378	23113142	379	13113233	380	23113241	381	13113332	382	13114142	383
13114241	384	32211125	385	42211133	386	52211141	387	22211216	388	32211224	389
42211232	390	22211315	391	32211323	392	42211331	393	22211414	394	32211422	395
22211513	396	32211521	397	23121125	398	33121133	399	43121141	400	22212125	401
23121224	402	33121232	403	12212216	404	13121315	405	32212232	406	33121331	407
12212315	408	22212323	409	23121422	410	12212414	411	13121513	412	12212513	413
13122125	414	23122133	415	33122141	416	12213125	417	13122224	418	32213141	419
12213224	420	22213232	421	23122331	422	12213323	423	13122422	424	12213422	425
13123133	426	23123141	427	12214133	428	13123232	429	12214232	430	13123331	431
13124141	432	12215141	433	31311116	434	41311124	435	51311132	436	31311215	437
41311223	438	51311231	439	31311314	440	41311322	441	31311413	442	41311421	443
31311512	444	22221116	445	32221124	446	42221132	447	21312116	448	22221215	449
41312132	450	42221231	451	21312215	452	31312223	453	41312231	454	21312314	455
22221413	456	32221421	457	21312413	458	31312421	459	22221611	460	13131116	461
23131124	462	33131132	463	12222116	464	13131215	465	23131223	466	33131231	467
11313116	468	12222215	469	22222223	470	32222231	471	11313215	472	21313223	473
31313231	474	23131421	475	11313314	476	12222413	477	22222421	478	11313413	479
13131611	480	13132124	481	23132132	482	12223124	483	13132223	484	23132231	485
11314124	486	12223223	487	22223231	488	11314223	489	21314231	490	13132421	491
12223421	492	13133132	493	12224132	494	13133231	495	11315132	496	12224231	497
31321115	498	41321123	499	51321131	500	31321214	501	41321222	502	31321313	503
41321321	504	31321412	505	31321511	506	22231115	507	32231123	508	42231131	509
21322115	510	22231214	511	41322131	512	21322214	513	31322222	514	32231321	515
21322313	516	22231412	517	21322412	518	22231511	519	21322511	520	13141115	521
23141123	522	33141131	523	12232115	524	13141214	525	23141222	526	11323115	527
12232214	528	22232222	529	23141321	530	11323214	531	21323222	532	13141412	533
11323313	534	12232412	535	13141511	536	12232511	537	13142123	538	23142131	539
12233123	540	13142222	541	11324123	542	12233222	543	13142321	544	11324222	545
12233321	546	13143131	547	11325131	548	31331114	549	41331122	550	31331213	551
41331221	552	31331312	553	31331411	554	22241114	555	32241122	556	21332114	557
22241213	558	32241221	559	21332213	560	31332221	561	21332312	562	22241411	563
21332411	564	13151114	565	23151122	566	12242114	567	13151213	568	23151221	569
11333114	570	12242213	571	22242221	572	11333213	573	21333221	574	13151411	575
11333312	576	12242411	577	11333411	578	12243122	579	11334122	580	11334221	581
41341121	582	31341311	583	32251121	584	22251212	585	22251311	586	13161113	587
12252113	588	11343113	589	13161311	590	12252311	591	24111125	592	14111216	593
24111224	594	14111315	595	24111323	596	34111331	597	14111414	598	24111422	599
14111513	600	24111521	601	14112125	602	24112133	603	34112141	604	14112224	605
24112232	606	14112323	607	24112331	608	14112422	609	14112521	610	14113133	611
24113141	612	14113232	613	14113331	614	14114141	615	23211116	616	33211124	617
43211132	618	23211215	619	33211223	620	23211314	621	33211322	622	23211413	623
33211421	624	23211512	625	14121116	626	24121124	627	34121132	628	13212116	629
14121215	630	33212132	631	34121231	632	13212215	633	23212223	634	33212231	635
13212314	636	14121413	637	24121421	638	13212413	639	23212421	640	14121611	641
14122124	642	24122132	643	13213124	644	14122223	645	24122231	646	13213223	647
23213231	648	13213322	649	14122421	650	14123132	651	13214132	652	14123231	653
13214231	654	32311115	655	42311123	656	52311131	657	32311214	658	42311222	659

