INTERNATIONAL STANDARD ISO 2022

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

Code extension techniques for use with the ISO 7-bit coded character set

First edition — 1973-07-01

UDC 681.3.042

Ref. No. ISO 2022-1973 (E)

Descriptors : data processing, data transmission, character sets, coding, extensions.

Price based on 23 pages

FOREWORD

ISO (the International Organization for Standardization) is a worldwide federation of national standards institutes (ISO Member Bodies). The work of developing International Standards is carried out through ISO Technical Committees. Every Member Body interested in a subject for which a Technical Committee has been set up 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.

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.

International Standard ISO 2022 was drawn up by Technical Committee ISO/TC 97, *Computers and information processing,* and circulated to the Member Bodies in June 1972.

It has been approved by the Member Bodies of the following countries :

Bulgaria	Italy	Thailand
Czechoslovakia	Japan	United Kingdom
Denmark	Netherlands	U.S.A.
France	Romania	U.S.S.R.
Germany	Sweden	
Ireland	Switzerland	

The Member Bodies of the following countries expressed disapproval of the document on technical grounds :

Australia
Belgium
South Africa, Rep. of

Printed in Switzerland

CONTENTS

Code extension techniques for use with the ISO 7-bit coded character set

1 SCOPE

1.1 This International Standard specifies methods of extending the 7-bit code, remaining in a 7-bit environment or increasing to an 8-bit environment. These techniques are described in four inter-related sections dealing respectively with :

a) the extension of the 7-bit code remaining in a 7-bit environment;

b) the structure of a family of 8-bit codes;

c) the extension of an 8-bit code remaining in an 8-bit environment;

d) the relationship between the 7-bit code and an 8-bit code.

1.2 While the 7-bit code of ISO 646 is the agreed code for information interchange, an 8-bit code as described in this International Standard is provided for information interchange within an 8-bit environment.

1.3 It is not the intention of this International Standard that all instances of its application accommodate all of its provisions. However, it is intended that, when code extension techniques are used, the applicable parts of this International Standard are to be followed.

When two systems with different levels of implementation of code extension techniques are required to communicate with one another they will do so using the code extension techniques they have in common.

1.4 Code extension techniques are classified and some classes are given a structure in this International Standard. Other assignments of bit combinations associated with the designation of the classes are made in accordance with ISO 2375. Specific assignments of bit combinations to relate individual codes with their invocation or designation are also to be made in accordance with that International Standard.

1.5 Code extension techniques are designed to be used for data to be processed serially in a forward direction. Use of these techniques in strings of data which are processed other than serially in a forward direction or included in data formatted for fixed record processing may have undesirable results or may require additional special treatment to ensure correct interpretation.

2 FIELD OF APPLICATION

The 7-bit code of ISO 646 allows, through its different versions, the representation of up to 128 characters. Additionally, that document allows the representation of other graphics by the combination of two graphic characters with the back space control. In some instances the code of ISO 646 lacks sufficient controls or graphics to satisfy the needs of an application.

These needs may be satisfied by means of code extension which is the subject of this International Standard.

This International Standard presents a review of the salient structure of the 7-bit code and then builds upon that structure to describe various means of extending the control and graphic sets of the code. It also describes structures and techniques to construct and formalize codes related to the 7-bit code. These related codes are structured so as to allow application dependent usage without preventing the interchangeability of data employing them. This document describes :

a) the structure of the 7-bit code;

b) extension of the 7-bit code, remaining in a 7-bit environment and making use of code extension techniques;

c) increasing the number of bits to 8, yet retaining a structure compatible with the 7-bit structure;

d) increasing the number of bits to 8 and applying similar code extension techniques.

In order to use identical techniques in each of the above cases, and to facilitate conversion between them, standard rules are necessary. This has the advantage of :

a) reducing the risk of conflict between systems required to inter-operate;

b) permitting provision for code extension in the design of systems;

c) providing standardized methods of calling into use agreed sets of characters;

d) allowing the interchange of data between 7-bit and 8-bit environments, etc.

This International Standard also describes the structure of families of codes which are related to the code of ISO 646 by their structure.

3 REFERENCES

ISO 646, *7-bit coded character set for information processing interchange.*

ISO 2375, *Data processing — Procedure for registration of escape sequences.* [1]

4 DEFINITIONS AND NOTATION

4.1 Definitions

For the purpose of this International Standard, the following definitions apply :

4.1.1 character : A member of a set of elements that is used for the organisation, control or representation of data.

4.1.2 code; coded character set : A set of unambiguous rules that establish a character set and the one-to-one relationship between the characters of the set and their bit combinations.

4.1.3 bit combination : An ordered set of bits that represents a character.

4.1.4 code table : A table showing the character corresponding to each bit combination in a code.

4.1.5 position : An item in a code table identified by its column and row coordinates.

4.1.6 byte : A bit string that is operated upon as a unit and whose size is independent of redundancy or framing techniques.

4.1.7 control function : An action that affects the recording, processing, transmission or interpretation of data.

4.1.8 control character : A character whose occurrence in a particular context initiates, modifies or stops a control function.

4.1.9 graphic character : A character, other than a control character, that has a visual representation normally handwritten, printed or displayed.

4.1.10 code extension : Techniques for the encoding of characters that are not included in the character set of a given code.

4.1.11 escape sequence : A bit string that is used for control purposes in code extension procedures and that consists of two or more bit combinations. The first of these combinations corresponds to the escape character.

4.1.12 final character : The character whose bit combination terminates an escape sequence.

4.1.13 intermediate character : A character whose bit combination occurs between the escape character and the final character in an escape sequence consisting of more than two bit combinations.

4.1.14 to designate : To identify a set of characters that are to be represented, in some cases immediately and in others on the occurrence of a further control function, in a prescribed manner.

1) At present at the stage of draft.

4.1.15 to invoke : To cause a designated set of characters to be represented by the prescribed bit combinations whenever those bit combinations occur, until an appropriate code extension function occurs.

4.1.16 to represent :

1) To use a prescribed bit combination with the meaning of a character in a set of characters that has been designated and invoked.

2) To use an escape sequence with the meaning of an additional control character.

4.1.17 environment : The characteristic that identifies the number of bits used to represent a character in a data processing or data communication system or in part of such a system.

4.1.18 national version : A code of 128 characters that is identical to the 7-bit coded character set of ISO 646 except in those positions in which ISO 646 makes provision for the assignment of alternative graphics and in those positions conforms to the requirements of ISO 646.

4.2 Notation

In this International Standard the following notations are used :

For the bits of a 7-bit combination :	b_7	b_6	b_5	b_4	b_3	b_2	b_1
For the bits of an 8-bit combination :	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1
Bit weight for column and row reference :	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
	column				row			

A bit combination is sometimes referred to by the column and row numbers of its position in the code table. The column number is the decimal equivalent of bits $b_7 - b_5$ (or $a_8 - a_5$) and the row number is the decimal equivalent of bits $b_4 - b_1$ (or $a_4 - a_1$), giving to these bits the weights shown above.

In representing the decimal equivalents, the convention is to append a leading zero to the column number for 8-bit columns 00 to 09. As an example the position of the "space" character in the 7-bit code table is 2/0; the position of the same character in an 8-bit code table is 02/0.

Character mnemonics such as SO, ESC and column/row numbers such as 0/5 and 1/7 are shown underlined to emphasize the fact that they stand for one bit-combination only.

5 EXTENSION OF THE 7-BIT CODE REMAINING IN A 7-BIT ENVIRONMENT

5.1 Introduction

5.1.1 *The structure of the 7-bit code*

The 7-bit code table which is the basis of code extension techniques for use with the 7-bit coded character set of ISO 646 consists of areas for an ordered set of control characters and graphic characters grouped as follows :

1) the area for a set of 32 control characters allocated to columns 0 and 1;

2) the space character in position 2/0 which may be regarded either as a control character or a non-printing graphic character;

3) the area for a set of 94 graphic characters allocated to columns 2 to 7;

4) the delete character in position 7/15.

