ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-35

EXTENSION OF THE 7-BIT CODED CHARACTER SET

2nd Edition – January 1980

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône - 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-35

EXTENSION OF THE 7-BIT CODED CHARACTER SET

2nd Edition – January 1980

TABLE OF CONTENTS

Table of Contents (cont'd)

1. GENERAL

1.1 Scope

This Standard ECMA-35 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:

- the extension of the 7-bit code remaining in a 7-bit environment;
- the structure of a family of 8-bit codes;
- the extension of an 8-bit code remaining in an 8-bit environment;
- the relationship between the 7-bit code and an 8-bit code.

While the 7-bit code of Standard ECMA-6 is the agreed code for information interchange, an 8-bit code as described in this Standard is provided for information interchange within an 8-bit environment.

Code extension techniques are classified and some classes are given a structure in this Standard. Some other code extension facilities are provided in Standard ECMA-48. 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.

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.

1.2 Conformance

Full conformance to a standard means that all its requirements are met. For such conformance to be unique the standard must contain no options. This is typically the case for hardware standards.

This Standard ECMA-35 is of a different nature and as a result, it is only practicable to envisage limited conformance to it, as defined hereunder.

This Standard addresses whole classes of provisions, and it is not intended that they are all implemented in all instances.

Under limited conformance the following is required:

i) When code extension techniques are used, the applicable parts of this Standard shall be followed.

ii) When two systems with different levels of implementation of code extension techniqucs are required to communicate with one another, they shall do so using the code extention techniques they have in common.

iii) Code extension techniques not described in this Standard shall not be used.

2. FIELD OF APPLICATION

The 7-bit code of Standard ECMA-6 allows, through its different versions, the representation of up to 128 characters. Additionally, that standard allows the representation of other graphics by the combination of two graphic characters with the character BACKSPACE. In some instances the code of ECMA-6 lacks sufficient control functions 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 Standard ECMA-35.

This 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 function 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:

- the structure of the 7-bit code;
- the extension of the 7-bit code, remaining in a 7-bit environment and making use of code extension techniques;
- increasing the number of bits to 8, yet retaining a structure compatible with the 7-bit structure;
- 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:

- reducing the risk of conflict between systems required to inter-operate;
- permitting provision for code extension in the design of systems;
- providing standardized methods of calling into use agreed sets of characters;
- allowing the interchange of data between 7-bit and 8-bit environments, etc.

This Standard also describes the structure of families of codes which are related to the code of ECMA-6 by their structure.

3. REFERENCES

ECMA-6	7-Bit Input/Output Coded Character Set
ECMA-43	8-Bit Coded Character Set
ECMA-48	Additional Control Functions for Character-Imaging I/O Devices

4. DEFINITIONS AND NOTATION

4.1 Definitions

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

4.1.1 Bit combination: An ordered set of bits that represents a character.

4.1.2 Byte: A bit string that is operated upon as a unit and the size of which is independent of redundancy or framing techniques.

4.1.3 Character: A member of a set of elements that is used for the organization, control or representation of data.

4.1.4 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.

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

4.1.6 Code Table: A table showing the character corresponding to each bit combination in a code.

4.1.7 Control Character: A control function the coded representation of which consists of a single bit combination.

4.1.8 Control Function: An action that affects the recording, processing, transmission or interpretation of data. The coded representation of a control function consists of one or more bit combinations.

4.1.9 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.

4.1.10 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.

4.1.11 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 character ESCAPE.

4.1.12 Final character: The character the bit combination of which terminates an escape sequence.

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

4.1.14 Intermediate character: A character the bit combination of which occurs between the ESCAPE character and the Final character in an escape sequence consisting of more than two bit combinations.

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

4.1.17 To represent:

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

ii) To use an escape sequence with the meaning of an additional control function.

4.1.18 Version of the code

A version of the 7-bit code is a code table in which all options left open by Table 1 of ECMA-6, i.e. those affected by Notes 2 to 5 have been exercised. A single character must be allocated to each of the positions for which this freedom exists or it must be declared unused.

4.2 Notation

In this Standard the following notations are used:

Bits of a 7-bit combination:	-	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Bits of an 8-bit combination:	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_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 b_8 - b_5) and the row number is the decimal equivalent of bits b_4 - b_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 SPACE in the 7-bit code table is 2/0; the position of the same character in an 8-bit code table is 02/0.

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 ECMA-6 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 character SPACE 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 character DELETE in position 7/15.

This is shown in Figure 1.