Table H1. The Bar-Space Sequence Table. Cluster 0

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
32311313	660	42311321	661	32311412	662	32311511	663	23221115	664	33221123	665
22312115	666	23221214	667	33221222	668	22312214	669	32312222	670	33221321	671
22312313	672	23221412	673	22312412	674	23221511	675	22312511	676	14131115	677
24131123	678	13222115	679	14131214	680	33222131	681	12313115	682	13222214	683
23222222	684	24131321	685	12313214	686	22313222	687	14131412	688	12313313	689
13222412	690	14131511	691	13222511	692	14132123	693	24132131	694	13223123	695
14132222	696	12314123	697	13223222	698	14132321	699	12314222	700	13223321	701
14133131	702	13224131	703	12315131	704	41411114	705	51411122	706	41411213	707
51411221	708	41411312	709	41411411	710	32321114	711	42321122	712	31412114	713
41412122	714	42321221	715	31412213	716	41412221	717	31412312	718	32321411	719
31412411	720	23231114	721	33231122	722	22322114	723	23231213	724	33231221	725
21413114	726	22322213	727	32322221	728	21413213	729	31413221	730	23231411	731
21413312	732	22322411	733	21413411	734	14141114	735	24141122	736	13232114	737
14141213	738	24141221	739	12323114	740	13232213	741	23232221	742	11414114	743
12323213	744	22323221	745	14141411	746	11414213	747	21414221	748	13232411	749
11414312	750	14142122	751	13233122	752	14142221	753	12324122	754	13233221	755
11415122	756	12324221	757	11415221	758	41421113	759	51421121	760	41421212	761
41421311	762	32331113	763	42331121	764	31422113	765	41422121	766	31422212	767
32331311	768	31422311	769	23241113	770	33241121	771	22332113	772	23241212	773
21423113	774	22332212	775	23241311	776	21423212	777	22332311	778	21423311	779
14151113	780	24151121	781	13242113	782	23242121	783	12333113	784	13242212	785
14151311	786	11424113	787	12333212	788	13242311	789	11424212	790	12333311	791
11424311	792	13243121	793	11425121	794	41431211	795	31432112	796	31432211	797
22342112	798	21433112	799	21433211	800	13252112	801	12343112	802	11434112	803
11434211	804	15111116	805	15111215	806	25111223	807	15111314	808	15111413	809
15111512	810	15112124	811	15112223	812	15112322	813	15112421	814	15113132	815
15113231	816	24211115	817	24211214	818	34211222	819	24211313	820	34211321	821
24211412	822	24211511	823	15121115	824	25121123	825	14212115	826	24212123	827
25121222	828	14212214	829	24212222	830	14212313	831	24212321	832	14212412	833
15121511	834	14212511	835	15122123	836	25122131	837	14213123	838	24213131	839
14213222	840	15122321	841	14213321	842	15123131	843	14214131	844	33311114	845
33311213	846	33311312	847	33311411	848	24221114	849	23312114	850	33312122	851
34221221	852	23312213	853	33312221	854	23312312	855	24221411	856	23312411	857
15131114	858	14222114	859	15131213	860	25131221	861	13313114	862	14222213	863
15131312	864	13313213	865	14222312	866	15131411	867	13313312	868	14222411	869
15132122	870	14223122	871	15132221	872	13314122	873	14223221	874	13314221	875
42411113	876	42411212	877	42411311	878	33321113	879	32412113	880	42412121	881
32412212	882	33321311	883	32412311	884	24231113	885	34231121	886	23322113	887
33322121	888	22413113	889	23322212	890	24231311	891	22413212	892	23322311	893
22413311	894	15141113	895	25141121	896	14232113	897	24232121	898	13323113	899
14232212	900	15141311	901	12414113	902	13323212	903	14232311	904	12414212	905
13323311	906	15142121	907	14233121	908	13324121	909	12415121	910	51511112	911
51511211	912	42421112	913	41512112	914	42421211	915	41512211	916	33331112	917
32422112	918	33331211	919	31513112	920	32422211	921	31513211	922	24241112	923
23332112	924	24241211	925	22423112	926	23332211	927	21514112	928		

Table H1. The Bar-Space Sequence Table. Cluster 3

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
51111125	0	61111133	1	41111216	2	51111224	3	61111232	4	41111315	5
51111323	6	61111331	7	41111414	8	51111422	9	41111513	10	51111521	11
41111612	12	41112125	13	51112133	14	61112141	15	31112216	16	41112224	17
51112232	18	31112315	19	41112323	20	51112331	21	31112414	22	41112422	23
31112513	24	41112521	25	31112612	26	31113125	27	41113133	28	51113141	29
21113216	30	31113224	31	41113232	32	21113315	33	31113323	34	41113331	35
21113414	36	31113422	37	21113513	38	31113521	39	21113612	40	21114125	41
31114133	42	41114141	43	11114216	44	21114224	45	31114232	46	11114315	47
21114323	48	31114331	49	11114414	50	21114422	51	11114513	52	21114521	53
11115125	54	21115133	55	31115141	56	11115224	57	21115232	58	11115323	59
21115331	60	11115422	61	11116133	62	21116141	63	11116232	64	11116331	65
41121116	66	51121124	67	61121132	68	41121215	69	51121223	70	61121231	71
41121314	72	51121322	73	41121413	74	51121421	75	41121512	76	41121611	77
31122116	78	41122124	79	51122132	80	31122215	81	41122223	82	51122231	83
31122314	84	41122322	85	31122413	86	41122421	87	31122512	88	31122611	89
21123116	90	31123124	91	41123132	92	21123215	93	31123223	94	41123231	95
21123314	96	31123322	97	21123413	98	31123421	99	21123512	100	21123611	101
11124116	102	21124124	103	31124132	104	11124215	105	21124223	106	31124231	107
11124314	108	21124322	109	11124413	110	21124421	111	11124512	112	11125124	113
21125132	114	11125223	115	21125231	116	11125322	117	11125421	118	11126132	119
11126231	120	41131115	121	51131123	122	61131131	123	41131214	124	51131222	125
41131313	126	51131321	127	41131412	128	41131511	129	31132115	130	41132123	131
51132131	132	31132214	133	41132222	134	31132313	135	41132321	136	31132412	137
31132511	138	21133115	139	31133123	140	41133131	141	21133214	142	31133222	143
21133313	144	31133321	145	21133412	146	21133511	147	11134115	148	21134123	149
31134131	150	11134214	151	21134222	152	11134313	153	21134321	154	11134412	155
11134511	156	11135123	157	21135131	158	11135222	159	11135321	160	11136131	161
41141114	162	51141122	163	41141213	164	51141221	165	41141312	166	41141411	167
31142114	168	41142122	169	31142213	170	41142221	171	31142312	172	31142411	173
21143114	174	31143122	175	21143213	176	31143221	177	21143312	178	21143411	179
11144114	180	21144122	181	11144213	182	21144221	183	11144312	184	11144411	185
11145122	186	11145221	187	41151113	188	51151121	189	41151212	190	41151311	191
31152113	192	41152121	193	31152212	194	31152311	195	21153113	196	31153121	197
21153212	198	21153311	199	11154113	200	21154121	201	11154212	202	11154311	203
41161112	204	41161211	205	31162112	206	31162211	207	21163112	208	21163211	209
42111116	210	52111124	211	62111132	212	42111215	213	52111223	214	62111231	215
42111314	216	52111322	217	42111413	218	52111421	219	42111512	220	42111611	221
32112116	222	42112124	223	52112132	224	32112215	225	42112223	226	52112231	227
32112314	228	42112322	229	32112413	230	42112421	231	32112512	232	32112611	233
22113116	234	32113124	235	42113132	236	22113215	237	32113223	238	42113231	239
22113314	240	32113322	241	22113413	242	32113421	243	22113512	244	22113611	245
12114116	246	22114124	247	32114132	248	12114215	249	22114223	250	32114231	251
12114314	252	22114322	253	12114413	254	22114421	255	12114512	256	12115124	257
22115132	258	12115223	259	22115231	260	12115322	261	12115421	262	12116132	263
12116231	264	51211115	265	61211123	266	11211164	267	51211214	268	61211222	269
11211263	270	51211313	271	61211321	272	11211362	273	51211412	274	51211511	275
42121115	276	52121123	277	62121131	278	41212115	279	42121214	280	61212131	281
41212214	282	51212222	283	52121321	284	41212313	285	42121412	286	41212412	287
42121511	288	41212511	289	32122115	290	42122123	291	52122131	292	31213115	293
32122214	294	42122222	295	31213214	296	41213222	297	42122321	298	31213313	299
32122412	300	31213412	301	32122511	302	31213511	303	22123115	304	32123123	305
42123131	306	21214115	307	22123214	308	32123222	309	21214214	310	31214222	311
32123321	312	21214313	313	22123412	314	21214412	315	22123511	316	21214511	317
12124115	318	22124123	319	32124131	320	11215115	321	12124214	322	22124222	323
11215214	324	21215222	325	22124321	326	11215313	327	12124412	328	11215412	329