This is shown in Figure 1.

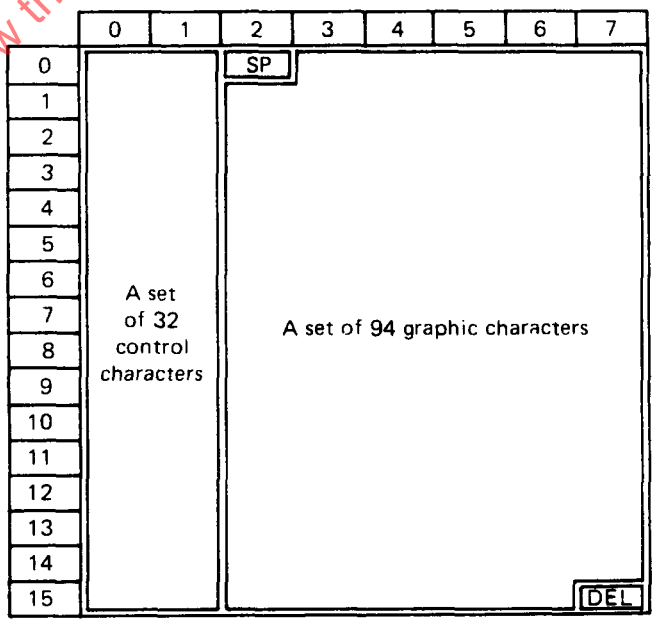


FIGURE 1 – **The structure of the 7-bit code**

5.1.2 *Extension by substitution*

In many cases the provisions of ISO 646 will satisfy the requirements of an application. Other applications will be satisfied by the use of a similarly structured code in which some of the characters of ISO 646 are substituted by other characters. Such substitution may be regarded as a replacement of the control set and/or the graphic set according as new controls and/or graphic characters are required.

5.1.3 *Extension by increasing the repertoire of characters*

This International Standard provides for additional characters to the 128 provided by the structure of the 7-bit code in the following ways :

1) additional single controls;

2) additional sets of 32 control characters;

3) additional sets of 94 graphic characters;

4) additional sets of more than 94 graphic characters each represented by more than one byte.

5.1.4 *The elements of code extension*

Many applications will require combinations of the above code extension facilities. The elements of code extension are shown in Figure 2, where the names of elements are defined as follows :

– C0 set : a set of 32 control characters (columns 0 and 1);

– C1 set : an additional set of 32 control characters;

– G0 set : a set of 94 graphic characters (columns 2 to 7) (a multiple byte set also functions as the G0 set);

– G1 set : an additional set of 94 graphic characters.

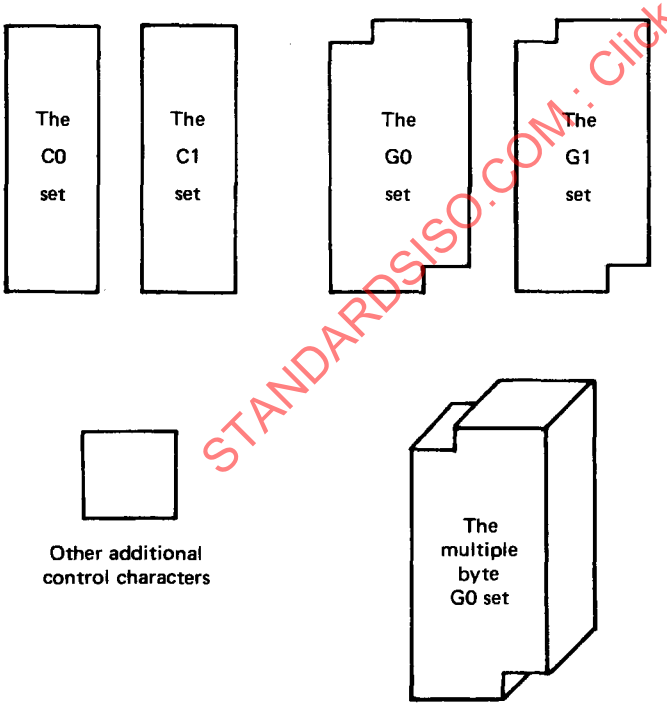


FIGURE 2 – **The elements of code extension**

NOTE – It is intended that a set of control characters and a set of graphic characters which are permitted by ISO 646, if they are used, are assigned to the C0 set and the G0 set respectively.

5.1.5 *Compatibility*

For purpose of interchange there are identified various levels of compatibility which may be preserved when applying extension facilities. The following three such levels are distinguished in this International Standard :

1) a version permitted by ISO 646.

2) a compatible variant.

A compatible variant is defined as a set which is compatible with the ISO 646 inasmuch as

– columns 0 and 1 contain only control characters;

– columns 2 to 7 are used for graphic characters only (apart from DEL);

– the ten transmission control characters and NUL, SO, SI, CAN, SUB, ESC, SP and DEL remain unaltered in their meanings and in their positions in the code table;

– graphics of ISO 646 are not moved to other positions (a non-latin alphabet containing graphics which are also included in the latin alphabet is not subject to this rule).

3) other sets structured as in 5.1.1 above. To be able to provide the facilities of code extension of this International Standard, the escape and/or the shift-out and shift-in characters must remain unaltered in their meanings and their positions in the code table.

5.1.6 *Code extension characters*

In the 7-bit code, the following characters are provided for the purpose of code extension :

– the escape character	ESC
– the shift-out character	SO
– the shift-in character	SI
– the data link escape character	DLE

This International Standard does not describe the use of the data link escape character which is reserved for the provision of additional transmission controls; the use of this character is specified in other ISO publications.

5.2 Extension of the graphic set by means of the shift-out and shift-in characters

5.2.1 *Use of shift-out and shift-in*

The shift-out character SO and the shift-in character SI are used exclusively for extension of the graphics.

The character SO invokes an additional set of 94 graphics : the G1 set. This set replaces the graphic characters of the G0 set. Graphic characters need not be assigned to all the positions of the additional set, nor, except as specified below, need all the graphic characters of the additional set be different from the graphic characters of the G0 set.

The character SI invokes the graphic characters of the G0 set that are to replace the graphic characters of the additional set.

The meanings of the following bit combinations are not affected by the occurrence of SO and SI :

1) those corresponding to the control characters in columns 0 and 1 and position 7/15;

2) the one corresponding to the space character in position 2/0;

3) those included in any escape sequence.

The space character occurs only at position 2/0; it shall not be assigned to any position in the alternative graphic set. These provisions do not preclude the assignment to positions in any graphic set of characters equivalent to spaces of size other than that of the space assigned to position 2/0.

At the beginning of any information interchange the shift status shall be defined by SI or SO. When in the shift-in status SI has no effect, and when in the shift-out status SO has no effect.

5.2.2 *Unique shift-out set*

Some applications require the use of only one additional set of 94 graphic characters. In such a case, that unique set is invoked by each use of SO. The set is identified either by an appropriate ESC sequence as described in 5.3.7 or by agreement between the interchanging parties.

5.2.3 *Multiple shift-out sets*

If two or more additional graphic sets are required to coexist in a system, the set to be used next is designated by the appropriate ESC sequence. That set is then invoked by the use of SO.

The use of SI re-invokes the graphics of the G0 set last designated but does not affect the identity of the designated G1 set. An additional set may be invoked any number of times by successive use of SO until it is superseded by another G1 set designated by another escape sequence.