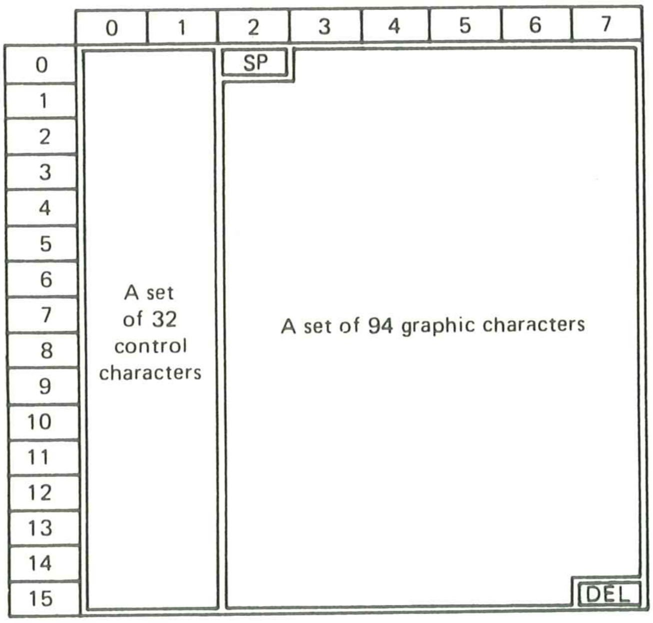
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SP
A set of 32 control characters
A set of 94 graphic characters
DEL

FIGURE 1

5.1.2 Extension by substitution

In many cases the provisions of ECMA-6 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 ECMA-6 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 control functions and/or graphic characters are required.

5.1.3 Extension by increasing the repertoire of characters

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

- additional single control functions,
- additional sets of 32 control functions,
- additional sets of 94 graphic characters,
- 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 and control functions,
- G0 set: a set of 94 graphic characters (columns 2 to 7) (a multiple byte set also functions as the G0 set),
- G1, G2, G3 sets: additional sets of 94 graphic characters.

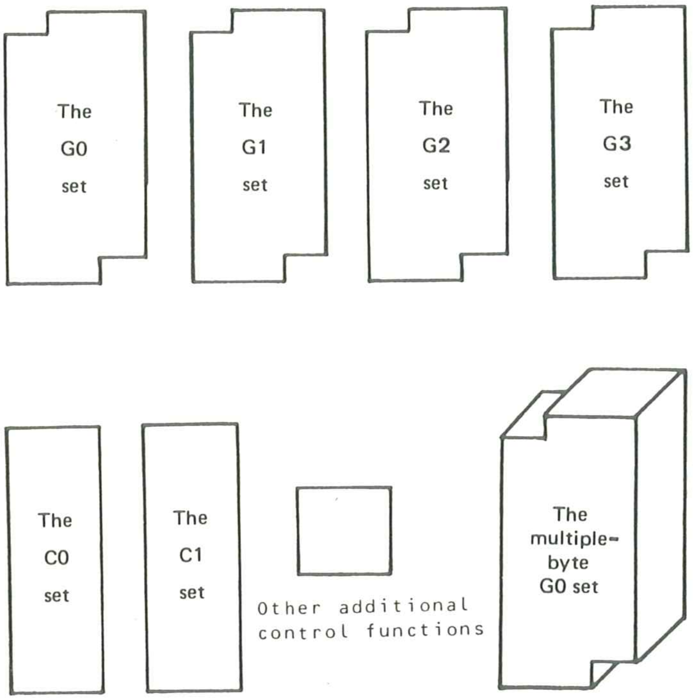

FIGURE 2

NOTE 1

It is intended that a set of control characters and a set of graphic characters which are permitted by ECMA-6, 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 Standard:

- a version according to ECMA-6,
- a compatible variant.

A compatible variant is defined as a set which is compatible with ECMA-6 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 ECMA-6 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).
- Other sets structured as in 5.1.1 above. To be able to provide the facilities of code extension of this Standard, the characters ESCAPE, SHIFT-OUT and SHIFT-IN must remain unaltered in their meanings and their positions in the code table.

5.1.6 Code extension characters of ECMA-6

In ECMA-6 the following characters are provided for the purpose of code extension:

- ESCAPE	ESC
- SHIFT-OUT	SO
- SHIFT-IN	SI
- DATA LINK ESCAPE	DLE

This Standard does not describe the use of DATA LINK ESCAPE which is reserved for the provision of additional transmission control functions. The use of this character is specified in other ECMA standards.

5.1.7 Other code extension characters

In addition this Standard includes provision for two single-shift characters which are not included in ECMA-6; they are SINGLE-SHIFT 2 (SS2) and SINGLE-SHIFT 3 (SS3).

