INTERNATIONAL STANDARD

ISO
9282-1

First edition
1988-09-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Information processing — Coded representation of pictures —

Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment

Traitement de l'information — Représentation codée de l'image —

Partie 1: Principes de codage pour la représentation d'image dans un environnement codé à 7 et à 8 éléments

Reference number
ISO 9282-1 : 1988 (E)

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 9282-1 was prepared by Technical Committee ISO/TC 97, *Information processing systems.*

ISO 9282 consists of the following parts, under the general title *Information processing — Coded representation of pictures*:

— *Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment*

Annex A forms an integral part of this part of ISO 9282.

Introduction

This International Standard provides standard methods for picture coding in order to assist in coding system design and to prevent a proliferation of different unrelated coding techniques.

This part of ISO 9282 provides a coding scheme for the representation of pictures that can be generated by the majority of computer graphics applications; based on a 7-bit structure, this coding scheme may be used in a 7-bit or 8-bit environment.

This page intentionally left blank

Information processing — Coded representation of pictures —

Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment

1 Scope

This part of ISO 9282 defines

— the coding principles to be used in interchanging pictures consisting of graphic images in a 7-bit or 8-bit environment;

— the data structures to be used to represent the primitives describing a picture;

— the general datatypes which can be used as operands within a primitive.

This part of ISO 9282 does not deal with the presentation semantics of pictures. These are defined in the related International Standards.

This part of ISO 9282 applies to the data streams containing data structured in accordance with picture coding methods defined in ISO 9281.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 9282. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 9282 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 646 : 1983, *Information processing — ISO 7-bit coded character sets for information interchange.*

ISO 2022 : 1986, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques.*

ISO 6429 : 1983, *Information processing — ISO 7-bit and 8-bit coded character sets — Additional control functions for character imaging devices.*

ISO 9281 : —[1)], *Information processing — Identification of picture coding methods.*

1) To be published.

3 Definitions and notation

3.1 Definitions

For the purposes of this part of ISO 9282, the following definitions apply:

3.1.1 bit combination; byte: An ordered set of bits that represents an opcode or an operand, or used as a part of the representation of an opcode or an operand.

3.1.2 code: A set of unambiguous rules that establishes a one-to-one relationship between each opcode or operand of a set and their coded representation by one or more bit combinations within that set.

3.1.3 code table: A table showing the general distribution of opcodes and operands to bit combinations in a code.

3.1.4 opcode: A one or multi-byte coded representation that identifies a function required by a picture standard.

An opcode may be followed by zero or more operands.

3.1.5 opcode table: A table showing the function allocated to each bit combination reserved for opcodes.

3.1.6 operand: A single or multiple coded representation used to specify the parameters required by an opcode.

3.2 Notation

3.2.1 7-bit byte

The bits of a 7-bit byte are identified by b_7, b_6, b_5, b_4, b_3, b_2 and b_1 where b_7 is the highest-order, or most-significant bit and b_1 is the lowest order, or least-significant bit.

The bit combinations are identified by notations of the form x/y, where x is a number in the range 0 to 7 and y is a number in the range 0 to 15, corresponding to the column and row designation, respectively, of a code table.

The correspondence between the notations of the form x/y and the bit combinations consisting of the bits b_7 to b_1, is as follows:

— x is the number represented by b_7, b_6 and b_5 where these bits are given the weights 4, 2 and 1, respectively.

— y is the number represented by b_4, b_3, b_2 and b_1 where these bits are given the weights 8, 4, 2 and 1, respectively.

3.2.2 8-bit byte

The bits of an 8-bit byte are identified by b_8, b_7, b_6, b_5, b_4, b_3, b_2 and b_1, where b_8 is the highest-order, or most-significant bit and b_1 is the lowest-order, or least-significant bit.

The bit combinations are identified by notations of the form xx/yy, where xx and yy are numbers in the range 00 to 15. The correspondence between the notations of the form xx/yy and the bit combinations consisting of the bits b_8 to b_1, is as follows:

— xx is the number represented by b_8, b_7, b_6 and b_5 where these bits are given the weights 8, 4, 2 and 1, respectively.

