

**ECMA**  
EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

---

**STANDARD ECMA-35**

**CODE EXTENSION TECHNIQUES**

4th Edition – March 1985

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**

**CODE EXTENSION TECHNIQUES**

4th Edition – March 1985

### BRIEF HISTORY

TC1 participate very actively in the work of ISO/TC97/SC2 on code extension and contributed numerous technical papers to SC2/WG-1, the group entrusted with the preparation of ISO 2022, the international standard for code extension techniques. ECMA published its first Standard ECMA-35 on the same subject in 1971. Two further editions in 1980 and 1982 reflected the progress achieved internationally. The present edition is technically identical with the 1985 edition of ISO 2022.

Adopted by the General Assembly of ECMA on Dec. 13, 1984.

## TABLE OF CONTENTS

	<u>Page</u>
1. GENERAL	1
1.1 Scope	1
1.2 Field of Application	1
2. CONFORMANCE	2
3. REFERENCES	3
4. DEFINITIONS AND NOTATION	3
4.1 Definitions	3
4.1.1 Bit combination	3
4.1.2 Byte	3
4.1.3 Character	3
4.1.4 Coded character set; Code	3
4.1.5 Code extension	3
4.1.6 Code table	3
4.1.7 Control character	3
4.1.8 Control function	4
4.1.9 To designate	4
4.1.10 Environment	4
4.1.11 Escape sequence	4
4.1.12 Final character	4
4.1.13 Graphic character	4
4.1.14 Graphic symbol	4
4.1.15 Intermediate character	4
4.1.16 To invoke	4
4.1.17 Position	4
4.1.18 To represent	5
4.1.19 Version of the 7-bit code	5
4.1.20 Version of the 8-bit code	5
4.2 Notation	5
5. EXTENSION OF THE 7-BIT CODE REMAINING IN A 7-BIT ENVIRONMENT	5
5.1 Introduction	5
5.1.1 Structure of the 7-bit code	5
5.1.2 Extension by substitution	6
5.1.3 Extension by increasing the repertoire of characters	6
5.1.4 Elements of code extension	7
5.1.5 Compatibility	8
5.1.6 Code extension characters of ECMA-6	8
5.1.7 Other code extension characters	8
5.1.8 Combination of graphic characters	9
5.2 Extension of the Graphic Set by Means of Shift Functions	9
5.2.1 Use of locking-shift functions	9
5.2.2 Use of single-shift functions	10

TABLE OF CONTENTS (cont'd)

	<u>Page</u>
5.2.3 Unique additional graphic sets	10
5.2.4 Multiple graphic sets	11
5.3 Code Extension by Means of Escape Sequences	12
5.3.1 Purposes of escape sequences	12
5.3.2 Structure of escape sequences	13
5.3.3 Categories of escape sequence	13
5.3.4 Single additional control functions	16
5.3.5 Sets of 32 control characters for columns 0 and 1	16
5.3.6 Sets of 32 control functions for representation by ESC Fe	17
5.3.7 Sets of 94 graphic characters	17
5.3.8 Sets of 96 graphic characters	17
5.3.9 Set of graphic characters with multiple-byte representation	18
5.3.10 Dynamically Redefinable Character Sets (DRCS)	19
5.3.11 Other coding systems	20
5.3.12 Announcement of extension facilities	21
5.3.13 Revision of Registered Sets	21
5.3.14 Three-character escape sequences without assigned meanings	21
5.3.15 Summary of assignments of Intermediate characters	22
5.4 Initial Designation and Invocation	23
5.5 Pictorial Representation of Code Extension in 7-bit Environment	23
6. STRUCTURE OF A FAMILY OF 8-BIT CODES	25
6.1 The 8-bit Code Table	25
6.2 The Family Concept	25
7. USE OF CODE EXTENSION IN AN 8-BIT CODE	25
7.1 Elements of Code Extension in an 8-Bit Environment	26
7.2 Extension of the Graphic Set by Means of Shift Functions	27
7.2.1 Use of locking-shift functions	27
7.2.2 Use of single-shift functions	27
7.3 Code Extension by Means of Escape Sequences	28
7.3.1 Two-character escape sequences	28
7.3.2 Three-character escape sequences	28
7.3.3 Escape sequences with four or more characters	28
7.4 Sets of Graphic Characters with Multiple-Byte Representation	28
7.5 Compatibility	29
7.6 Pictorial Representation of Code Extension in 8-bit Environment	29
8. ANNOUNCEMENT OF EXTENSION FACILITIES USED	31
8.1 General	31
8.2 Restrictions	31