It is not necessary to revert to the G0 set by use of SI before changing from one G1 set to another by means of a further escape sequence. When the system is in the shift-out state, the use of such a further escape sequence leaves the shift status unaltered, and the additional set is invoked.

Figure 3 is a schematic representation of the above.

In some devices or systems there may be a requirement to re-establish the shift-in state before designating a new shift-out set by means of an escape sequence. This can be achieved by inserting SI before the escape sequence which designates the subsequent shift-out set. The use of such a procedure is subject to agreement between the interchanging parties.

5.3 Code extension by means of escape sequences

5.3.1 *Purposes of escape sequences*

Escape sequences provide single or sets of control functions other than for transmission control. Escape sequences are also used to designate sets of graphics, different uses of some or all of the 7-bit code combinations, and coded character sets with a number of bits other than 7.

Thus escape sequences are required to provide, for example :

- a single control character not already in the code;
- a set of control characters not already in the code;
- a set of graphic characters not already in the code;
- a code structure different from the code.

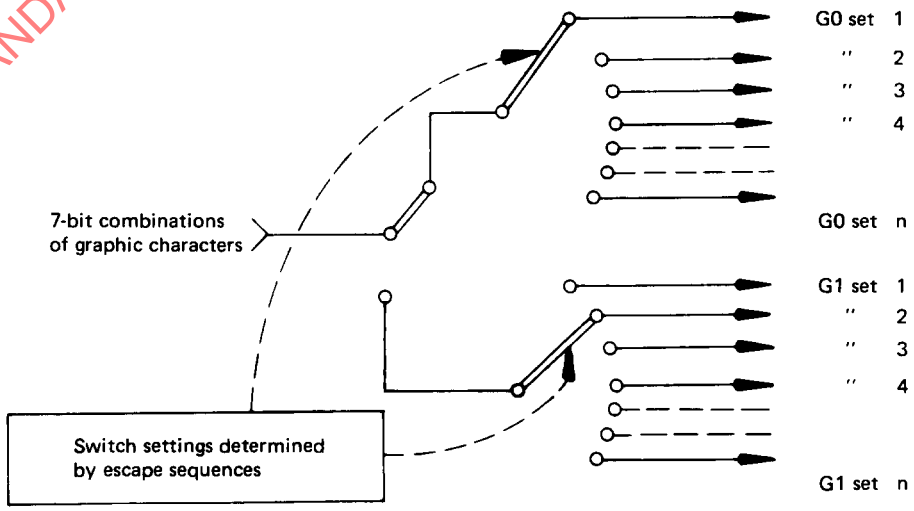


FIGURE 3

5.3.2 *Structure of escape sequences*

An escape sequence consists of two or more 7-bit combinations. The first is always the bit combination of ESC and the last is always one of the Final characters. An escape sequence may also contain any number of 7-bit combinations representing Intermediate characters.

The meaning of an escape sequence is determined by the 7-bit combination representing its Intermediate character(s), if any, and by the 7-bit combination representing its Final character.

WARNING : Although in this International Standard, escape sequences are described in terms of characters or of positions of the code table, the meaning of an escape sequence is determined only by its bit combinations and it is unaffected by any meaning previously assigned to these bit combinations taken individually.

Intermediate characters are the 16 characters of column 2 of the 7-bit code table.

NOTE — In this International Standard, any one of these 16 intermediate characters is denoted by the symbol : (I).

Final characters are the 79 characters of columns 3 to 7 of the 7-bit code table excluding position 7/15.

NOTE — In this International Standard, any one of these 79 final characters is denoted by the symbol : (F).

Prohibited characters are the control characters in columns 0 and 1 and the character in position 7/15.

The 33 prohibited characters shall not be used as either intermediate or final characters to construct an escape sequence.

As these prohibited characters may appear in an ESC sequence in error, it may be necessary within an application to provide methods of identifying such a situation and of recovering from it but this is not covered by this International Standard.

5.3.3 *Categories of escape sequence*

The use of escape sequences is specified in this International Standard. However, escape sequences with final characters from column 3 are reserved for private use subject to the categorization outlined below.

WARNING : The implementors of any private escape sequence described as such in this document are alerted to the fact that other implementors may give different meanings to the same escape sequence or may use different escape sequences to mean the same thing. Furthermore, such meanings may subsequently be assigned to standardized escape sequences. Interchanging parties are warned that the use of such private escape sequences may reduce their capability to interchange data subsequently.

5.3.3.1 *Two-character escape sequences*

A two-character escape sequence takes the form :

ESC (F)

Such escape sequences are used to represent single additional control characters.

The 79 two-character sequences are split into three types, depending on the Final character, as shown in Figure 4.

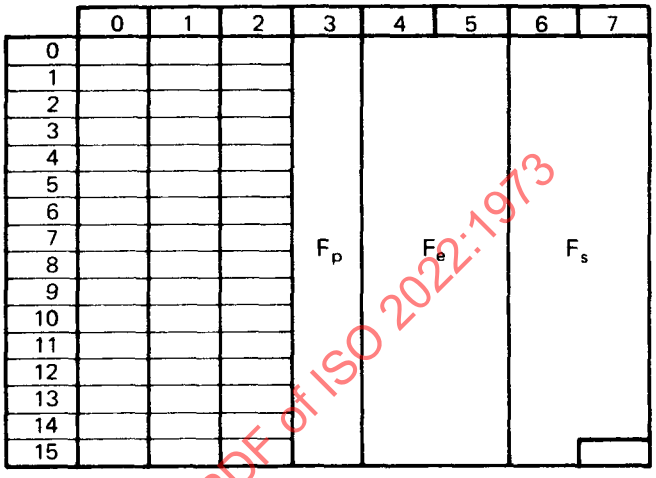


FIGURE 4

An ESC (F_s) sequence represents, depending on the Final character used, a single additional standardized control character. 31 Final characters of columns 6 and 7 are provided for this purpose.

An ESC (F_e) sequence represents, depending on the Final character used, an individual control character of an additional standardized set of 32 control characters (see 5.3.6). The 32 Final characters of columns 4 and 5 are provided for this purpose. Some applications require the use of only one such additional set. In this case, the set is identified either by the appropriate ESC sequence, as described in 5.3.6, or by agreement between the interchanging parties. If more than one additional set of controls are required to coexist in a system, the set to be used next is designated and invoked by the appropriate ESC sequence.

An ESC (F_p) sequence represents, depending on the Final character used, a single additional control character without standardized meaning for private use as required, subject to the prior agreement of the sender and the recipient of the data.

The 16 Final characters of column 3 are provided for this purpose.

5.3.3.2 *Three-character escape sequences*

A three-character escape sequence takes the form :

ESC (I) (F)

All types of three-character escape sequences are grouped into classes, according to their purpose, by means of their Intermediate characters, as shown in 5.3.4 to 5.3.11 (see Table 1, page 10).

These sequences are split into two types according to their Final character as shown in Figure 5.

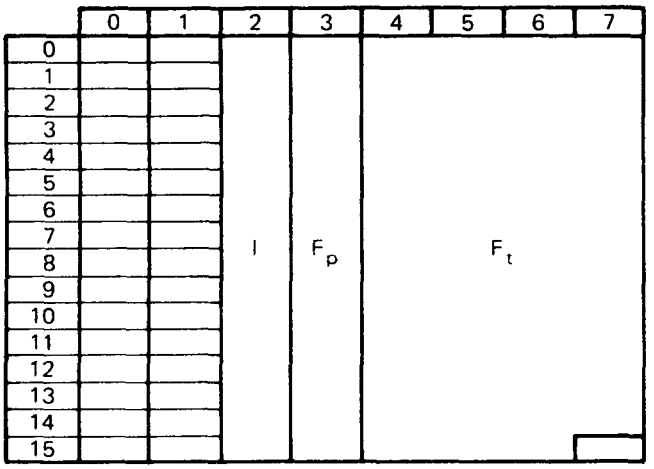


FIGURE 5

ESC (I) (F_t) sequences are used for standardized purposes. 63 F_t characters of columns 4 to 7 are provided for this purpose.