Table H1. The Bar-Space Sequence Table. Cluster 3

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
12124511	330	12125123	331	22125131	332	11216123	333	12125222	334	11216222	335
12125321	336	11216321	337	12126131	338	51221114	339	61221122	340	11221163	341
51221213	342	61221221	343	11221262	344	51221312	345	11221361	346	51221411	347
42131114	348	52131122	349	41222114	350	42131213	351	52131221	352	41222213	353
51222221	354	41222312	355	42131411	356	41222411	357	32132114	358	42132122	359
31223114	360	32132213	361	42132221	362	31223213	363	41223221	364	31223312	365
32132411	366	31223411	367	22133114	368	32133122	369	21224114	370	22133213	371
32133221	372	21224213	373	31224221	374	21224312	375	22133411	376	21224411	377
12134114	378	22134122	379	11225114	380	12134213	381	22134221	382	11225213	383
21225221	384	11225312	385	12134411	386	11225411	387	12135122	388	11226122	389
12135221	390	11226221	391	51231113	392	61231121	393	11231162	394	51231212	395
11231261	396	51231311	397	42141113	398	52141121	399	41232113	400	51232121	401
41232212	402	42141311	403	41232311	404	32142113	405	42142121	406	31233113	407
32142212	408	31233212	409	32142311	410	31233311	411	22143113	412	32143121	413
21234113	414	31234121	415	21234212	416	22143311	417	21234311	418	12144113	419
22144121	420	11235113	421	12144212	422	11235212	423	12144311	424	11235311	425
12145121	426	11236121	427	51241112	428	11241161	429	51241211	430	42151112	431
41242112	432	42151211	433	41242211	434	32152112	435	31243112	436	32152211	437
31243211	438	22153112	439	21244112	440	22153211	441	21244211	442	12154112	443
11245112	444	12154211	445	11245211	446	51251111	447	42161111	448	41252111	449
32162111	450	31253111	451	22163111	452	21254111	453	43111115	454	53111123	455
63111131	456	43111214	457	53111222	458	43111313	459	53111321	460	43111412	461
43111511	462	33112115	463	43112123	464	53112131	465	33112214	466	43112222	467
33112313	468	43112321	469	33112412	470	33112511	471	23113115	472	33113123	473
43113131	474	23113214	475	33113222	476	23113313	477	33113321	478	23113412	479
23113511	480	13114115	481	23114123	482	33114131	483	13114214	484	23114222	485
13114313	486	23114321	487	13114412	488	13114511	489	13115123	490	23115131	491
13115222	492	13115321	493	13116131	494	52211114	495	62211122	496	12211163	497
52211213	498	62211221	499	12211262	500	52211312	501	12211361	502	52211411	503
43121114	504	53121122	505	42212114	506	43121213	507	53121221	508	42212213	509
52212221	510	42212312	511	43121411	512	42212411	513	33122114	514	43122122	515
32213114	516	33122213	517	43122221	518	32213213	519	42213221	520	32213312	521
33122411	522	32213411	523	23123114	524	33123122	525	22214114	526	23123213	527
33123221	528	22214213	529	32214221	530	22214312	531	23123411	532	22214411	533
13124114	534	23124122	535	12215114	536	13124213	537	23124221	538	12215213	539
22215221	540	12215312	541	13124411	542	12215411	543	13125122	544	12216122	545
13125221	546	12216221	547	61311113	548	11311154	549	21311162	550	61311212	551
11311253	552	21311261	553	61311311	554	11311352	555	11311451	556	52221113	557
62221121	558	12221162	559	51312113	560	61312121	561	11312162	562	12221261	563
51312212	564	52221311	565	11312261	566	51312311	567	43131113	568	53131121	569
42222113	570	43131212	571	41313113	572	51313121	573	43131311	574	41313212	575
42222311	576	41313311	577	33132113	578	43132121	579	32223113	580	33132212	581
31314113	582	32223212	583	33132311	584	31314212	585	32223311	586	31314311	587
23133113	588	33133121	589	22224113	590	23133212	591	21315113	592	22224212	593
23133311	594	21315212	595	22224311	596	21315311	597	13134113	598	23134121	599
12225113	600	13134212	601	11316113	602	12225212	603	13134311	604	11316212	605
12225311	606	11316311	607	13135121	608	12226121	609	61321112	610	11321153	611
21321161	612	61321211	613	11321252	614	11321351	615	52231112	616	12231161	617
51322112	618	52231211	619	11322161	620	51322211	621	43141112	622	42232112	623
43141211	624	41323112	625	42232211	626	41323211	627	33142112	628	32233112	629
33142211	630	31324112	631	32233211	632	31324211	633	23143112	634	22234112	635
23143211	636	21325112	637	22234211	638	21325211	639	13144112	640	12235112	641
13144211	642	11326112	643	12235211	644	11326211	645	61331111	646	11331152	647
11331251	648	52241111	649	51332111	650	43151111	651	42242111	652	41333111	653
33152111	654	32243111	655	31334111	656	23153111	657	22244111	658	21335111	659