— yy is the number represented by b_4, b_3, b_2 and b_1 where these bits are given the weights 8, 4, 2 and 1, respectively.

3.2.3 Byte interpretation

Bits within a byte may be interpreted to represent numbers in binary notation by attributing the following weights to the individual bits:

Bits of a 7-bit byte	—	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Bits of an 8-bit byte	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Weight	128	64	32	16	8	4	2	1

Using these weights, the bit-combinations of the 7-bit byte represent numbers in the range 0 to 127. The bit-combinations of the 8-bit byte represent numbers in the range 0 to 255.

3.3 Layout of the code table

3.3.1 7-bit representation

In a 7-bit representation, a code table consists of 128 positions arranged in 8 columns and 16 rows. The columns are numbered 0 to 7 and the rows 0 to 15.

The code table positions are identified by notations of the form x/y, where x is the column number and y is the row number.

The positions of the code table are in one-to-one correspondence with the bit combinations. The notation of a code table position, of the form x/y, is the same as that of the corresponding bit combination.

3.3.2 8-bit representation

In an 8-bit representation, a code table consists of 256 positions arranged in 16 columns and 16 rows. The columns are numbered and the rows are numbered 00 to 15.

The code table positions are identified by notations of the form xx/yy, where xx is the column number and yy is the row number.

The positions of the code table are in one-to-one correspondence with the bit combinations. The notation of a code table position, of the form xx/yy, is the same as that of the corresponding bit combination.

4 Encoding principles

This part of ISO 9282 deals with the encoding principles of

— the opcodes of the primitives;

— the operands of the primitives.

All such encoding is defined in terms of a 7-bit byte. When used in an 8-bit environment, bit b_8 of each byte shall be zero (except within the "string" format).

Each primitive is coded according to the following rules:

— a primitive is composed of one opcode and zero or more operands as required;

— the opcodes are encoded in column 2 or 3 of the Code Table (table 1);

— operands are encoded in columns 4 to 7. (However, the coded representation of a "string" operand may include bit combinations from other columns of the Code Table — see the description of string format in 6.2.3.)

5 Encoding principles for opcodes

5.1 General

One of two encoding techniques may be used in the organization of opcodes for the definition of a code table:

— if the number of opcodes required in a particular standard built upon these coding principles is less than or equal to 32, compact structure may be used as described in 5.2;

— if a greater number of opcodes is required then an extensible structure may be used as described in 5.3.

This permits the definition of more efficient code tables when the number of opcodes is small as well as allowing the development of standards which require an unlimited number of opcodes.

The identification of either opcode structure is achieved through the identification mechanism defined in ISO 9281.

Table 1 — Code table as used for picture coding

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0								
0	0	0	1	1								
0	0	1	0	2								
0	0	1	1	3								
0	1	0	0	4								
0	1	0	1	5								
0	1	1	0	6								
0	1	1	1	7								
1	0	0	0	8								
1	0	0	1	9								
1	0	1	0	10								
1	0	1	1	11								
1	1	0	0	12								
1	1	0	1	13								
1	1	1	0	14								
1	1	1	1	15								
					Reserved for control functions		Opcodes		Operands			

5.2 Compact opcode encoding

In the case where 32 or less opcodes are needed the encoding of opcodes is simply accomplished by assigning a code table position to each of the opcodes from the 32 code table positions in columns 2 and 3 of the code table. The general structure of an opcode of this type is shown in figure 1 below.

5.3 Extensible opcode encoding

In the case where an unlimited number of opcodes may be needed the encoding of opcodes requires that the opcodes be divided into

— a basic opcode set and,

— an extension opcode set.

The description of the encoding technique for the basic opcode set is given in 5.3.1. The description of the extension mechanism is given in 5.3.2.

5.3.1 Encoding technique of the basic opcode set

The basic opcode set consists of single-byte and double-byte opcodes. The general structure of such opcodes is shown in figure 2.

For single-byte opcodes, the opcode length indicator, bit b_5, is ZERO (opcodes of column 2), bits b_4 to b_1 are used to encode the opcode.