ESC (I) (F_p) sequences are reserved for private use. 16 F_p characters of column 3 are provided for this purpose.

5.3.4 *Single additional control characters*

ESC 2/3 (F) represents a single additional control character depending on the final character used.

5.3.5 *Sets of 32 control characters for columns 0 and 1*

ESC 2/1 (F) designates and invokes the C0 set of 32 control characters for representation by the bit combinations of columns 0 and 1.

The ten transmission control characters, when included in a C0 set, shall retain their meanings and their positions in the code table. No other transmission control characters may be included in a C0 set.

To reduce the risk of conflict in the interchange of data, this set should have the following characteristics :

– inclusion of the ten transmission control characters;

– inclusion of the characters NUL, SO, SI, CAN, SUB, and ESC with their meanings and their positions in the 7-bit code table unaltered.

Consideration should be given to the effect that changing the meaning of control characters can have on equipment when interchanging data. For example the bit combination corresponding to HT will have the effect of "horizontal tabulation" to a system designed to respond to this control character.

5.3.6 *Sets of 32 control characters for representation by ESC F_e*

ESC 2/2 (F) designates and invokes the C1 set of 32 control characters without affecting the C0 set.

Individual control characters of such a set are represented by means of ESC (F_e) sequences instead of a single bit-combination. A C1 set shall not include transmission control characters.

5.3.7 *Sets of 94 graphic characters*

ESC 2/8 (F) and ESC 2/12 (F) designate sets of 94 graphic characters which will be used as the G0 set. The designated set is invoked by SI.

ESC 2/9 (F) and ESC 2/13 (F) designate sets of 94 graphic characters which will be used as the G1 set. The designated set is invoked by SO.

NOTE – Two groups of graphic sets are mentioned above. The groups together make up a single repertory of graphic sets which may be designated either as a G0 or as a G1 set. No significance is attached to the groupings other than that their existence allows more such graphic sets to be defined within the scope of three character escape sequences as defined in this International Standard. There are therefore 126 (2 × 63) such sets possible without requiring further extension.

5.3.8 *Codes which require special interpretation*

ESC 2/5 (F) designates and invokes a code that requires special interpretation, such as :

– a code with a number of bits other than 7, excluding those 8-bit codes structured in accordance with this International Standard.

– a 7-bit code whose characteristics differ from those in this International Standard.

The final character assignments are such that within the F_t and F_p groups the following classification occurs :

Final in column	Broad categorization
3	a private code with any number of bits
4	a code of less than 7 bits
5	a code of 7 bits
6	a code of 8 bits
7	a code of more than 8 bits

5.3.9 *Sets of graphics with multiple byte representation*

ESC 2/4 (F) designates sets of graphic characters that are represented by two or more bytes each corresponding to a bit combination in columns 2 to 7, apart from positions 2/0 and 7/15. The designated set is invoked by SI and is therefore regarded as a G0 set. Within such a set, each

graphic character is represented by the same number of bytes, as shown in Figure 6 below :

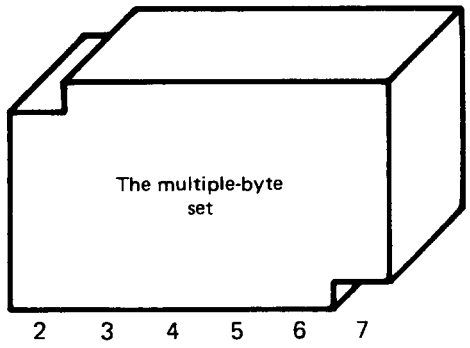


FIGURE 6

5.3.10 *Announcement of extension facilities*

ESC 2/0 (F) announces the extension facilities used in conjunction with data which follow. The use of these sequences is specified in section 8.

5.3.11 *Three-character escape sequences without assigned meanings*

The escape sequences ESC 2/6 (F), ESC 2/7 (F), ESC 2/10 (F), ESC 2/11 (F), ESC 2/14 (F), and ESC 2/15 (F) have not been assigned meanings and are reserved for further standardization.

5.3.12 *Escape sequences having four or more characters*

Escape sequences having four or more characters will be interpreted according to the following :

1) The first intermediate character will indicate the class of usage identical with three character escape sequences above.

2) The second and any additional intermediate characters will be associated with the final character to permit additional entities within the class defined by the intermediate character.

3) All escape sequences having four or more characters whose final character is of the (F_t) type are reserved for further standardization.

4) All escape sequences whose final character is of the (F_p) type (private) are not to be the subject of further standardization.

(See warning in 5.3.3 and the grouping assigned in 5.3.3.2.)

5.4 Omission of escape sequences

If the interchanging parties have agreed upon a single G0 set, a single G1 set, a single C0 set and a single C1 set (or on as many of these sets as are to be used), they may also agree to omit the use of escape sequences to designate or invoke them. Interchanging parties are warned however that such agreements may reduce their capability to interchange data subsequently.

5.5 Pictorial and tabular representations

Figure 7 summarises, in a schematic form, the standard means of code extension within a 7-bit environment.

Table 1 summarises the assignment of intermediate characters in escape sequences.