NOTE 2

These two control characters are defined in Standard ECMA-48 and a coded representation for them is given in that document. Another coded representation for SS2 is given in one of the sets of control characters included in the "ISO Register of Character Sets to be Used with Escape Sequences", issued under the provisions of ISO 2375. The registration number of that set is 036.

5.2 Extension of the Graphic Set by Means of Shift Characters

The shift characters in this Standard are:

SI, SO, SS2, SS3.

5.2.1 Use of SHIFT-OUT and SHIFT-IN

The characters SHIFT-OUT (SO) and SHIFT-IN (SI) are used exclusively for extension of the graphic set.

The character SO invokes an additional set of 94 graphics: the G1 set. This set replaces 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:

- those corresponding to the control characters in columns 0 and 1 and position 7/15;
- the one corresponding to the character SPACE in position 2/0;
- those included in any escape sequence;
- the one following SS2 or SS3.

The character SPACE 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 SPACE in 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 escape 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 escape 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.

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.

Figure 3 is a schematic representation of the above.

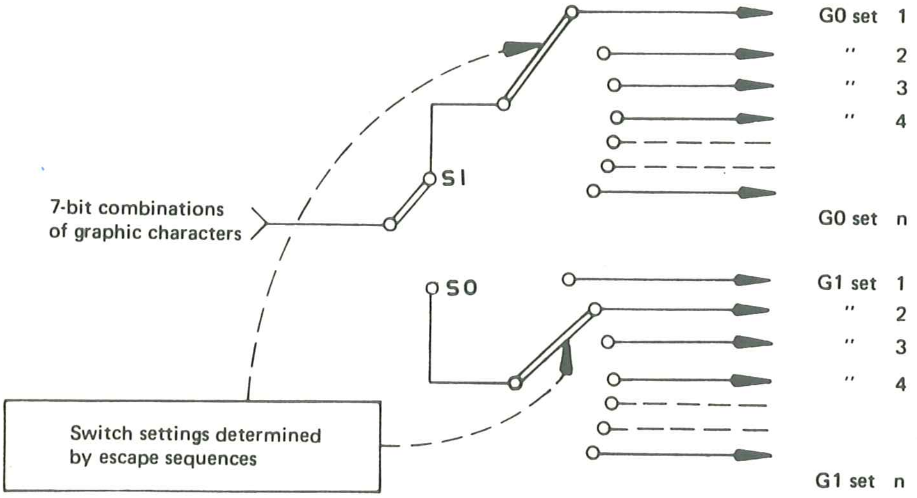
GO set 1
" 2
" 3
" 4
GO set n
SI
7-bit combinations of graphic characters
SO
G1 set 1
" 2
" 3
" 4
G1 set n
Switch settings determined by escape sequences

FIGURE 3

5.2.4 The use of single-shift characters

The single-shift characters SS2 and SS3 are used exclusively for extension of the graphic   set. SS2 invokes one character from the last designated G2 set. SS3 invokes one character from the last designated G3 set.

These invocations alter the meaning of the immediately following bit combination only and ascribe to it the meaning of the corresponding bit combination of the G2 or G3 set. The bit combination permitted to follow SS2 or SS3 is limited to one of those from 2/1 to 7/14 inclusive (see 7.4). The use of a single-shift character does not affect the current shift status established by SI or SO.

5.2.5 Unique single-shift set

Some applications require the use of only one G2 and/or one G3 set. Such a set is identified either by an appropriate escape sequence as described in 5.3.7 or by agreement between the interchanging parties.

5.2.6 Multiple single-shift set

If two or more G2 sets are required to co-exist in a system the set to be used next is designated by the appropriate escape sequence. The same applies to two or more G3 sets.

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, (sets of 94 graphic characters or sets with multiple byte representation) to designate and invoke a complete code, to announce the extension facilities used, or for private use.

Thus escape sequences are required to provide, for example:

- a single control function 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 that of the code.

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 ESCAPE and the last is always that of the Final character. 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.

NOTE 3

Although in this Standard, escape sequences are described in terms of characters of the International Reference Version (IRV) of ECMA-6 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 assigned to these bit combinations taken individually.

Intermediate characters are the 16 characters of column 2 of the IRV table.

NOTE 4

In this Standard, any one of these 16 intermediate characters is denoted by the symbol: I.

Final characters are the 79 characters of colums 3 to 7 of the IRV table excluding position 7/15.

NOTE 5

In this 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 escape 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 Standard.

5.3.3 Categories of escape sequence

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

NOTE 6

The implementors of any private escape sequence described as such in this Standard 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 functions.