Table H1. The Bar-Space Sequence Table. Cluster 3

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
13154111	660	12245111	661	11336111	662	11341151	663	44111114	664	54111122	665
44111213	666	54111221	667	44111312	668	44111411	669	34112114	670	44112122	671
34112213	672	44112221	673	34112312	674	34112411	675	24113114	676	34113122	677
24113213	678	34113221	679	24113312	680	24113411	681	14114114	682	24114122	683
14114213	684	24114221	685	14114312	686	14114411	687	14115122	688	14115221	689
53211113	690	63211121	691	13211162	692	53211212	693	13211261	694	53211311	695
44121113	696	54121121	697	43212113	698	44121212	699	43212212	700	44121311	701
43212311	702	34122113	703	44122121	704	33213113	705	34122212	706	33213212	707
34122311	708	33213311	709	24123113	710	34123121	711	23214113	712	24123212	713
23214212	714	24123311	715	23214311	716	14124113	717	24124121	718	13215113	719
14124212	720	13215212	721	14124311	722	13215311	723	14125121	724	13216121	725
62311112	726	12311153	727	22311161	728	62311211	729	12311252	730	12311351	731
53221112	732	13221161	733	52312112	734	53221211	735	12312161	736	52312211	737
44131112	738	43222112	739	44131211	740	42313112	741	43222211	742	42313211	743
34132112	744	33223112	745	34132211	746	32314112	747	33223211	748	32314211	749
24133112	750	23224112	751	24133211	752	22315112	753	23224211	754	22315211	755
14134112	756	13225112	757	14134211	758	12316112	759	13225211	760	12316211	761
11411144	762	21411152	763	11411243	764	21411251	765	11411342	766	11411441	767
62321111	768	12321152	769	61412111	770	11412152	771	12321251	772	11412251	773
53231111	774	52322111	775	51413111	776	44141111	777	43232111	778	42323111	779
41414111	780	34142111	781	33233111	782	32324111	783	31415111	784	24143111	785
23234111	786	22325111	787	21416111	788	14144111	789	13235111	790	12326111	791
11421143	792	21421151	793	11421242	794	11421341	795	12331151	796	11422151	797
11431142	798	11431241	799	11441141	800	45111113	801	45111212	802	45111311	803
35112113	804	45112121	805	35112212	806	35112311	807	25113113	808	35113121	809
25113212	810	25113311	811	15114113	812	25114121	813	15114212	814	15114311	815
15115121	816	54211112	817	14211161	818	54211211	819	45121112	820	44212112	821
45121211	822	44212211	823	35122112	824	34213112	825	35122211	826	34213211	827
25123112	828	24214112	829	25123211	830	24214211	831	15124112	832	14215112	833
15124211	834	14215211	835	63311111	836	13311152	837	13311251	838	54221111	839
53312111	840	45131111	841	44222111	842	43313111	843	35132111	844	34223111	845
33314111	846	25133111	847	24224111	848	23315111	849	15134111	850	14225111	851
13316111	852	12411143	853	22411151	854	12411242	855	12411341	856	13321151	857
12412151	858	11511134	859	21511142	860	11511233	861	21511241	862	11511332	863
11511431	864	12421142	865	11512142	866	12421241	867	11512241	868	11521133	869
21521141	870	11521232	871	11521331	872	12431141	873	11522141	874	11531132	875
11531231	876	11541131	877	36112112	878	36112211	879	26113112	880	26113211	881
16114112	882	16114211	883	45212111	884	36122111	885	35213111	886	26123111	887
25214111	888	16124111	889	15215111	890	14311151	891	13411142	892	13411241	893
12511133	894	22511141	895	12511232	896	12511331	897	13421141	898	12512141	899
11611124	900	21611132	901	11611223	902	21611231	903	11611322	904	11611421	905
12521132	906	11612132	907	12521231	908	11612231	909	11621123	910	21621131	911
11621222	912	11621321	913	12531131	914	11622131	915	11631122	916	11631221	917
14411141	918	13511132	919	13511231	920	12611123	921	22611131	922	12611222	923
12611321	924	13521131	925	12612131	926	12621122	927	12621221	928		