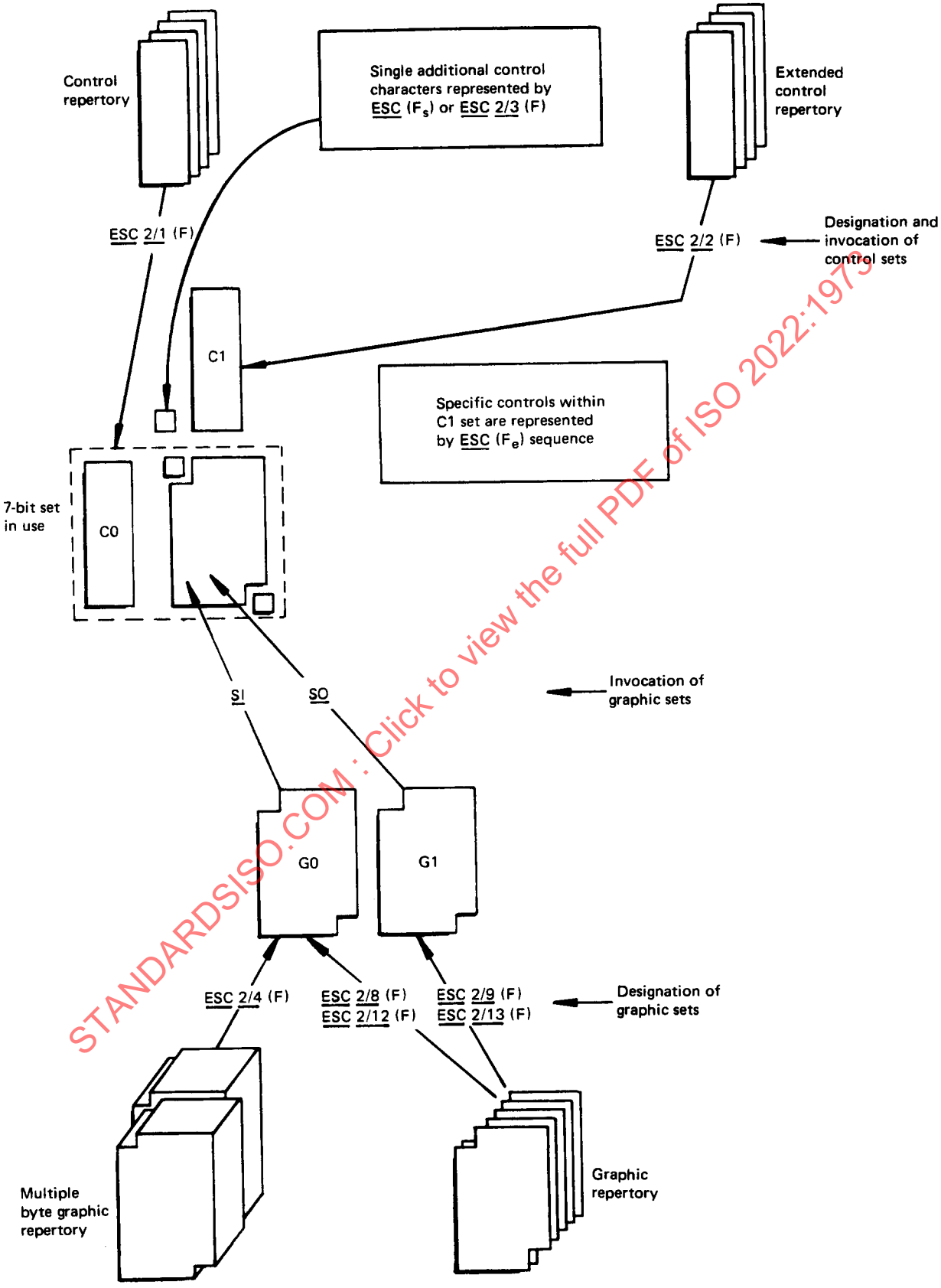


FIGURE 7 – **Code extension in a 7-bit environment**

TABLE 1 — **Summary of assignments of three character escape sequences**

Column/Row	Bits of intermediate characters $b_7 b_6 \ldots b_1$	Category	Grouping	See clause
2/0	0 1 0 0 0 0 0	Announcers		5.3.10
2/1	0 1 0 0 0 0 1	CONTROLS	C0 set	5.3.5
2/2	0 1 0 0 0 1 0		C1 set	5.3.6
2/3	0 1 0 0 0 1 1		Single controls	5.3.4
2/4	0 1 0 0 1 0 0	GRAPHICS	Multiple byte sets	5.3.9
2/5	0 1 0 0 1 0 1	Codes requiring special interpretation		5.3.8
2/6	0 1 0 0 1 1 0	Reserved for future standardization		5.3.11
2/7	0 1 0 0 1 1 1			
2/8	0 1 0 1 0 0 0	GRAPHICS[1)]	G0 set	5.3.7
2/9	0 1 0 1 0 0 1		G1 set	
2/10	0 1 0 1 0 1 0		Reserved for future standardization	5.3.11
2/11	0 1 0 1 0 1 1			
2/12	0 1 0 1 1 0 0		G0 set	5.3.7
2/13	0 1 0 1 1 0 1		G1 set	
2/14	0 1 0 1 1 1 0		Reserved for future standardization	5.3.11
2/15	0 1 0 1 1 1 1			

1) There is a single repertory of sets of 94 graphic characters. Any member of the repertory may be designated as either a G0 set or G1 set. Four designating escape sequences, two for G0 and two for G1 are provided for designating members of the repertory.

6 STRUCTURE OF A FAMILY OF 8-BIT CODES

The family of 8-bit codes specified in this International Standard is obtained by the addition of one bit to each of the bit combinations of the 7-bit code, thus producing a set of 256 8-bit combinations. The characters of the 7-bit set are assigned to the 128 bit combinations whose eighth bit is zero. In this way, the set as defined in 5.1 forms a defined and integral part of an 8-bit code that is structured in accordance with this International Standard. The 128 additional bit combinations, whose eighth bit is "one" are available for further assignment.

6.1 The 8-bit code table

A 16 by 16 array of columns numbered 00 to 15 and rows numbered 0 to 15 contains 256 code positions (see Figure 8).

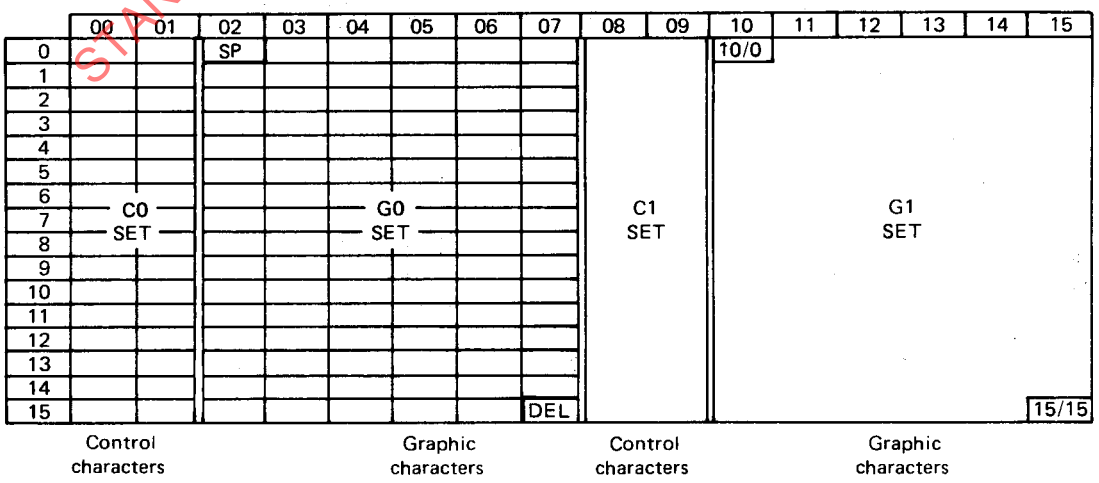


FIGURE 8

Columns 00 to 07 of this array contain 128 character positions which are in one-to-one correspondence with the characters of the 7-bit set. Their coded representation is the same as in the 7-bit environment with the addition of an eighth, most significant bit, which is ZERO.

Columns 08 to 15 of this array contain a further 128 code positions; the eighth bit of their coded representations is ONE.

Columns 08 and 09 are provided for control characters and columns 10 to 15 for graphic characters, subject to the exception of positions 10/0 and 15/15 described below.

The control characters in columns 08 and 09 of an 8-bit code shall not include transmission control characters. Provision of data transmission capability for 8-bit codes includes the use of the data link escape characters and is the subject of other ISO publications.

6.2 The family concept

In order to cope with the different needs of the various industries, fields of application or systems, this International Standard defines the concept of a family of 8-bit codes as follows :

1) a set of 32 additional control characters can be selected for columns 08 and 09;

2) a set of 94 additional graphic characters can be selected for columns 10 to 15 (excluding positions 10/0 and 15/15).

There are standard techniques for identifying selections of sets of controls and graphics for 8-bit codes. These techniques are described below.

7 THE USE OF CODE EXTENSION IN AN 8-BIT CODE

The techniques of extending an 8-bit code described in this International Standard have been purposely made compatible with those used to extend the 7-bit code.

The escape character is used in an 8-bit code in exactly the same way as in the 7-bit code to construct ESC sequences. The meanings of these sequences are not altered in an 8-bit code. All characters in columns 08 -15 are excluded from assignment in escape sequences and any occurrences of them in an escape sequence are error conditions for which no standard recovery procedures are prescribed in this International Standard.

7.1 Definition of an 8-bit code

As described in section 6, the code table can be considered as having four main parts :

- the C0 control set;
- the G0 graphic set;
- the C1 control set;
- the G1 graphic set;

The remainder of the code table consists of positions 02/0 (SP), 07/15 (DEL), 10/0 and 15/15.

The C0 and the G0 sets are designated, and invoked by the same escape sequences as in the 7-bit environment (see 5.3.5 and 5.3.7).

