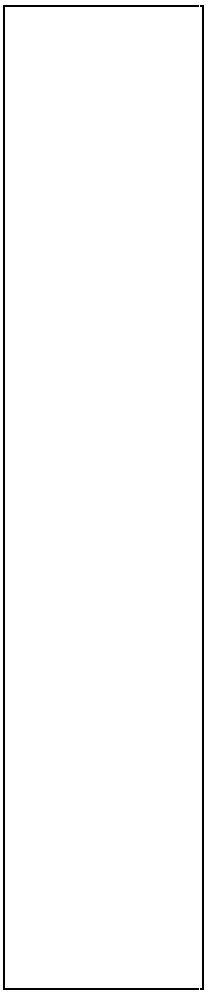
The drawaing contained in this Recommendation have been done in Autocad

Fig. 4-2/T.150/T0803930-89 = 15 cm



6.3 The structure for opcode encoding is given in Figure 4—3/T.150.

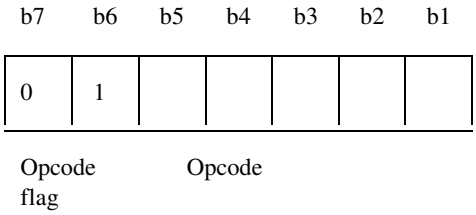


FIGURE 4—3/T.150

Opcode encoding structure

For single—byte opcodes the opcode length indicator bit b5 is ZERO. Bits b4 to b1 represent the opcode, i.e. the opcodes are taken from column 2. For two—byte opcodes the opcode length indicator bit b5 of the first byte is ONE. Bits b4 to b1 of the first byte and bits b5 to b1 of the second byte represent the opcode, i.e. the first byte of the opcode is taken from column 3, the second byte is taken from column 2 or 3.

6.4 The general format for operand encoding is given Figure 4—4/T.150.

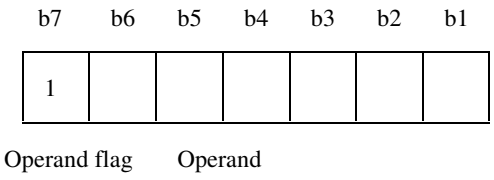


FIGURE 4—4/T.150

Operand encoding structure

The operand part of a primitive may contain one or more operands, each operand consisting of one or more bytes.

6.5 The encoding of the operands may make use of the following DATA TYPES:

— point P
— colour index CI
— integer number I
— real number R

These data types are coded according to the basic format.

6.6 The basic format for operand encoding is given in Figure 4—5/T.150.

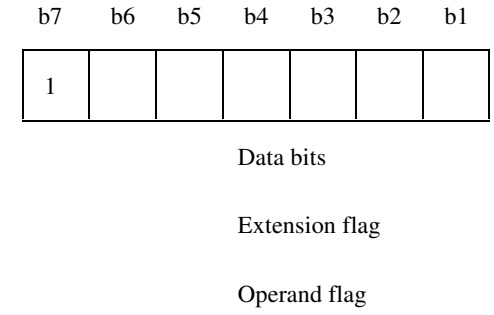


FIGURE 4—5/T.150

The basic format structure

6.7 Each basic format operand is coded as a sequence of one or more bytes.

Bit b6 of each byte is the extension flag. For single byte operands, the extension flag is ZERO. In multiple byte operands, the extension flag is ONE in all bytes, except the last byte, where it is ZERO.

The most significant part of the operand is coded in the first byte. The least significant part of the operand is coded in the last byte.

In data types P, I and R, bit 5 of the first byte represents the sign bit. Bit 5 = 0 corresponds to positive values. Following data bits represent a binary number. Bit 1 of the last byte is to be considered as the unit of this binary representation.

Data type CI is coded in one single byte (b6 = 0). Bits 5 to 1 give the binary representation of colour indexes.

The coding proposed here for data types P, CI, I and R although derived from Recommendation T.101, Annex C, is a simplified version of the encoding method for these data types, which is only valid after adequate initialization of the protocol description primitives.

6.8 The position of a single point, as well as the position of the first point of a sequence is given by absolute coordinate values x0 and y0, expressed in grid units GU. The encoding structure is given in Figure 4—6/T.150.