For double byte opcodes the opcode length indicator, bit b_5, of the first byte is ONE. Bits b_4 to b_1 of the first byte and bits b_5 to b_1 of the second byte are used to encode the opcode.

The bit representation 3/15, EXTEND OPCODE SPACE, (EOS), is used in a different sense (see 5.3.2).

This encoding technique can thus provide a basic opcode set of 496 opcodes, being

— 16 single-byte opcodes (code table column 2);

— (15 × 32) = 480 double-byte opcodes (first byte from code table column 3 (except 3/15, second byte from either column 2 or column 3).

5.3.2 Extension mechanism

The basic opcode set can be extended by means of the EXTEND OPCODE SPACE (EOS, 3/15) to provide an unlimited number of extension opcode sets.

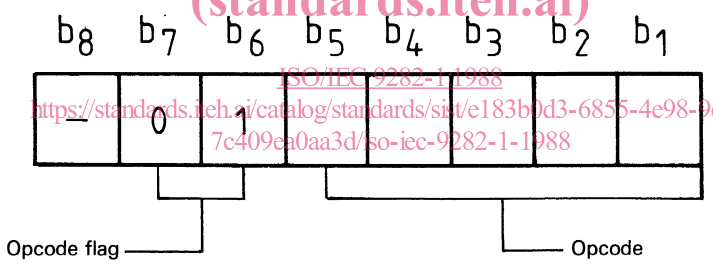

Figure 1 — Compact opcode encoding structure

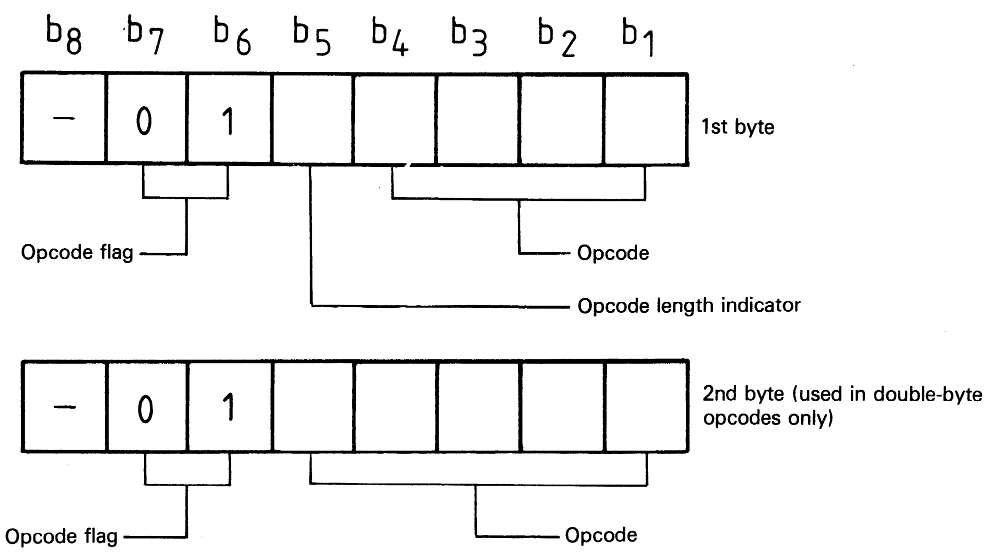

Figure 2 — Opcode encoding structure

The N-th extension opcode set consists of opcodes of the basic opcode set, prefixed N times with the EOS byte. The three possible formats of an opcode from an N-th expansion opcode set are

Opcode format	Extension code	Basic opcode set codes
1	$\underbrace{\langle EOS \rangle \ldots \langle EOS \rangle}_{n \text{ times}}$	$\langle 2/x \rangle$
2	$\underbrace{\langle EOS \rangle \ldots \langle EOS \rangle}_{n \text{ times}}$	$\langle 3/y \rangle$ $\langle 2/z \rangle$
3	$\underbrace{\langle EOS \rangle \ldots \langle EOS \rangle}_{n \text{ times}}$	$\langle 3/y \rangle$ $\langle 3/z \rangle$