The C1 set of control characters is designated and invoked by means of an escape sequence as in the 7-bit environment (see 5.3.6). These control characters are represented by the bit combinations of columns 08 and 09.

The G1 set of graphic characters is designated and invoked by means of an escape sequence as in the 7-bit environment (see 5.3.7). These graphic characters are represented by the bit combination of columns 10 to 15.

7.2 Code extension by means of escape sequences

Once the 8-bit code is established in accordance with 7.1, code extension means are available, making use of escape sequences as described therein and below.

7.2.1 *Two-character escape sequences*

Two-character escape sequences have the same structure as in the 7-bit environment (see 5.3.2).

ESC (F_s) sequences represent single additional controls with the same meaning they have in the 7-bit environment.

The use of ESC (F_e) sequence in an 8-bit environment is contrary to the intention of this International Standard but should they occur their meaning is the same as in the 7-bit environment.

7.2.2 *Three-character escape sequences*

Three-character escape sequences have the same structure and meaning as in the 7-bit environment (see 5.3).

7.2.3 *Escape sequences with four or more characters*

These escape sequences have the same structure and meaning as in the 7-bit environment (see 5.3).

7.3 Pictorial representation

Figure 9 summarises in a schematic form a standard means of extension available in an 8-bit environment.

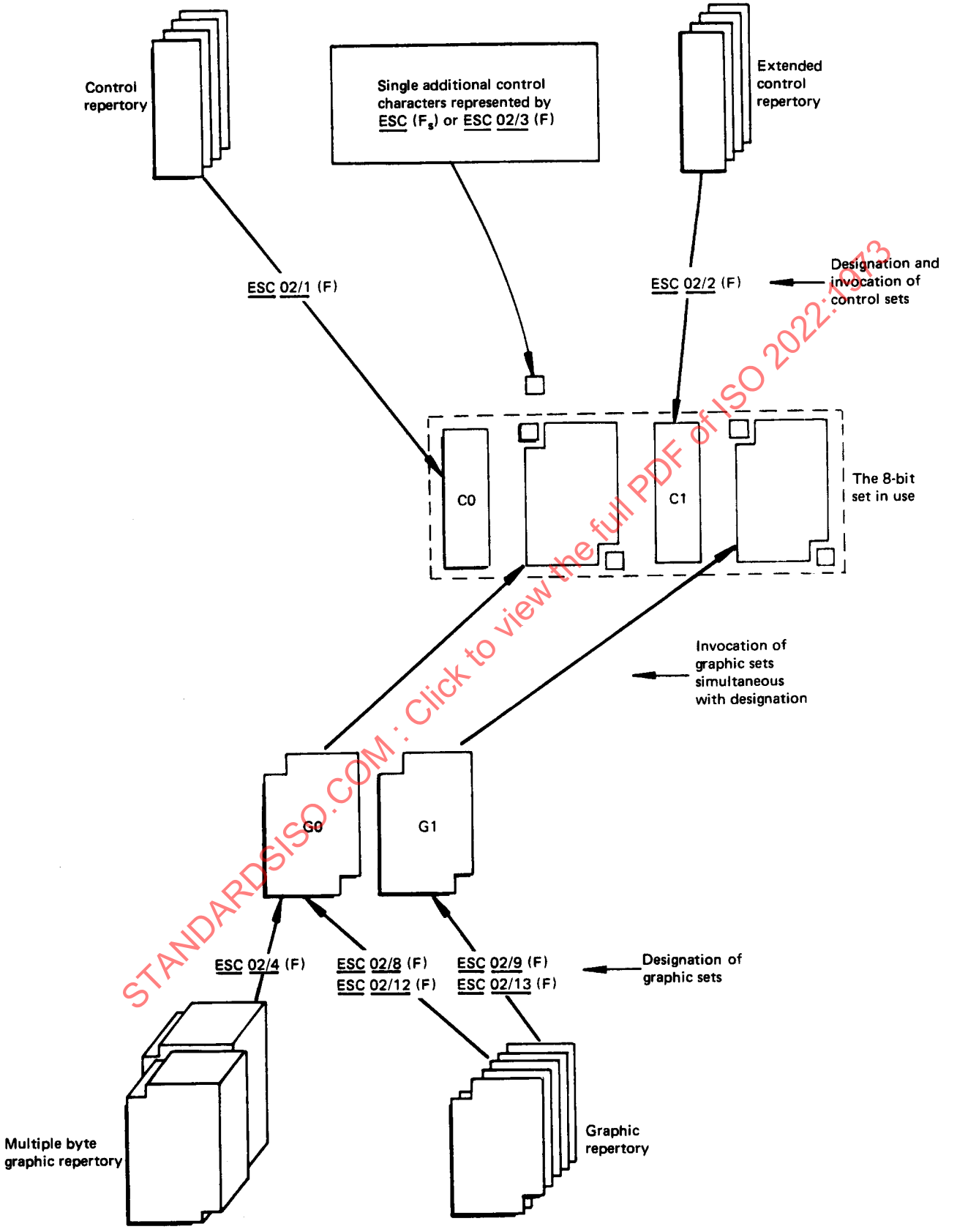


FIGURE 9 – **Code extension in a 8-bit environment**

8 ANNOUNCEMENT OF EXTENSION FACILITIES USED

The class of three character escape sequences ESC 2/0 (F) is used in data interchange to announce the code extension facilities utilized in the data which follow. Subject to agreement between the interchanging parties, such an announcing sequence may be omitted. The final character of the announcing sequence indicates the facilities used for representing graphic sets in 7 and 8-bit environments, and the number of bits used as follows :

Final characters	Facilities utilized
4/1	The G0 set only shall be used. The escape sequence which designates this set also invokes it into columns 2 to 7. SI and SO shall not be used. In an 8-bit environment, columns 10 to 15 are not used.
4/2	The G0 and G1 sets shall be used. In both 7 and 8-bit environments SI invokes G0 into columns 2 to 7 and SO invokes G1 into columns 2 to 7. In an 8-bit environment columns 10 to 15 are not used.
4/3	The G0 and G1 sets shall be used in an 8-bit environment only. The designating escape sequences also invoke the G0 and G1 sets into columns 2 to 7 and columns 10 to 15 respectively. SI and SO shall not be used.
4/4	The G0 and G1 sets shall be used. In a 7-bit environment, SI invokes G0 into columns 2 to 7 and SO invokes G1 into columns 2 to 7. In an 8-bit environment, the designating escape sequences also invoke the G0 and G1 sets into columns 2 to 7 and 10 to 15 respectively; SI and SO shall not be used.

(A pictorial representation of these cases is shown in Figure 10.)

NOTE – In a 7-bit environment, data announced by a sequence ESC 2/0 4/4 have the same form as data announced by a sequence ESC 2/0 4/2.

Both are provided for those interchange situations in which it is agreed to differentiate between 7-bit and 8-bit originated data, in the 7-bit environment.