6.9 If the coordinate value fits in a single byte, the extension flag is set to ZERO. In that case the x—value is contained in one byte, the y—value is contained in the subsequent byte(s).

6.10 If the coding of a coordinate value requires more than one byte, the complete position information is contained in two contiguous series of bytes. The first series contains the x—value, the second series contains the y—value.

6.11 Each such series consists of contiguous bytes. The extension flag of all bytes in one series, except the last byte, is set to ONE.

The extension flag of the last byte in the series is set to ZERO.

7 Incremental mode coding format

7.1 For incremental mode, the presentation elements trace and closed area are coded, according to the following sequence:

— first point's position;

— DCC introducer;

— incremental sequence.

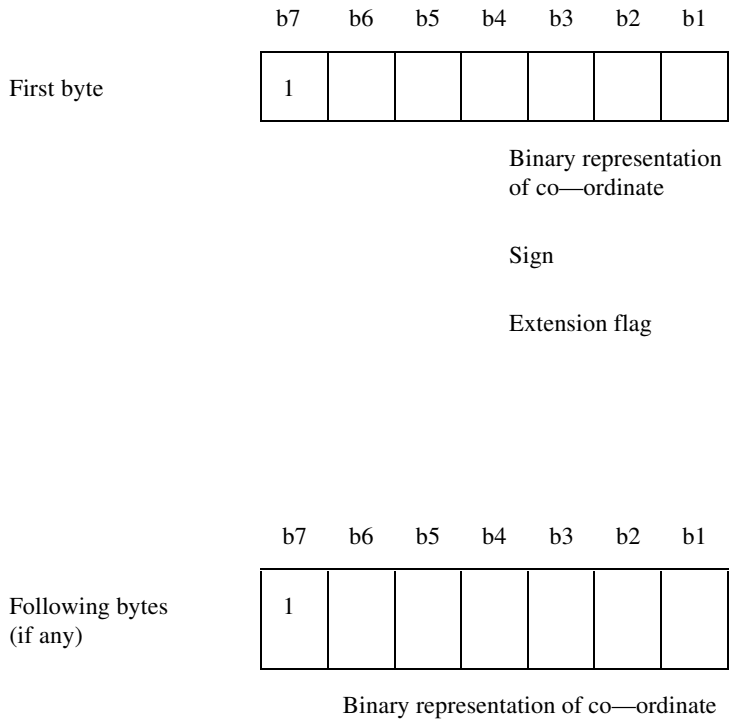


FIGURE 4—6/T.150

Absolute co—ordinate encoding structure

7.2 The position of the first point is coded as defined in §§ 6.8 to 6.11.

7.3 DCC is the abbreviation of differential chain code. The DCC introducer is required in order to preserve compatibility with Recommendation T.101.

7.4 The DCC introducer consists of two bytes, see Figure 4—7/T.150.

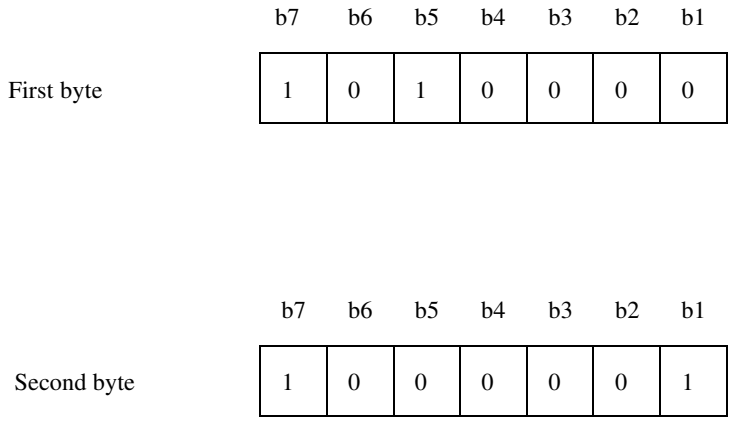


FIGURE 4—7/T.150

DCC introducer encoding

7.5 The format for encoding of the incremental sequence is given in Figure 4—8/T.150.

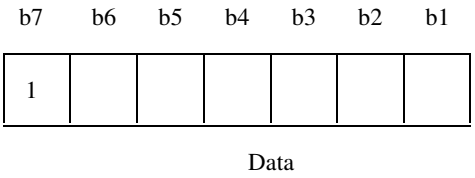


FIGURE 4—8/T.150

Incremental sequence encoding

7.6 The incremental sequence encoding uses variable length words. To accommodate these words in a sequence of bytes as given in Figure 4—8/T.150, the bit positions b6 to b1 of successive bytes are used as if they constitute a continuous bit channel. The first bit of the first variable length word is placed at b6, and so on.