EOS = 3/15

x = 0, 1, ..., 15

y = 0, 1, ..., 14

z = 0, 1, ..., 15

n = 0, 1, ..., ...

where

n = 0 results the basic opcode set

n = 1 results the 1st extension opcode set

n = N results the N-th extension opcode set

The number of opcodes supplied by this encoding technique (basic opcode set plus extension opcode sets) is

— 16 single-byte opcodes from the opcode set (opcode format 1, n = 0)

— 480 double-byte opcodes from the basic opcode set (opcode format 2 and 3, n = 0)

16 double-byte opcodes from the 1st extension opcode set (opcode format 1, n = 1)

— 480 N-byte opcodes from extension opcode set $N-2$ (opcode format 2 and 3, n = $N-2$)

16 N-byte opcodes from extension opcode set $N-1$ (opcode format 1, n = $N-1$)

6 Encoding principles for operands

6.1 General

The operand part of a primitive may contain any number of operands, including zero. Each such operand may consist of one or more bytes.

The general format of an operand byte is given in figure 3:

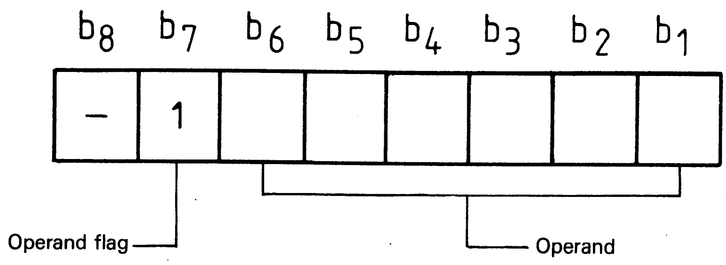

Figure 3 — Operand encoding structure

6.2 Format definitions

Operands may be encoded using three formats:

— basic format;

— bitstream format;

— string format.

In addition the encoding of operands may be controlled by a state variable. Such a state variable is set to a value by an application at the time a coding/decoding process is initialized. Depending on the functional standard under which the encoding principles defined in this part of ISO 9282 are used, such a state variable may remain fixed, or may be dynamically modified by a function specified in the functional standard.

The list of state variables used in this part of ISO 9282 is given in annex A.

This part of ISO 9282 only refers to the value of state variables and does not deal with the definition of functions allowing to modify them.

6.2.1 Basic format

An operand in basic format is represented as a sequence of one or more bytes; the structure is shown in figure 4:

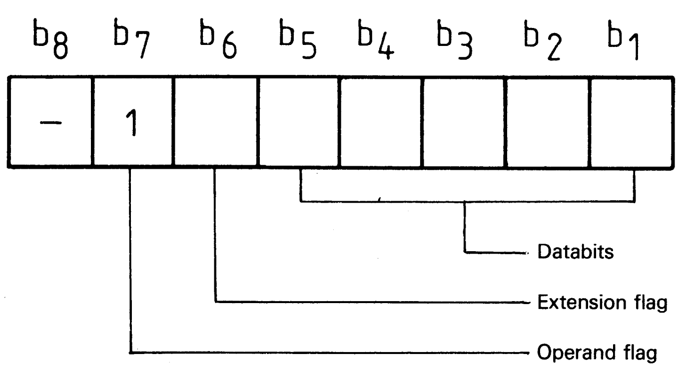

Figure 4 — Basic format structure

For single-byte operands, the extension flag, bit b_6, is ZERO. In multiple-byte operands, the extension flag is ONE in all bytes except the last byte, where it is ZERO.

6.2.2 Bitstream format

An operand in bitstream format is represented as a sequence of one or more bytes; the structure is shown in figure 5:

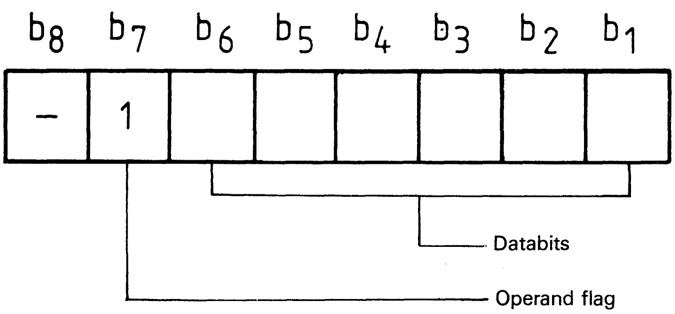

Figure 5 — Bitstream format structure

Data in bitstream format are packed in consecutive databits within an operand byte starting from the higher numbered bits to the lower numbered bits of the first byte for the most significant part of the bitstream data.