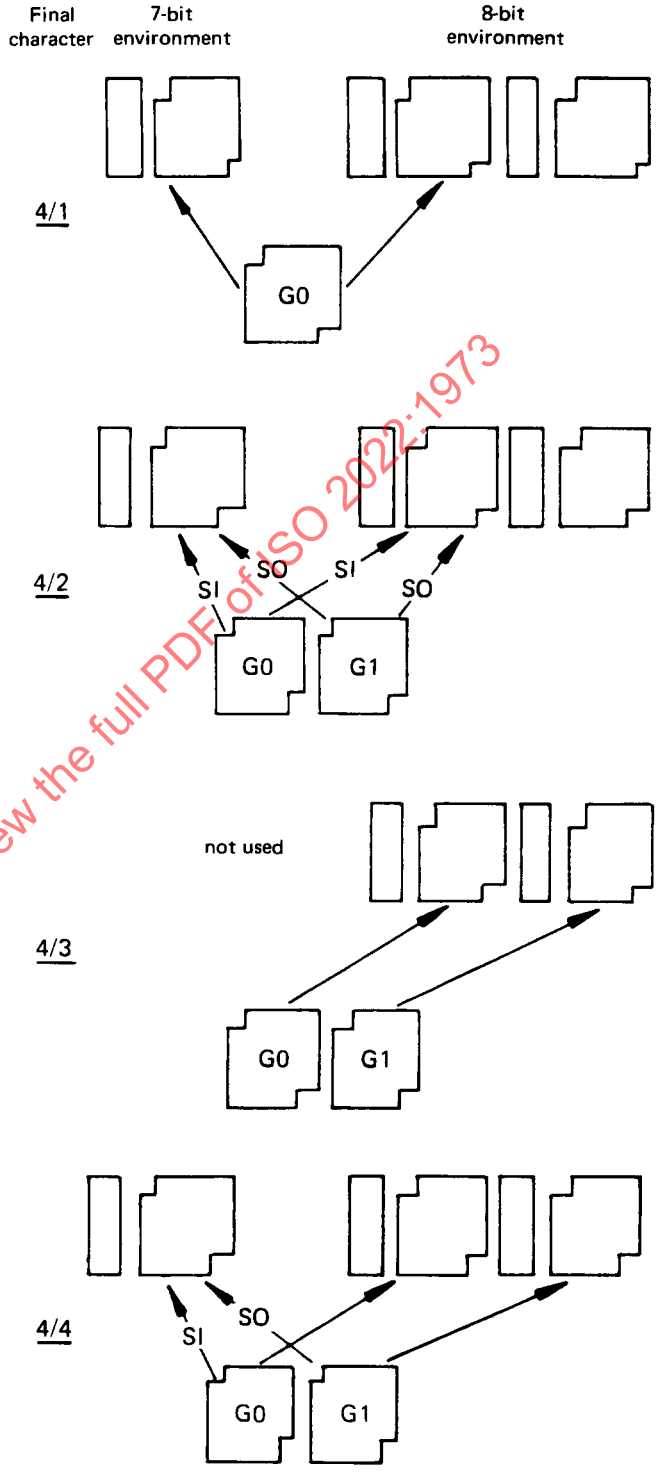


FIGURE 10

9 RELATIONSHIP BETWEEN 7-BIT AND 8-BIT CODES

9.1 Transformation between 7-bit and 8-bit codes

Transformation between 7-bit and 8-bit codes depends on which facilities of code extension are included in the application. Examples of such transformations are included in the Appendix in flowchart form.

9.2 Representation of the 7-bit code in an 8-bit environment

It may sometimes be desirable, as for example, in a store and forward application, to retain information in 7-bit form while in an 8-bit environment. In this case, for each of the characters, bits b_7 to b_1 are represented by bits a_7 to a_1 respectively and a_8 is set to zero.

Indication that true 8-bit coded data follow is achieved by the occurrence of the announcing sequences ESC 2/0 4/3, or ESC 2/0 4/4.

Indication that 7-bit coded data follow is achieved by the occurrence of one of the announcing sequences ESC 2/0 4/1, ESC 2/0 4/2.

9.3 Representation of positions 10/0 and 15/15 in a 7-bit environment

No meaning is assigned to positions 10/0 and 15/15 in this International Standard. If there is a requirement to represent these positions in a 7-bit environment, a private escape sequence shall be used.

10 SPECIFIC MEANING OF ESCAPE SEQUENCES

10.1 The meanings of individual escape sequences are not specified in this International Standard. Instead, their meanings will be specified using the procedures established by ISO 2375. That International Standard is to be followed in preparing and maintaining a register of escape sequences and their meanings. These registration procedures do not apply to escape sequences reserved for private use.

10.2 Furthermore, when required, the classes of 3-character escape sequences which are not defined in this International Standard will be allocated by the ISO Coding Committee (ISO/TC 97/SC 2) using the same procedures mentioned in 10.1.

APPENDIX

The implementation of the extension of the 7-bit code, described in this International Standard, depends on the practical need of the implementor or user of the code extension. Remaining in accordance with this International Standard, they may differ in

- including all or parts of the control sets and graphic sets;
- identifying the sets by ESC-sequences or having an agreement;
- dropping or not the superfluous invokers;
- examining or not the structure of an ESC-sequence;
- handling the character DEL, etc.

In this Appendix there are shown a few of the possible implementations which may help the user to understand this International Standard. A summary of the figures shown in this Appendix is given in Table 2. There are three types of figures : Structure, Interpretation and Transformation.

Table 3 shows the symbols used in the flow charts and the associated meaning. Table 4 shows the items used in the flow charts, their meaning and the possible values.

Z.1 STRUCTURE

The pictorial representations of the structure of the 7-bit and 8-bit code extension have all the same format as follows :

- at the top of the diagram are the repertory of single controls and sets of controls;
- below that are the escape sequences that are used for the designation and invocation of the control sets as the C0 and C1 set;
- at the bottom of the diagram is the repertory of graphic sets;
- above that are the escape sequences used for designation of the graphic set as the G0 or G1 set;
- above that are the shift characters that are used for the invocation of the G0 or G1 set;
- at the centre all these elements appear as a single form, the 7-bit set or 8-bit set in use.

Z.2 INTERPRETATION

The interpretation of data containing 7-bit and/or 8-bit code extension is shown by means of flowcharts. The interpretation is separate from the transformation.

Z.3 TRANSFORMATION

The transformation of the data from 7-bit to 8-bit and from 8-bit to 7-bit is shown by means of flowcharts. The transformation is separate from the interpretation.

TABLE 2 – **Summary of the given examples**

Included Sets	Bits	Structure	Interpretation	Transformation	
				7 to 8	8 to 7
C0 C1	7	Figure 11	Figure 14	Figure 16	Figure 17
	8	Figure 12	Figure 15		
G0 G1	7 and 8	Figure 13			

TABLE 3 – **Meaning of the flowchart symbols**

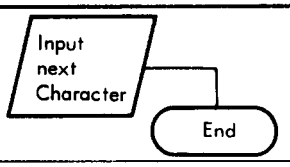
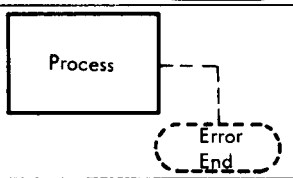
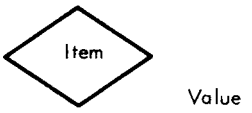
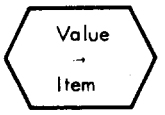
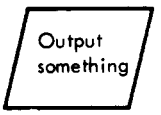
Symbol	Name	Meaning
	Input	To fetch the next character from the input character string and go to the end if there is no further character available.
	Process	To execute the defined process. There may be an error end because of lack of more characters.
	Decision	To examine the item as having the specified value.
	Preparation	To set the item to the specified value.
	Output	To take the last character (output character) or the specified character(s) (output 1/11, 4/14) or the manipulated character (output, a_8 = dropped) to the output character string.
	Predefined process	To execute the appropriate predefined process.