7.7 The end of the incremental sequence is identified by the end of block code. The remaining bit positions between end of block code and the next byte boundary have no meaning. They will be ignored.

8 Displacement mode coding format

8.1 For displacement mode, the presentation elements trace, closed area and marker are coded according to the following sequence:

— first points position;

— following points.

8.2 For points after the first point in a point list, each displacement is measured with respect to the preceding point of the point list. These displacements are coded as the first point of the list of points.

9 Encoding of the primitives

9.1 The opcodes are defined in Table 4—4/T.150. (The notation x/y means column x, row y, in a 8 × 16 code table.)

9.2 The notational conventions used are defined in Table 4—5/T.150.

In the further §§ 9.3 to 9.5 the encoding of each primitive is defined as well as the order of the parameters, along with their specific data type.

9.3 The presentation elements trace, closed area and marker are encoded as follows:

Trace

<Trace opcode: 2/0> <point: point list> (2)

OR

<Trace opcode: 2/0> <point: first point>

<DCC introducer: 5/0, 4/1> <Incremental sequence>

Closed area

<Closed area opcode: 2/1> <point: point list> (3)

OR

<Closed area opcode: 2/1>

<Point: first point> <DCC introducer: 5/0, 4/1>

<Incremental sequence>

Marker

<Marker opcode: 3/2, 2/11, 5/2> <point: position>

OR

<Marker opcode: 3/2, 2/11, 5/2> <point: first point> (1)

<DCC introducer: 5/0 4/1> <Incremental sequence>

Clear

<Clear opcode: 3/2, 2/0, 4/0>

TABLE 4—4/T.150

Incremental Trace Coding opcodes

Element		Code		
		byte 1	byte 2	byte 3
Presentation elements	Trace	2/0	—	—
	Closed area	2/1	—	—
	Marker	3/2	2/11	5/2
	Clear	3/2	2/0	4/0
Attribute setting	Set trace thickness	3/1	2/1	
	Set trace texture	3/1	2/2	
	Set trace colour index	3/1	2/0	
	Set closed area interior style	3/1	2/5	
	Set closed area style index	3/1	2/6	
	Set closed area colour index	3/1	2/4	
	Set marker type	3/1	2/12	
	Set marker size	3/1	2/13	
	Set marker colour index	3/1	2/11	
Protocol descriptor	Set domain ring	3/2	2/4	
	Set co—ordinate precision	3/2	2/9	

TABLE 4—5/T.150

Notational conventions

Item	Meaning
<symbols>	1 occurrence
<symbols> (n)	n or more occurrences, with n 1
[comments]	Explanation of a production
<x : y>	Construction x with meaning y.

9.4 The attribute setting primitives are encoded as follows:

Trace thickness

<Set trace thickness opcode: 3/1, 2/1>
<real = trace thickness scale factor>

Trace texture

<Set trace texture opcode: 3/1, 2/2>
<integer: trace texture> =

<integer: 0> [SOLID]
<integer: 1> [DASHED]
<integer: 2> [DOTTED]
<integer: 3> [DASHED DOTTED]
<all other values> [RESERVED]

Trace colour

<Set trace colour index opcode: 3/1, 2/0>
<colour index: trace colour index> =

<index: 0> [black]

<index: 1> [red]
<index: 2> [green]
<index: 3> [yellow]
<index: 4> [blue]
<index: 5> [magenta]
<index: 6> [cyan]
<index: 7> [white]

Closed area interior style

<Set closed area interior style opcode: 3/1, 2/5>
<integer: fill area interior style>

<integer: 0>	[HOLLOW]
<integer: 1>	[SOLID]
<integer: 2>	[PATTERN]
<integer: 3>	[HATCH]
<all other values>	[RESERVED]