TABLE OF CONTENTS (cont'd)

	<u>Page</u>
9. RELATIONSHIP BETWEEN 7-BIT AND 8-BIT CODES	35
9.1 Transformation Between 7-bit and 8-bit Codes	35
9.2 Representation of the 7-bit Code in an 8-bit En- vironment	35
9.3 Interaction of Shift Functions	35
9.4 Preservation of Information on Retransformation	35
10. RELATIONSHIP WITH PRESENTATION PROTOCOL	36
11. SPECIFIC MEANINGS OF ESCAPE SEQUENCES	36
APPENDIX A - THE INTERNATIONAL REGISTER OF CODED CHARACTER SETS TO BE USED WITH ESCAPE SEQUENCES	37
APPENDIX B - SHIFT FUNCTIONS	38
APPENDIX C - SUMMARY OF THE ESCAPE SEQUENCES DEFINED IN THIS STANDARD	39
APPENDIX D - MAIN DIFFERENCES BETWEEN THE 3rd EDITION (1982) 40 AND THE PRESENT 4th EDITION OF THIS STANDARD	

## 1. GENERAL

### 1.1 Scope

This ECMA 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 clauses 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.

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

Code extension techniques are classified and some classes are given a structure in this Standard. Specific assignments of bit combinations to relate individual character sets and control functions with their invocation or designation are to be made in accordance with ISO 2375 (see Appendix A).

### 1.2 Field of Application

While the 7-bit code of 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.

The 7-bit code of ECMA-6 allows the representation of up to 128 characters. Additionally, that Standard allows the representation of other graphic characters by the combination of two or more graphic characters with the control characters BACKSPACE or CARRIAGE RETURN. In some instances, the character set of ECMA-6 lacks sufficient control functions or graphic characters to satisfy the needs of an application. These needs may be satisfied by means of code extension which is the subject of this Standard.

The principles established in this Standard may be utilized to form supplementary code extension facilities. For example Standard ECMA-48 has followed such a procedure to formulate some parameterized control functions.

This Standard presents a review of the salient structure of the 7-bit code and then builds upon that structure to specify various means of extending the control function and graphic sets of the code. It also specifies 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. It describes:

- i) the structure of the 7-bit code,
- ii) the extension of the 7-bit code, remaining in a 7-bit environment and making use of code extension techniques,
- iii) the structure of a family of 8-bit codes, remaining compatible with the 7-bit structure,
- iv) the extension of an 8-bit code, remaining in an 8-bit environment, and making use of code extension techniques.

In order to use identical techniques in each of the above cases, and to facilitate conversion between them, standard rules for code extension 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.

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-length record processing may have undesirable results or may require additional special treatment to ensure correct interpretation.

## 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 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 the code extension techniques described in this Standard are used, they shall be implemented by the control functions defined in this Standard with the meaning and coded representation specified in this Standard;
- ii) when two systems with different levels of implementation of code extension techniques are required to communicate with one another, they shall do so using the code extension techniques they have in common;

- iii) no coded representation that is either reserved for registration and not assigned or reserved for future standardization shall be used;
- iv) no registered escape sequence shall be used with a meaning different from that defined by the registration.

### 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
- ISO 2375 : Procedure for registration of escape sequences.

### 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 or is used as a part of the representation of 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 used for the organization, control or representation of data.