Table H1. The Bar-Space Sequence Table. Cluster 6

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
21111155	0	31111163	1	11111246	2	21111254	3	31111262	4	11111345	5
21111353	6	31111361	7	11111444	8	21111452	9	11111543	10	61112114	11
11112155	12	21112163	13	61112213	14	11112254	15	21112262	16	61112312	17
11112353	18	21112361	19	61112411	20	11112452	21	51113114	22	61113122	23
11113163	24	51113213	25	61113221	26	11113262	27	51113312	28	11113361	29
51113411	30	41114114	31	51114122	32	41114213	33	51114221	34	41114312	35
41114411	36	31115114	37	41115122	38	31115213	39	41115221	40	31115312	41
31115411	42	21116114	43	31116122	44	21116213	45	31116221	46	21116312	47
11121146	48	21121154	49	31121162	50	11121245	51	21121253	52	31121261	53
11121344	54	21121352	55	11121443	56	21121451	57	11121542	58	61122113	59
11122154	60	21122162	61	61122212	62	11122253	63	21122261	64	61122311	65
11122352	66	11122451	67	51123113	68	61123121	69	11123162	70	51123212	71
11123261	72	51123311	73	41124113	74	51124121	75	41124212	76	41124311	77
31125113	78	41125121	79	31125212	80	31125311	81	21126113	82	31126121	83
21126212	84	21126311	85	11131145	86	21131153	87	31131161	88	11131244	89
21131252	90	11131343	91	21131351	92	11131442	93	11131541	94	61132112	95
11132153	96	21132161	97	61132211	98	11132252	99	11132351	100	51133112	101
11133161	102	51133211	103	41134112	104	41134211	105	31135112	106	31135211	107
21136112	108	21136211	109	11141144	110	21141152	111	11141243	112	21141251	113
11141342	114	11141441	115	61142111	116	11142152	117	11142251	118	51143111	119
41144111	120	31145111	121	11151143	122	21151151	123	11151242	124	11151341	125
11152151	126	11161142	127	11161241	128	12111146	129	22111154	130	32111162	131
12111245	132	22111253	133	32111261	134	12111344	135	22111352	136	12111443	137
22111451	138	12111542	139	62112113	140	12112154	141	22112162	142	62112212	143
12112253	144	22112261	145	62112311	146	12112352	147	12112451	148	52113113	149
62113121	150	12113162	151	52113212	152	12113261	153	52113311	154	42114113	155
52114121	156	42114212	157	42114311	158	32115113	159	42115121	160	32115212	161
32115311	162	22116113	163	32116121	164	22116212	165	22116311	166	21211145	167
31211153	168	41211161	169	11211236	170	21211244	171	31211252	172	11211335	173
21211343	174	31211351	175	11211434	176	21211442	177	11211533	178	21211541	179
11211632	180	12121145	181	22121153	182	32121161	183	11212145	184	12121244	185
22121252	186	11212244	187	21212252	188	22121351	189	11212343	190	12121442	191
11212442	192	12121541	193	11212541	194	62122112	195	12122153	196	22122161	197
61213112	198	62122211	199	11213153	200	12122252	201	61213211	202	11213252	203
12122351	204	11213351	205	52123112	206	12123161	207	51214112	208	52123211	209
11214161	210	51214211	211	42124112	212	41215112	213	42124211	214	41215211	215
32125112	216	31216112	217	32125211	218	31216211	219	22126112	220	22126211	221
11221136	222	21221144	223	31221152	224	11221235	225	21221243	226	31221251	227
11221334	228	21221342	229	11221433	230	21221441	231	11221532	232	11221631	233
12131144	234	22131152	235	11222144	236	12131243	237	22131251	238	11222243	239
21222251	240	11222342	241	12131441	242	11222441	243	62132111	244	12132152	245
61223111	246	11223152	247	12132251	248	11223251	249	52133111	250	51224111	251
42134111	252	41225111	253	32135111	254	31226111	255	22136111	256	11231135	257
21231143	258	31231151	259	11231234	260	21231242	261	11231333	262	21231341	263
11231432	264	11231531	265	12141143	266	22141151	267	11232143	268	12141242	269
11232242	270	12141341	271	11232341	272	12142151	273	11233151	274	11241134	275
21241142	276	11241233	277	21241241	278	11241332	279	11241431	280	12151142	281
11242142	282	12151241	283	11242241	284	11251133	285	21251141	286	11251232	287
11251331	288	12161141	289	11252141	290	11261132	291	11261231	292	13111145	293
23111153	294	33111161	295	13111244	296	23111252	297	13111343	298	23111351	299
13111442	300	13111541	301	63112112	302	13112153	303	23112161	304	63112211	305
13112252	306	13112351	307	53113112	308	13113161	309	53113211	310	43114112	311
43114211	312	33115112	313	33115211	314	23116112	315	23116211	316	12211136	317
22211144	318	32211152	319	12211235	320	22211243	321	32211251	322	12211334	323
22211342	324	12211433	325	22211441	326	12211532	327	12211631	328	13121144	329

Table H1. The Bar-Space Sequence Table. Cluster 6

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
23121152	330	12212144	331	13121243	332	23121251	333	12212243	334	22212251	335
12212342	336	13121441	337	12212441	338	63122111	339	13122152	340	62213111	341
12213152	342	13122251	343	12213251	344	53123111	345	52214111	346	43124111	347
42215111	348	33125111	349	32216111	350	23126111	351	21311135	352	31311143	353
41311151	354	11311226	355	21311234	356	31311242	357	11311325	358	21311333	359
31311341	360	11311424	361	21311432	362	11311523	363	21311531	364	11311622	365
12221135	366	22221143	367	32221151	368	11312135	369	12221234	370	22221242	371
11312234	372	21312242	373	22221341	374	11312333	375	12221432	376	11312432	377
12221531	378	11312531	379	13131143	380	23131151	381	12222143	382	13131242	383
11313143	384	12222242	385	13131341	386	11313242	387	12222341	388	11313341	389
13132151	390	12223151	391	11314151	392	11321126	393	21321134	394	31321142	395
11321225	396	21321233	397	31321241	398	11321324	399	21321332	400	11321423	401
21321431	402	11321522	403	11321621	404	12231134	405	22231142	406	11322134	407
12231233	408	22231241	409	11322233	410	21322241	411	11322332	412	12231431	413
11322431	414	13141142	415	12232142	416	13141241	417	11323142	418	12232241	419
11323241	420	11331125	421	21331133	422	31331141	423	11331224	424	21331232	425
11331323	426	21331331	427	11331422	428	11331521	429	12241133	430	22241141	431
11332133	432	12241232	433	11332232	434	12241331	435	11332331	436	13151141	437
12242141	438	11333141	439	11341124	440	21341132	441	11341223	442	21341231	443
11341322	444	11341421	445	12251132	446	11342132	447	12251231	448	11342231	449
11351123	450	21351131	451	11351222	452	11351321	453	12261131	454	11352131	455
11361122	456	11361221	457	14111144	458	24111152	459	14111243	460	24111251	461
14111342	462	14111441	463	14112152	464	14112251	465	54113111	466	44114111	467
34115111	468	24116111	469	13211135	470	23211143	471	33211151	472	13211234	473
23211242	474	13211333	475	23211341	476	13211432	477	13211531	478	14121143	479
24121151	480	13212143	481	14121242	482	13212242	483	14121341	484	13212341	485
14122151	486	13213151	487	12311126	488	22311134	489	32311142	490	12311225	491
22311233	492	32311241	493	12311324	494	22311332	495	12311423	496	22311431	497
12311522	498	12311621	499	13221134	500	23221142	501	12312134	502	13221233	503
23221241	504	12312233	505	13221332	506	12312332	507	13221431	508	12312431	509
14131142	510	13222142	511	14131241	512	12313142	513	13222241	514	12313241	515
21411125	516	31411133	517	41411141	518	11411216	519	21411224	520	31411232	521
11411315	522	21411323	523	31411331	524	11411414	525	21411422	526	11411513	527
21411521	528	11411612	529	12321125	530	22321133	531	32321141	532	11412125	533
12321224	534	22321232	535	11412224	536	21412232	537	22321331	538	11412323	539
12321422	540	11412422	541	12321521	542	11412521	543	13231133	544	23231141	545
12322133	546	13231232	547	11413133	548	12322232	549	13231331	550	11413232	551
12322331	552	11413331	553	14141141	554	13232141	555	12323141	556	11414141	557
11421116	558	21421124	559	31421132	560	11421215	561	21421223	562	31421231	563
11421314	564	21421322	565	11421413	566	21421421	567	11421512	568	11421611	569
12331124	570	22331132	571	11422124	572	12331223	573	22331231	574	11422223	575
21422231	576	11422322	577	12331421	578	11422421	579	13241132	580	12332132	581
13241231	582	11423132	583	12332231	584	11423231	585	11431115	586	21431123	587
31431131	588	11431214	589	21431222	590	11431313	591	21431321	592	11431412	593
11431511	594	12341123	595	22341131	596	11432123	597	12341222	598	11432222	599
12341321	600	11432321	601	13251131	602	12342131	603	11433131	604	11441114	605
21441122	606	11441213	607	21441221	608	11441312	609	11441411	610	12351122	611
11442122	612	12351221	613	11442221	614	11451113	615	21451121	616	11451212	617
11451311	618	12361121	619	11452121	620	15111143	621	25111151	622	15111242	623
15111341	624	15112151	625	14211134	626	24211142	627	14211233	628	24211241	629
14211332	630	14211431	631	15121142	632	14212142	633	15121241	634	14212241	635
13311125	636	23311133	637	33311141	638	13311224	639	23311232	640	13311323	641
23311331	642	13311422	643	13311521	644	14221133	645	24221141	646	13312133	647
14221232	648	13312232	649	14221331	650	13312331	651	15131141	652	14222141	653
13313141	654	12411116	655	22411124	656	32411132	657	12411215	658	22411223	659