Closed area style index

<Set closed area style index opcode: 3/1, 2/6>
<integer: closed area style index> = interior style HATCH

<integer: 0>	[vertical lines]
<integer: 1>	[horizontal lines]
<integer: 2>	[45 degrees lines]
<integer: 3>	[—45 degrees lines]
<integer: 4>	[closed lines, vertical and horizontal]
<integer: 5>	[crossed lines, 45 and —45 degrees]
<all other values>	[reserved]

Closed area colour index

<Set closed area colour index opcode: 3/1, 2/4>
<colour index: closed area colour index> =

<index: 0> [black]
<index: 1> [red]
<index: 2> [green]
<index: 3> [yellow]
<index: 4> [blue]
<index: 5> [magenta]
<index: 6> [cyan]
<index: 7> [white]

Marker type

<Set marker type opcode: 3/1, 2/12>
<integer: market type> =

<integer: 0>	[DOT]
<integer: 1>	[PLUS SIGN]
<integer: 2>	[ASTERISK]
<integer: 3>	[CIRCLE]
<integer: 4>	[DIAGONAL CROSS]
<all other values>	[RESERVED]

Marker size

<Set marker size scale factor opcode: 3/1, 2/13>
<real: marker size scale factor>

Marker colour

<Set marker colour index opcode: 3/1, 2/11>
<colour index: marker colour index> =

<index: 0> [black]
<index: 1> [red]

<index: 2> [green]
<index: 3> [yellow]
<index: 4> [blue]
<index: 5> [magenta]
<index: 6> [cyan]
<index: 7> [white]

9.5 The protocol descriptor primitives are encoded as follows:

Set domain ring
<Set domain ring opcode: 3/2, 2/4>
<integer: angular resolution factor>
<integer: basic radius of the ring>

Set coordinate precision
<Set coordinate precision opcode: 3/2, 2/9>
<integer: magnitude code> [4]
<integer: granularity code> [1 —9, —10, —11]
<integer: default exponent> [1 —9, —10, —11]
<integer: explicit exponent allowed> [1]

9.6 *Remark 1* — The default value for "granularity code" and "default exponent" is —9.

All the described coding is correct if the values for granularity and for default exponent are equal, and if the value of "explicit exponent allowed" is 1 (i.e. forbidden).

Remark 2 — The primitive set coordinate precision has no effect on reals (e.g. thickness scale factor). Reals are expressed (by default) in fractions of 2 ** —9.

10 Example of differential chain coding

The trace of handwritten information is shown in Figure 4—9/T.150, where (P1, P2, P3) are the sampled points. These points are encoded in the incremental mode; the value of the ring radius is R = 2 and the value of the ring angular resolution factor is p = 0, so the number of reference points on the ring is N = 8 * R/(2 ** p) = 16. On Figure 4—9/T.150, for each point, the corresponding ring with several reference points is shown.

After coding, the new list of points is (Q1, Q2, Q3, Q4, Q5). The coordinate and reference points of Pi and Qj are shown on Table 4—6/T.150. The difference chain code bitstream is shown on Figure 4—10/T.150. This bitstream with the appropriate DCC header could be a block.

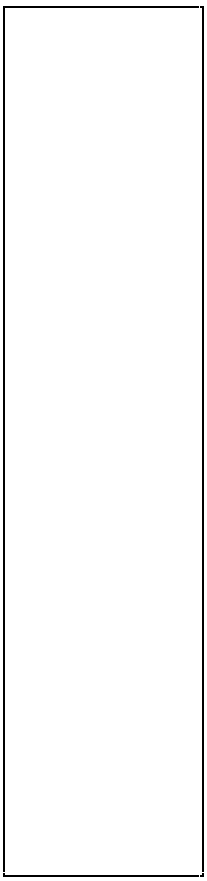
The initial trace can also be directly encoded in the displacement mode. Figure 4—11/T.150 shows how the list of points (P_1, P_2, P_3) is encoded in this mode.