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

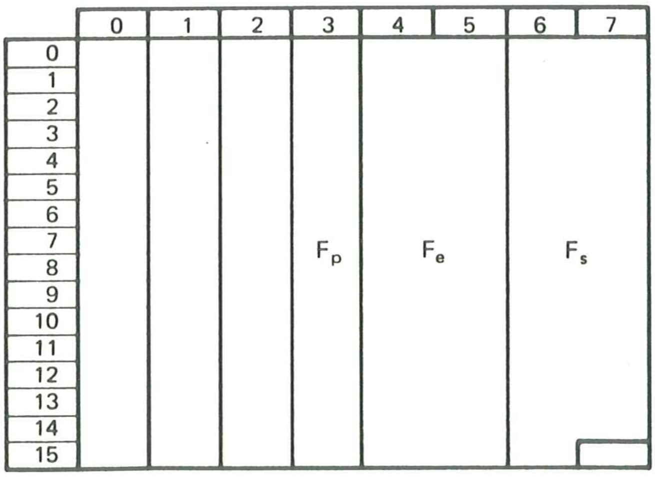

FIGURE 4

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

An ESC Fe sequence represents, depending on the Final character used, an individual control function of an additional standardized set of 32 control functions (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 escape sequence, as described in 5.3.6, or by agreement between the interchanging parties. If more than one additional set of control functions are required to co-exist in a system, the set to be used next is designated and invoked by the appropriate escape sequence.

An ESC Fp sequence represents, depending on the Final character used, a single additional control function 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 on page 19).

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

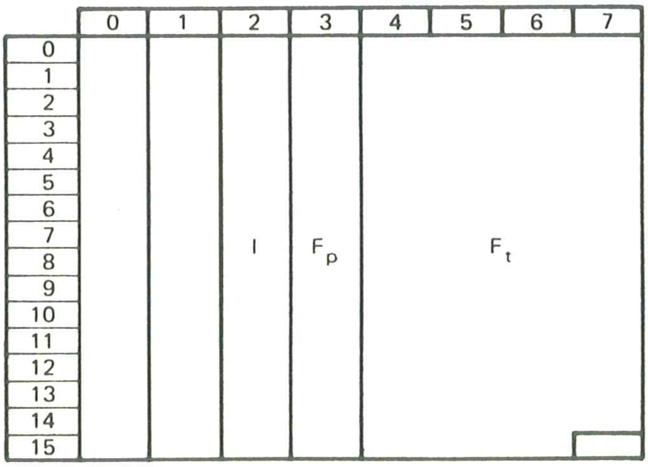

FIGURE 5

ESC I Ft 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 functions

ESC 2/3 F represents a single additional control function 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 IRV 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 functions for representation by ESC F_e

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

Individual control functions 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 functions.

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.

ESC 2/10 F and ESC 2/14 F designate sets of 94 graphic characters which will be used as the G2 set. The designated set is invoked by SS2.

ESC 2/11 F and ESC 2/15 F designate sets of 94 graphic characters which will be used as the G3 set. The designated set is invoked by SS3.

NOTE 7

For each of the G0, G1, G2 and G3 sets, two groups of escape sequences are mentioned above. Each pair together provides a single repertory of graphic sets which may be designated either as a G0, G1, G2 or G3 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 Standard. There are therefore 126 (2 x 63) such sets possible without requiring further extension.

5.3.8 Complete codes

ESC 2/5 F designates and invokes a complete code. By a complete code is meant one containing all characters needed, both control and graphic characters. If that code does not accord with the structure defined in this Standard, it may require special attention.

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

Final in column	Broad Classification
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

Each of the above classifications allows for no more than sixteen different codes. If more are required, a second Intermediate character is necessary; see 5.3.12.

5.3.9 Set 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:

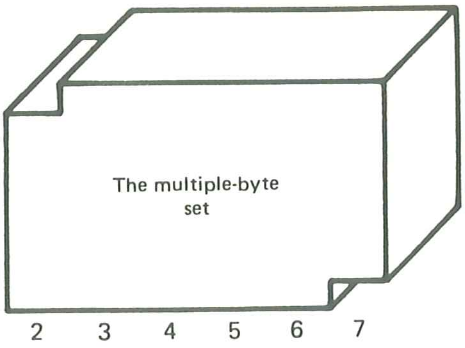

FIGURE 6

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 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:

- The first Intermediate character will indicate the class of usage identical with three-character escape sequences above.

- 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.

- All escape sequences having four or more characters whose Final character is of the F_t type are reserved for further standardization.

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

5.4 Omission of Escape Sequences

If the interchanging parties have agreed in advance upon no more than one of the G0, G1, G2, G3, C0 and C1 sets, 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 summarizes, in a schematic form, the standard means of code extension within a 7-bit environment.