Table H1. The Bar-Space Sequence Table. Cluster 6

bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val	bsbsbsbs	val
32411231	660	12411314	661	22411322	662	12411413	663	22411421	664	12411512	665
12411611	666	13321124	667	23321132	668	12412124	669	13321223	670	23321231	671
12412223	672	22412231	673	12412322	674	13321421	675	12412421	676	14231132	677
13322132	678	14231231	679	12413132	680	13322231	681	12413231	682	21511115	683
31511123	684	41511131	685	21511214	686	31511222	687	21511313	688	31511321	689
21511412	690	21511511	691	12421115	692	22421123	693	32421131	694	11512115	695
12421214	696	22421222	697	11512214	698	21512222	699	22421321	700	11512313	701
12421412	702	11512412	703	12421511	704	11512511	705	13331123	706	23331131	707
12422123	708	13331222	709	11513123	710	12422222	711	13331321	712	11513222	713
12422321	714	11513321	715	14241131	716	13332131	717	12423131	718	11514131	719
21521114	720	31521122	721	21521213	722	31521221	723	21521312	724	21521411	725
12431114	726	22431122	727	11522114	728	12431213	729	22431221	730	11522213	731
21522221	732	11522312	733	12431411	734	11522411	735	13341122	736	12432122	737
13341221	738	11523122	739	12432221	740	11523221	741	21531113	742	31531121	743
21531212	744	21531311	745	12441113	746	22441121	747	11532113	748	12441212	749
11532212	750	12441311	751	11532311	752	13351121	753	12442121	754	11533121	755
21541112	756	21541211	757	12451112	758	11542112	759	12451211	760	11542211	761
16111142	762	16111241	763	15211133	764	25211141	765	15211232	766	15211331	767
16121141	768	15212141	769	14311124	770	24311132	771	14311223	772	24311231	773
14311322	774	14311421	775	15221132	776	14312132	777	15221231	778	14312231	779
13411115	780	23411123	781	33411131	782	13411214	783	23411222	784	13411313	785
23411321	786	13411412	787	13411511	788	14321123	789	24321131	790	13412123	791
23412131	792	13412222	793	14321321	794	13412321	795	15231131	796	14322131	797
13413131	798	22511114	799	32511122	800	22511213	801	32511221	802	22511312	803
22511411	804	13421114	805	23421122	806	12512114	807	22512122	808	23421221	809
12512213	810	13421312	811	12512312	812	13421411	813	12512411	814	14331122	815
13422122	816	14331221	817	12513122	818	13422221	819	12513221	820	31611113	821
41611121	822	31611212	823	31611311	824	22521113	825	32521121	826	21612113	827
22521212	828	21612212	829	22521311	830	21612311	831	13431113	832	23431121	833
12522113	834	13431212	835	11613113	836	12522212	837	13431311	838	11613212	839
12522311	840	11613311	841	14341121	842	13432121	843	12523121	844	11614121	845
31621112	846	31621211	847	22531112	848	21622112	849	22531211	850	21622211	851
13441112	852	12532112	853	13441211	854	11623112	855	12532211	856	11623211	857
31631111	858	22541111	859	21632111	860	13451111	861	12542111	862	11633111	863
16211132	864	16211231	865	15311123	866	25311131	867	15311222	868	15311321	869
16221131	870	15312131	871	14411114	872	24411122	873	14411213	874	24411221	875
14411312	876	14411411	877	15321122	878	14412122	879	15321221	880	14412221	881
23511113	882	33511121	883	23511212	884	23511311	885	14421113	886	24421121	887
13512113	888	23512121	889	13512212	890	14421311	891	13512311	892	15331121	893
14422121	894	13513121	895	32611112	896	32611211	897	23521112	898	22612112	899
23521211	900	22612211	901	14431112	902	13522112	903	14431211	904	12613112	905
13522211	906	12613211	907	32621111	908	23531111	909	22622111	910	14441111	911
13532111	912	12623111	913	16311122	914	16311221	915	15411113	916	25411121	917
15411212	918	15411311	919	16321121	920	15412121	921	24511112	922	24511211	923
15421112	924	14512112	925	15421211	926	14512211	927	33611111	928		

AIM Glossary of Terms

2D Matrix Symbology — An arrangement of regular polygon shaped cells where the center to center distance of adjacent elements is uniform. The arrangement of the elements represents data and/or symbology functions. Matrix symbols may include recognition patterns which do not follow the same rule as the other elements within the symbol.

ABC Codabar — A variation of Codabar symbology, developed by the American Blood Commission for blood labeling, that incorporates a unique field concatenation capability based on patterns of start and stop codes in adjacent bar code symbols.