TABLE 4—6/T.150

T.150 coordinate values and reference point number

	X	Y		X	Y	reference point number
P1	10	10	Q1	10	10	—
			Q2	12	12	+2
P2	13	14	Q3	13	14	+1
			Q4	14	12	—6
P3	14	10	Q5	14	10	—1

Fig. 4—9/T.150/T0803940-89 = 12 cm



Trace opcode | Q1 (absolute coordinates) | | DCC introducer | |

00100000 | 01001010 | 01001010 | 01001010 | 01001010

value +10 | value +10

Q2 | Q3 | Q4 | Q5 | End of Bloc

01110010 | 01111100 | 01110111 | 01111111 | 01110000

+2 | +1 | —6 | —1 | EOB

Byte sequence: 2/0 4/10 4/10 5/0 4/1 7/2 7/12 7/7 7/15 7/0

FIGURE 4—10/T.15

DCC coded bitstream

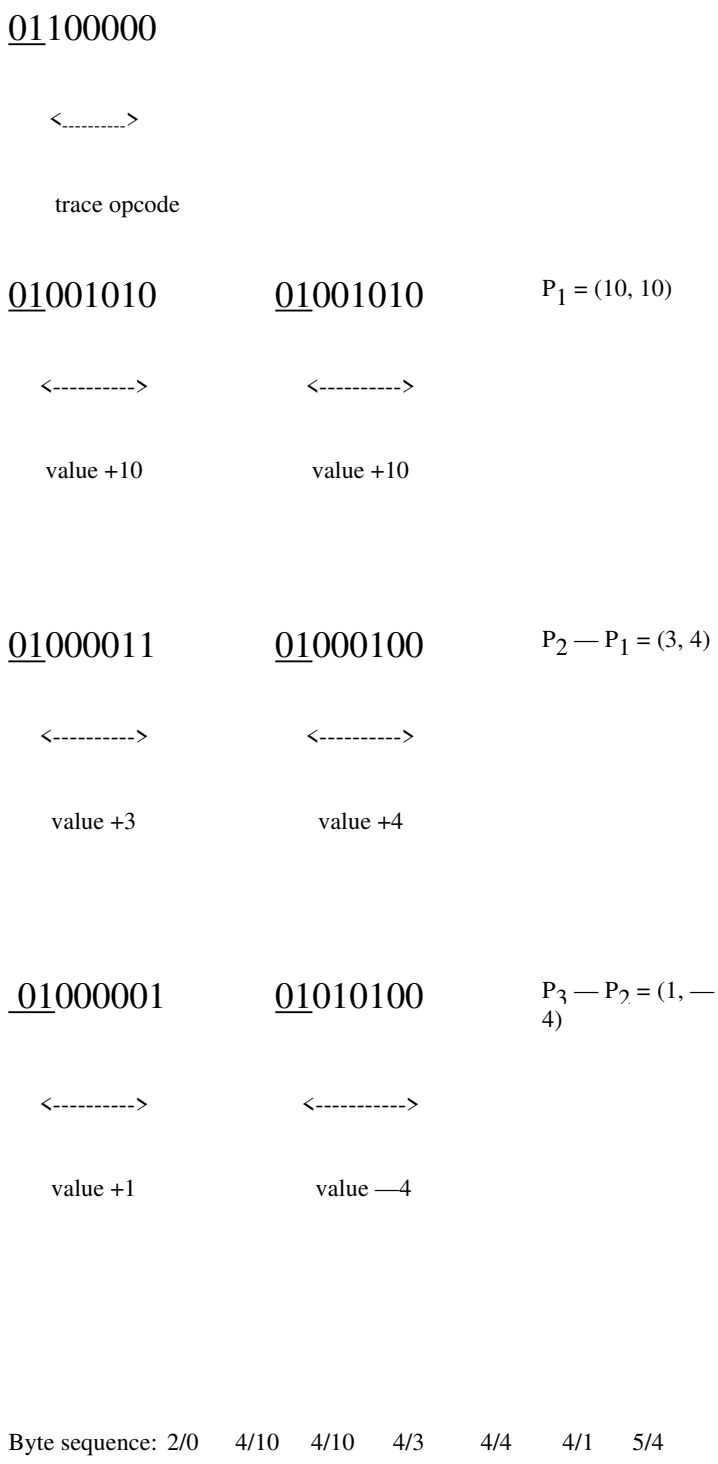


FIGURE 4—11/T.150

Displacement mode coded bitstream