Table 1 summarizes the assignment of Intermediate characters in escape sequences.

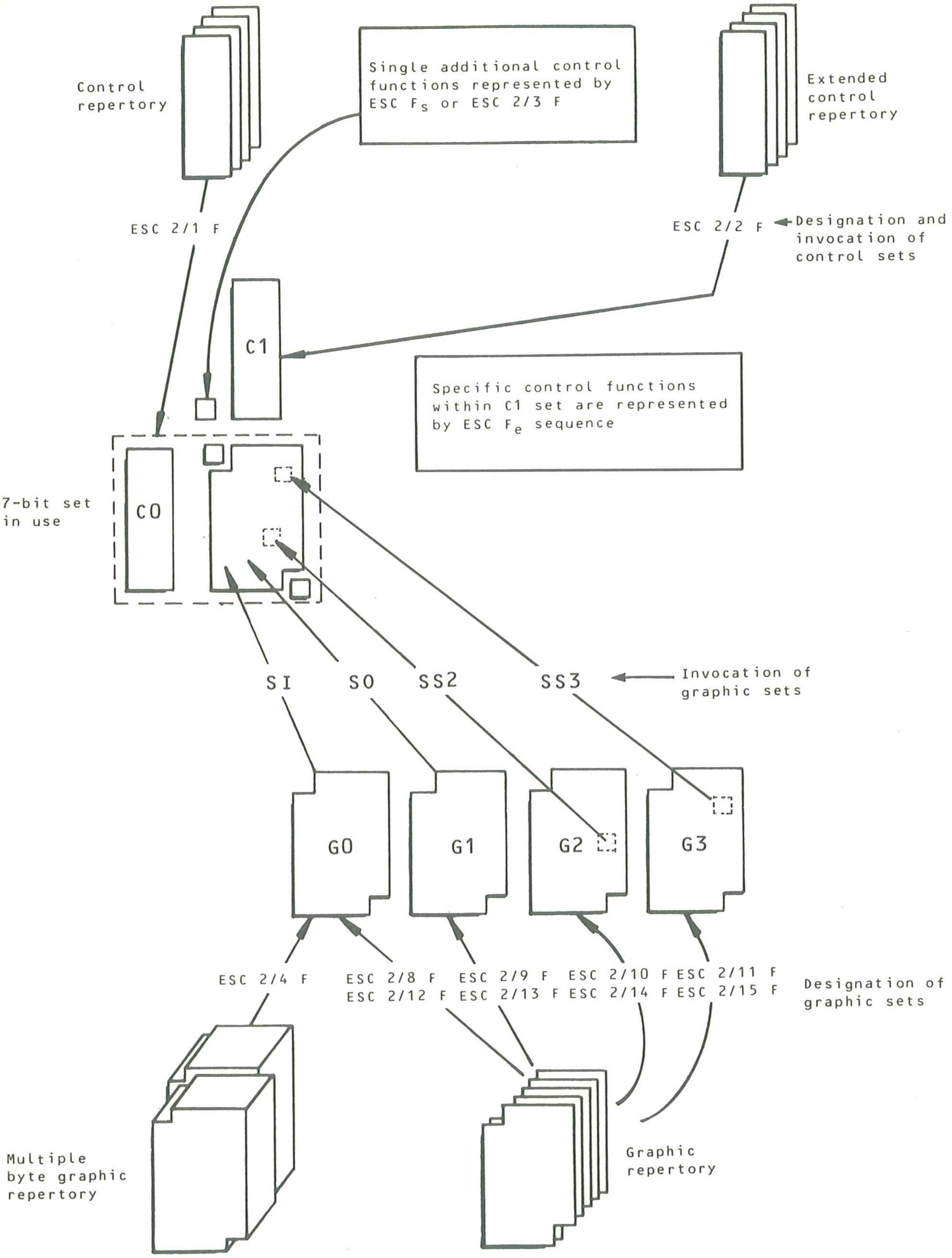
Control repertory
Single additional control functions represented by ESC F$_s$ or ESC 2/3 F
Extended control repertory
ESC 2/1 F
ESC 2/2 F
Designation and invocation of control sets
C1
Specific control functions within C1 set are represented by ESC F$_e$ sequence
7-bit set in use
C0
SI
SO
SS2
SS3
Invocation of graphic sets
G0
G1
G2
G3
ESC 2/4 F
ESC 2/8 F
ESC 2/12 F
ESC 2/9 F
ESC 2/13 F
ESC 2/10 F
ESC 2/14 F
ESC 2/11 F
ESC 2/15 F
Designation of graphic sets
Multiple byte graphic repertory
Graphic repertory