Alphanumeric — The character set which contains the letters A-Z and numbers 0-9.

ANSI — The American National Standards Institute, a non-governmental organization responsible for the development of voluntary industry standards.

Aperture — The effective opening in an optical system (scanner) that establishes the field of view.

ASCII — The character set and code described in American National Standard Code for Information Interchange, ANSI X3.4. The ASCII character set is used for information interchange between data processing systems, communication systems and associated equipment. The ASCII set consists of 128 characters including both control and printing characters.

Autodiscrimination — The ability of bar code reading equipment to recognize and correctly decode more than one symbology.

Bar — An element of lower reflectance in a printed bar code symbol.

Bar Code — An array of parallel rectangular bars and spaces that together represent data elements or characters in a particular symbology. The bars and spaces are arranged in a predetermined pattern following unambiguous rules defined by the symbology.

Bar Height — The dimension of a bar measured perpendicular to the bar width.

Bidirectionally Decodable — A bar code symbology capable of being read successfully independent of scanning direction.

Character

Data Character — A letter, digit or other member of the ASCII character set.

Symbol Character — A unique bar and/or space pattern which is defined for that symbology. There is not necessarily a one-to-one or unique correlation between symbol characters and data characters. Depending on the symbology, symbol characters may have a unique associated symbol value.

Character Self-Checking —The feature which allows a bar code reader to determine if a scanned group of elements is a valid symbol character. If a symbology is described as being character self-checking, a single printing defect (edge error) in any symbol character does not produce another valid character.

Character Set — Those characters available for encodation in a particular automatic identification technology.

Check Character — A character included within a symbol whose value is used for the purpose of performing a mathematical check to ensure the symbol has been decoded correctly.

Check Digit — See "Check Character."

Clear Area — See "Quiet Zone."

Code Set — The specific assignment of data characters to symbol characters.

Continuous Code — A bar code symbology where all spaces within the symbol are parts of characters, e.g., Interleaved 2-of-5. There is no intercharacter gap in a continuous bar code symbology.

CPI — Characters per inch (see "Density").

Data Character —See "Character."

Decoder — As part of a bar code reading system, the electronic package which receives the signals from the scanner, performs the algorithm to interpret the signals into meaningful data and provides the interface to other devices.

Density — The number of data characters which can be represented in a linear unit of measure. Bar code density is often expressed in characters per inch (CPI).

Diffuse Reflection — The component of reflected light which emanates in all directions from the reflecting surface.

Discrete Code — A bar code symbology where the spaces between characters (intercharacter gaps) are not part of the code, e.g., Code 39.

Edge-to-Edge Symbologies — A bar code symbology which can be decoded using edge-to-similar-edge measurements, such as from the start of one bar to the start of another or from the end of one bar to the end of another. Such symbologies are relatively insensitive to uniform bar growth or shrinkage. Code 93, Code 128, Code 49, and Code 16K are examples of edge-to-edge symbologies.

Element — A single bar or space.

Element Width — The thickness of an element measured from the edge closest to the symbol start character to the trailing edge of the same element.

Erasure Correction — The use of the error correction characters to correct data errors which have known locations (these locations may have insufficient contrast in the image, may fall outside of the image field, or may have incorrect parity for symbologies with symbol character parity.) Only one error correction character is required to correct each erasure.

Error Correction — The use of the error correction characters to locate and correct data errors which have unknown locations. Two error correction characters are required to correct each error (one to locate the error and the second to correct the character's data.)

Error Correction Characters — Symbol characters which are reserved for error correction and/or error detection. These characters are calculated mathematically from the other symbol characters. Many linear symbologies have one error correction character. It is reserved for error detection and is called a check character. Two dimensional symbologies may have many error correction characters which can be used for error correction, erasure correction, and error detection.

Error Detection — The use of the error correction characters to detect that the number of errors in the symbol exceeds the error correction capacity. Error detection will keep the symbol from being decoded as erroneous data. Error correction characters can be reserved for error detection. These characters then function as check characters. The error correction algorithm can also provide error detection by detecting invalid error correction calculation results. Erasure correction does not provide any error detection.

Extended ASCII — Those characters represented by values 128-255. These values represent different character sets, called Code Pages, in different computer systems. Code Pages are defined in the ISO 8859 standard.

Font — A specific size and style of type.

Galois Field —A finite set of numbers which are closed under their defined operations of addition, subtraction, multiplication and division.

Human Readable Interpretation —The letters, digits or other characters associated with specific symbol characters and printed along with the bar code symbol.

Informative — Information provided as guidance but which is not part of the standard. See also "Normative."

Intercharacter Gap — The space between two adjacent bar code characters in a discrete code. For example, the space between characters in Code 39.

ISBT 128 — A variation of Code 128 symbology, developed by the International Society for Blood Transfusion for blood labeling, that incorporates a unique field concatenation capability based on specific data characters contained in adjacent symbols.

Module — The narrowest nominal width unit of measure in a symbol. One or more modules are used to construct an element.

Modulo — An arithmetic operation in which the result is the remainder after division.

Multi-Row Symbology — Symbologies where a long symbol is broken into sections and "stacked" one upon another similar to sentences in a paragraph, e.g., Code 16K and Code 49.

N — The wide-to-narrow element width ratio for two-width symbologies.

Nominal — The intended value for a specified parameter. Tolerances are specified as positive and negative deviations from this value.

Normative — Information which is part of the standard. See also "Informative."

Numeric — A character set that includes only the numbers 0-9.

Overhead — In a bar code system, the fixed number of characters required for start, stop and checking in a given symbol. For example, a symbol requiring a start/stop and two check characters contains four characters of overhead. Thus, to encode three characters, seven characters are required to be printed.

Quiet Zone — Spaces preceding the start character of a symbol and following the stop character. Sometimes called the "Clear Area."

Reader — A device comprised of a scanner and a decoder.

Reflectance — The measure of the amount of light reflected from an illuminated surface.

Resolution — In a bar code system, the narrowest element dimension which can be distinguished by a particular reading device or printed with a particular device or method.

Scanner — An electronic device to acquire and convert reflected light from the bars and spaces of a symbol into electrical signals for processing by the decoder.