The end of a bitstream format operand cannot be derived from the bitstream format itself (the format is not self-delimiting).

The end of a bitstream format operand is delimited by

— the next opcode;

or

— an <end of block> value, which identifies the end of data being encoded in the bitstream format operand;

or

— the value of a state variable which defines the length of the operand which is encoded in the bitstream format.

When the data, which are to be coded in the bitstream format, do not match a whole number of bytes, the remaining bits of the most significant byte shall all be put as ZERO's.

6.2.3 String format

An operand in string format is encoded as a sequence of bytes; the structure is shown in figures 6 and 7.

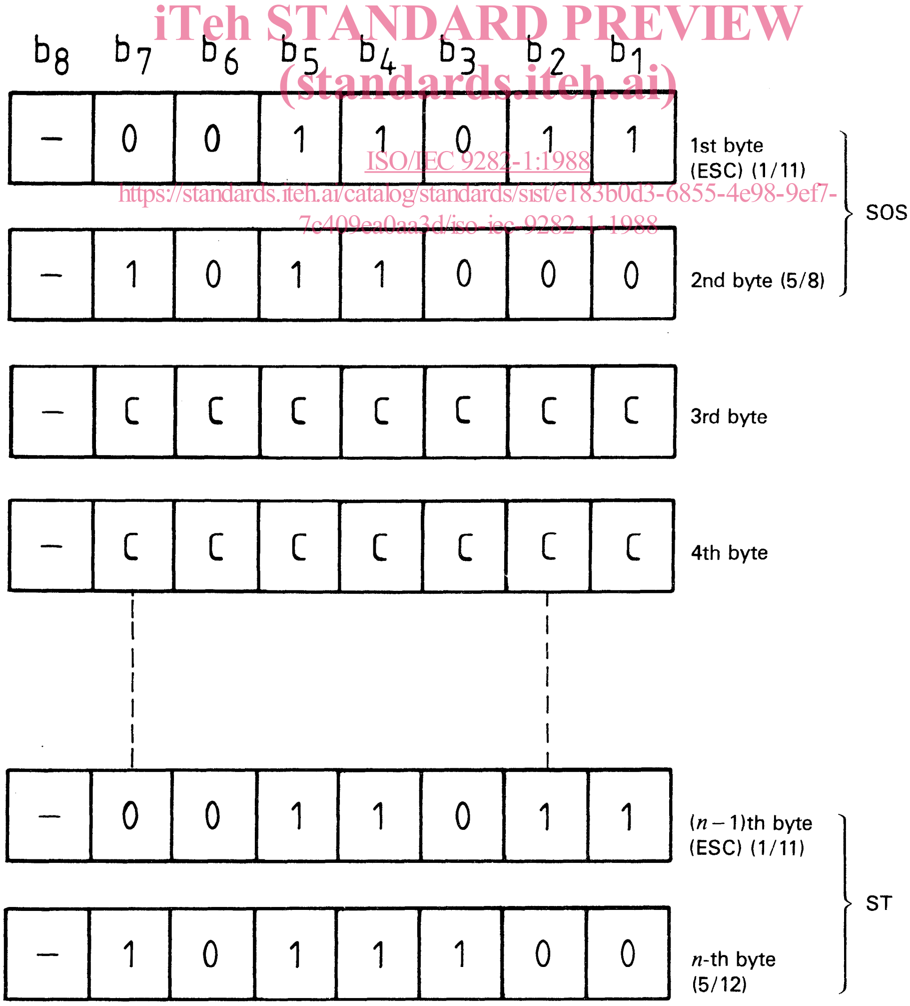

Figure 6 — String format structure in a 7-bit environment
(Databits are marked C)