FIGURE 7

TABLE 1

Code Position	Bits of Intermediate characters $b_7b_6 \ldots b_1$	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	CONTROL FUNCTIONS	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 control functions	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	Complete codes		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 *(see Note)*	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		G2 set	
2/11	0 1 0 1 0 1 1		G3 set	
2/12	0 1 0 1 1 0 0		G0 set	
2/13	0 1 0 1 1 0 1		G1 set	
2/14	0 1 0 1 1 1 0		G2 set	
2/15	0 1 0 1 1 1 1		G3 set	

NOTE 8

There is a single repertory of sets of 94 graphic characters. Any member of the repertory may be designated as either a G0, G1, G2 or G3 set. Eight designating escape sequences, two for each of the G0, G1, G2 and G3 sets, 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 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 the eighth bit of which 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 Standard. The 128 additional bit combinations, the eighth bit of which 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).

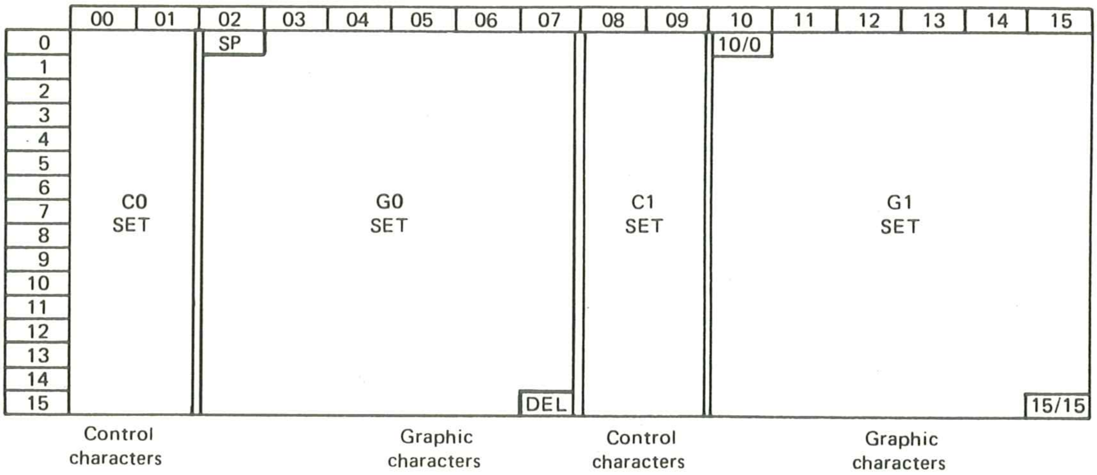

FIGURE 8

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 character DATA LINK ESCAPE and is the subject of other ECMA standards.

6.2 The Family Concept

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

- a set of 32 additional control characters can be selected for columns 08 and 09;
- 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 control functions 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 Standard have been purposely made compatible with those used to extend the 7-bit code.

The character ESCAPE is used in an 8-bit code in exactly the same way as in the 7-bit code to construct escape sequences. The meanings of these sequences are not altered in an 8-bit code. All characters in columns 08 to 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 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 combinations of columns 10 to 15.

7.2 Extension of the Graphic Set by Means of Single-Shift Characters

Use of the single-shift characters in an 8-bit code is identical to their use in a 7-bit code (see 5.2). The bit combination following SS2 or SS3 is a character from columns 02 to 07 except positions 02/0 and 07/15. All characters in columns 08 to 15 are excluded from assignment to the bit combination following SS2 or SS3 (see 9.4).

7.3 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.

7.3.1 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 control functions 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 Standard but should they occur their meaning is the same as in the 7-bit environment.

7.3.2 Three-character escape sequences

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

7.3.3 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).