TABLE 4 – **Items used in the flowcharts**

Item	Meaning	Possible values	
		in 7 bits	in 8 bits
Character	The bit pattern of the current character	0/0 to 7/15	00/0 to 15/15
Column	The column number of the current character	0 to 7	00 to 15
Status 7	Shift-status of the 7-bit code	SI SO undefined	
Bits	Number of the valid bits of the character		7 8 undefined

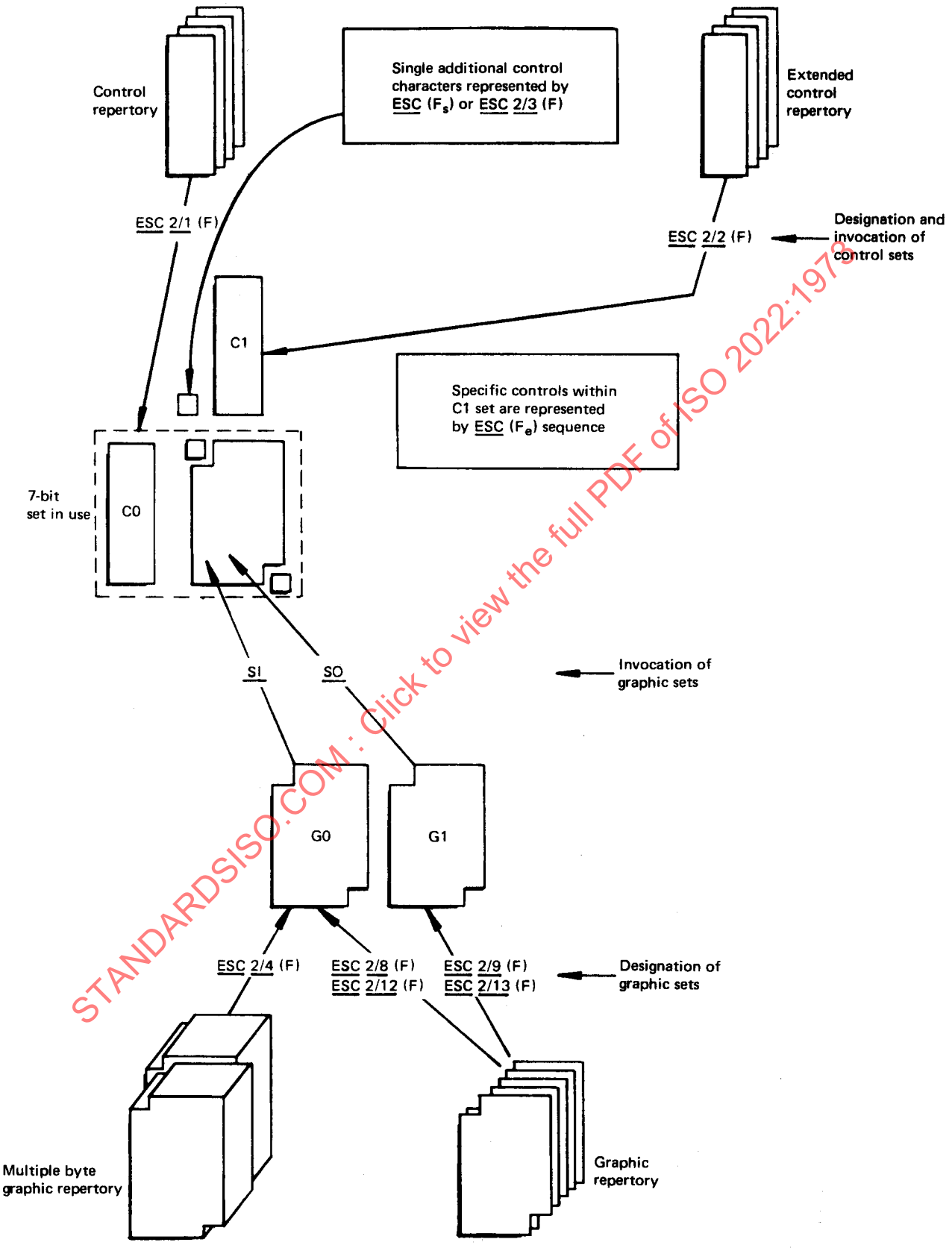


FIGURE 11 – **7-bit code extension structure using C0, C1, G0 and G1 sets**

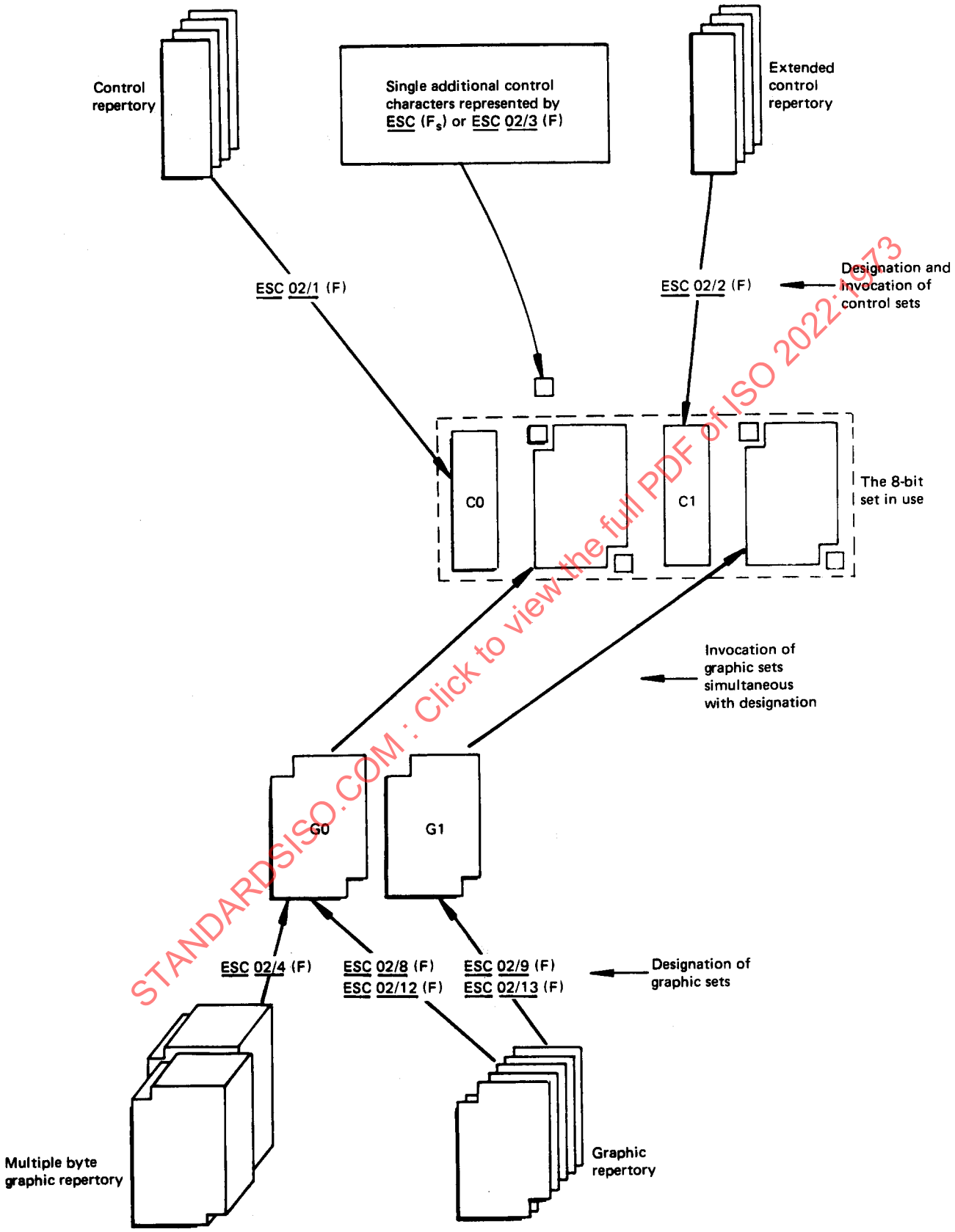


FIGURE 12 – **8-bit code extension structure using C0, C1, G0 and G1 sets**