Scan Profile — A record of the reflectance measured as a function of position during a scan across the entire bar code symbol.

Show-Through — The generally undesirable property of a substrate that permits underlying markings to be seen. This may adversely affect reader performance.

Space — An element of higher reflectance in a bar code symbol.

Spectral Band — The range of acceptable wavelengths of light, stated in nanometers, used by scanners or verification devices.

Specular Reflection — The mirror-like reflection of light from a surface where the angle of incidence equals the angle of reflection.

Spot — The undesirable presence of an area of low reflectance in a space.

Stacked Codes — See "Multi-row Symbology."

Start-Stop Character or Pattern — Special bar code patterns that indicate the ends of the symbol as well as scan direction. The start character is normally at the left-hand end of a horizontally oriented symbol. The stop character is normally at the right-hand end of a horizontally oriented symbol.

Symbol — A combination of bar code characters including start/stop characters, quiet zones, data characters, and check characters required by a particular symbology, which forms a complete, scannable entity.

Symbology — A specific bar code "language." Eleven symbologies are defined below.

- ***Codabar*** — A mostly-numeric bar code symbology. See AIM Uniform Symbology Specification for Codabar.
- ***Code 128*** — A full ASCII bar code symbology capable of encoding all 128 ASCII characters as well as the extended ASCII characters. See AIM Uniform Symbology Specification for Code 128.
- ***Code 16K*** — A full ASCII, multi-row bar code symbology capable of encoding all 128 ASCII characters. See AIM Uniform Symbology Specification for Code 16K.
- ***Code 39*** — (3 of 9 Code) A full alphanumeric bar code symbology. See AIM Uniform Symbology Specification for Code 39.
- ***Code 49*** — A full alphanumeric, multi-row bar code symbology capable of encoding all 128 ASCII characters. See AIM Uniform Symbology Specification for Code 49.
- ***Code 93*** — A full alphanumeric bar code symbology capable of encoding all 128 ASCII characters. See AIM Uniform Symbology Specification for Code 93.
- ***Code One*** — A full ASCII 2D matrix bar code symbology capable of encoding all 128 ASCII characters as well as the extended ASCII characters. See AIM Uniform Symbology Specification for Code One.
- ***EAN*** — A numeric bar code symbology that is the international standard for bar code identification goods that will be scanned at point-of-sale.Similar to U.P.C.
- ***Interleaved Two-of-Five*** — (I-2/5, ITF) A numeric bar code symbology. See AIM Uniform Symbology Specification for Interleaved 2-of-5.
- ***PDF417***— A full ASCII multi-row bar code symbology capable of encoding all 128 ASCII characters as well as the extended ASCII characters. See AIM Uniform Symbology Specification for PDF417
- ***U.P.C.*** — Universal Product Code, a numeric bar code symbology (administered by the Uniform Code Council) that is the North American standard for bar code identification of goods that will be scanned at point-of-sale. Similar to EAN.

Symbology Identifier — An optional three character code which may prefix transmitted data from a bar code reader indicating the symbology read and any options enabled in the reader or special features of a symbology encountered (e.g., presence of FNC1 in Code 128).

Two-Dimensional Matrix Symbology — See 2D Matrix Symbology

Two-Width Symbology — A bar code symbology whose bars and spaces are characterized simply as "wide" or "narrow." Codabar, Code 39, and Interleaved 2-of-5 are examples of two-width symbologies.

Void — The undesirable presence of an area of high reflectance in a bar.

"X" Dimension — The nominal dimension of the narrow bars and spaces in a bar code symbol.

"Z" Dimension — The average measured width dimension of the narrow bars and spaces in a bar code symbol.

Z — ((average narrow bar)+(average narrow space))/2.

Reference Documents

AIM USA "Guidelines on Symbology Identifiers" (Item X-4)
AIM USA Code 49 Developers Diskette (Item Z-40)
AIM USA Code One Developers Diskette (Item Z-46)
AIM USA PDF417 Developers Diskette (Item Z-47)
AIM USA "Uniform Symbology Specifications"
Interleaved 2-of-5 (Item X5-1)
Code 39 (Item X5-2)
Codabar (Item X5-3)
Code 128 (Item X5-4)
Code 93 (Item X5-5)
Code 16K (Item X5-6)
Code 49 (Item X5-7)
Code One (Item X5-8)
PDF417 (Item X5-9)

Available from:
AIM USA
634 Alpha Drive
Pittsburgh, PA 15238-2802
Phone: 412-963-8588
Fax: 412-963-8753

ANSI X3.182, "Bar Code Print Quality — Guideline"
Available from:
The American National Standards Institute
11 West 42nd St., 13th Floor
New York, NY 10036
Phone: 212-642-4900
Fax: 212-398-0023
Also available from AIM USA (address above).

CEN Specifications
Available from:
Commission for European Normalization
36 Rue de Stassart
B-1050 Bruxelles
Belgium
Phone: 32 2 519 6811
Fax:32 2 519 6819

EAN Specifications
Available from:
EAN International
Rue Royale 29
B-1000 Bruxelles
Belgium
Phone: 32 2 218 76 74
Fax: 322 218 75 85

Error Control Coding, Fundamentals and Applications, Lin, Shu and Costello Jr., D.J. Prentice-Hall, 1983.

ISBT 128 Symbology and Application Specification
Available from:
American Association of Blood Banks,
Regulatory Affairs Department
8101 Glenbrook Road
Bethesda, MD 20814
Phone: 301-215-6489
Fax: 301-907-6895

ISO 8859 Information Processing: 8-bit Code Processing Character Set
Parts 1-4 Latin
Part 5 Latin and Cyrillic
Part 6 Latin and Arabic
Part 7 Latin and Greek
Part 8 Latin and Hebrew
Part 9 Latin Alphabet #5
Available from:
The American National Standards Institute
11 West 42nd St., 13th Floor
New York, NY 10036
Phone: 212-642-4900
Fax: 212-398-0023

Theory and Practice of Error Control Codes, Blahut, Richard. Addison-Wesley, 1983.

U.P.C. Specifications
Available from:
The Uniform Code Council
8163 Old Yankee Road, Suite J
Dayton, OH 45458
Phone: 513-435-3870
Fax: 513-435-4749