7.4 Pictorial Representation

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

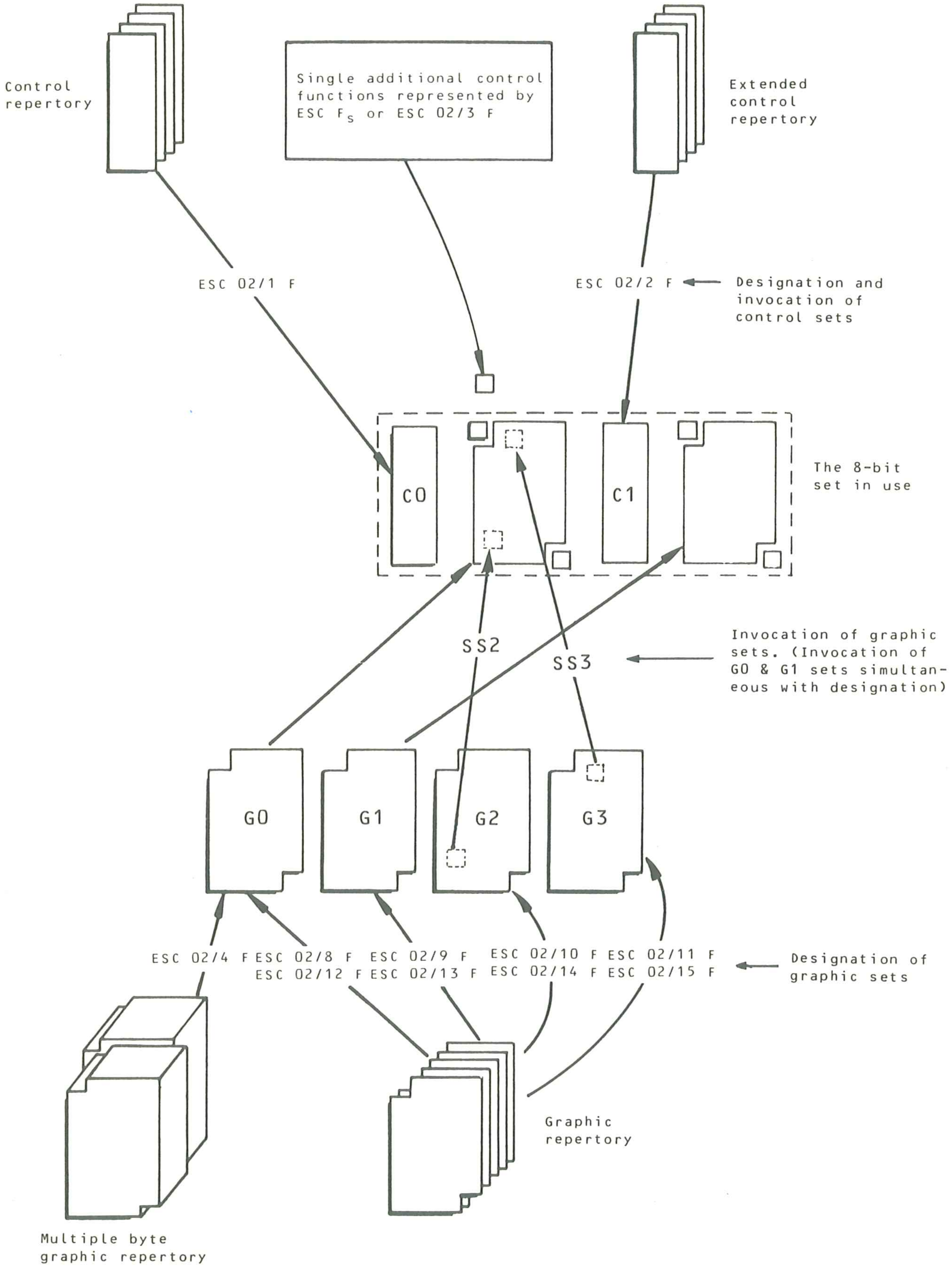
Control repertory
Single additional control functions represented by ESC F_s or ESC 02/3 F
Extended control repertory
ESC 02/1 F
ESC 02/2 F
Designation and invocation of control sets
C0
C1
The 8-bit set in use
SS2
SS3
Invocation of graphic sets. (Invocation of G0 & G1 sets simultaneous with designation)
G0
G1
G2
G3
ESC 02/4 F
ESC 02/8 F
ESC 02/9 F
ESC 02/10 F
ESC 02/11 F
ESC 02/12 F
ESC 02/13 F
ESC 02/14 F
ESC 02/15 F
Designation of graphic sets
Graphic repertory
Multiple byte graphic repertory

FIGURE 9

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-bit 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 9

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.

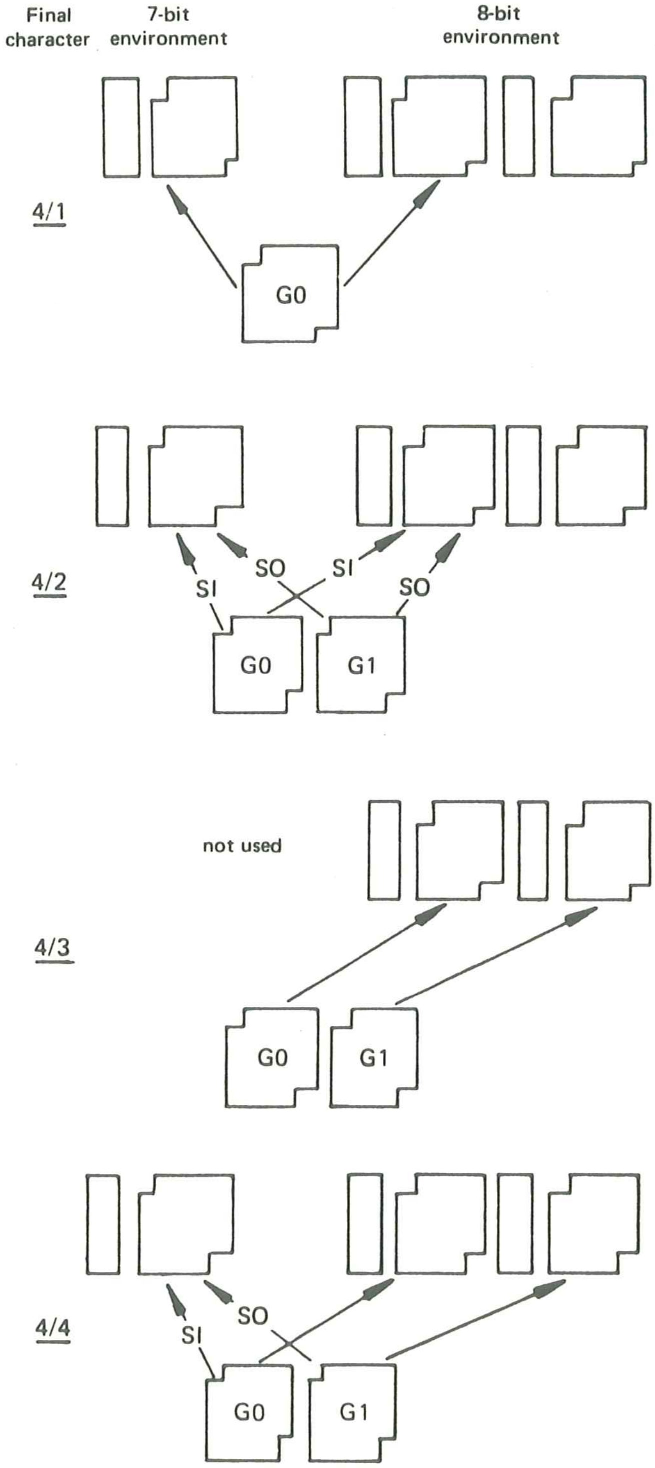
Final character
7-bit environment
8-bit environment
4/1
G0
4/2
SI
SO
SI
SO
G0
G1
not used
4/3
G0
G1
4/4
SI
SO
G0
G1

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.

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, b_8 is set to ZERO.

Indication that true 8-bit coded data follow is achieved by the occurrence of one 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, or 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 Standard. If there is a requirement to represent these positions in a 7-bit environment, a private escape sequence shall be used.

9.4 Interaction of Shift Characters

If 7-bit coded data employing SINGLE SHIFT and SHIFT-IN/ SHIFT-OUT facilities are transformed into 8-bit coded form, the normal rules for transformation may cause the bit combination following SS2 and SS3 to have its most significant bit changed from ZERO to ONE. To accord with the definitions in 5.2.4 and 7.2, only the seven least significant bits shall be given significance.

Similarly, transformation of 8-bit coded data employing SINGLE SHIFT facilities into 7-bit coded form may result in a SHIFT-IN or a SHIFT-OUT character being inserted immediately after the single-shift character. This additional SHIFT-IN or SHIFT-OUT character shall be disregarded, in so far as interpretation of the SINGLE SHIFT character is concerned, and the following bit combination shall be interpreted as representing a character from the G2 or the G3 set.

10. SPECIFIC MEANING OF ESCAPE SEQUENCES

10.1 The meanings of individual escape sequences are not specified in this 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/TC97/SC2) using the same procedures mentioned in 10.1.

APPENDIX

DIFFERENCES BETWEEN THE 1ST AND THE 2ND EDITION OF THIS STANDARD

i) Editing

The editing of the 2nd edition has been aligned on that of the 2nd edition of ISO 2022.

ii) G2 and G3 Sets

The concept of two additional graphic sets of 94 characters each was introduced, along with two non-locking shift control characters SINGLE-SHIFT 2 (SS2) and SINGLE-SHIFT 3 (SS3). The necessary form of escape sequences for designating and invoking G2 and G3 graphic sets were added.

These changes permit the designation of 376 graphic characters concurrently.

iii) Designation of complete 7-bit and 8-bit Codes

The 7-bit and 8-bit codes designated and invoked by ESC 2/5 F sequences are no longer limited to codes requiring special interpretation. This change will permit a single escape sequence to designate and invoke a complete 7-bit or 8-bit set structured in accordance with this Standard.