##### 4.1.4 Coded character set; Code

A set of unambiguous rules that establishes a character set and the one-to-one relationship between the characters of the set and their bit combinations.

##### 4.1.5 Code extension

The 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 allocated 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 and that has a coded representation consisting 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 bit combinations represents 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 function, that has a visual representation normally handwritten, printed or displayed.

4.1.14 Graphic symbol

A visual representation of a graphic character or of a control function.

4.1.15 Intermediate character

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

4.1.16 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.

4.1.17 Position

That part of a code table identified by its column and row coordinates.

#### 4.1.18 To represent

- To use a prescribed bit combination with the meaning of a character in a set of characters that has been designated and invoked; or
- to use an escape sequence with the meaning of an additional control function.

#### 4.1.19 Version of the 7-bit code

A 7-bit coded character set in which all options left open in ECMA-6 have been exercised. A single character must be allocated to each of the bit combinations for which this freedom exists or the bit combination must be declared unused.

#### 4.1.20 Version of the 8-bit code

An 8-bit coded character set in which all options left open in ECMA-43 have been exercised. A single character must be allocated to each of the bit combinations for which this freedom exists or the bit combination must be declared unused.

### 4.2 Notation

In this Standard the following notations are used:

Bits of a 7-bit combination	-	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>
Bits of an 8-bit combination	b <sub>8</sub>	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>
Bit weight for column and row reference	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
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<sub>7</sub> to b<sub>5</sub> (or b<sub>8</sub> to b<sub>5</sub>) and the row number is the decimal equivalent of bits b<sub>4</sub> to b<sub>1</sub>, giving to these bits the weights shown above. The column and row numbers are separated by a solidus.

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 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:

- a) columns 0 and 1 contain a set of 32 control characters,
- b) columns 2 to 7 contain either:
  - 1) the character SPACE in position 2/0, which may be regarded as a control character or a graphic character, the character DELETE in position 7/15 and a set of 94 graphic characters in positions 2/1 to 7/14,
  - or
  - 2) a set of 96 graphic characters in positions 2/0 to 7/15.

This is shown in Figure 1.

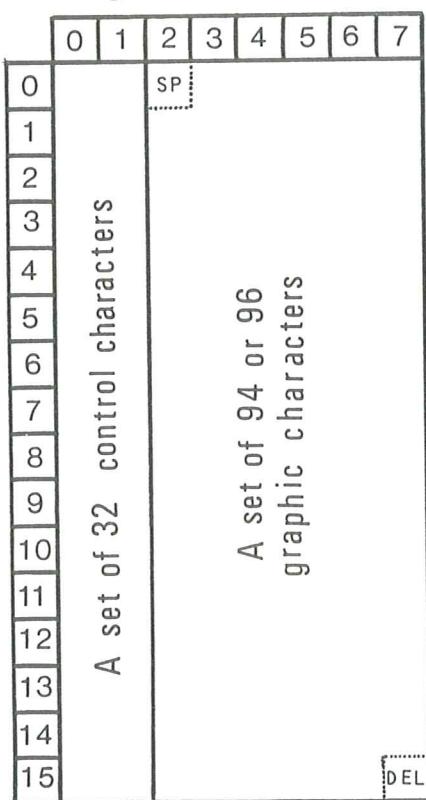


Figure 1 - 7-bit code structure

#### 5.1.2 Extension by substitution

In many cases, the provisions of ECMA-6 satisfy the requirements of an application. Other applications are 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 shall be regarded as constituting a new code, outside the provisions of ECMA-6.

#### 5.1.3 Extension by increasing the repertoire of characters

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

- i) additional single control functions,
- ii) additional sets of 32 control functions,
- iii) additional sets of 94 graphic characters,
- iv) additional sets of 96 graphic characters,
- v) additional sets of more than 94 or 96 graphic characters, each represented by more than one byte, i.e. multiple-byte sets.

Any additional set of characters, described in iii), iv) and v) above, shall not contain SPACE or any control character, such as DELETE.

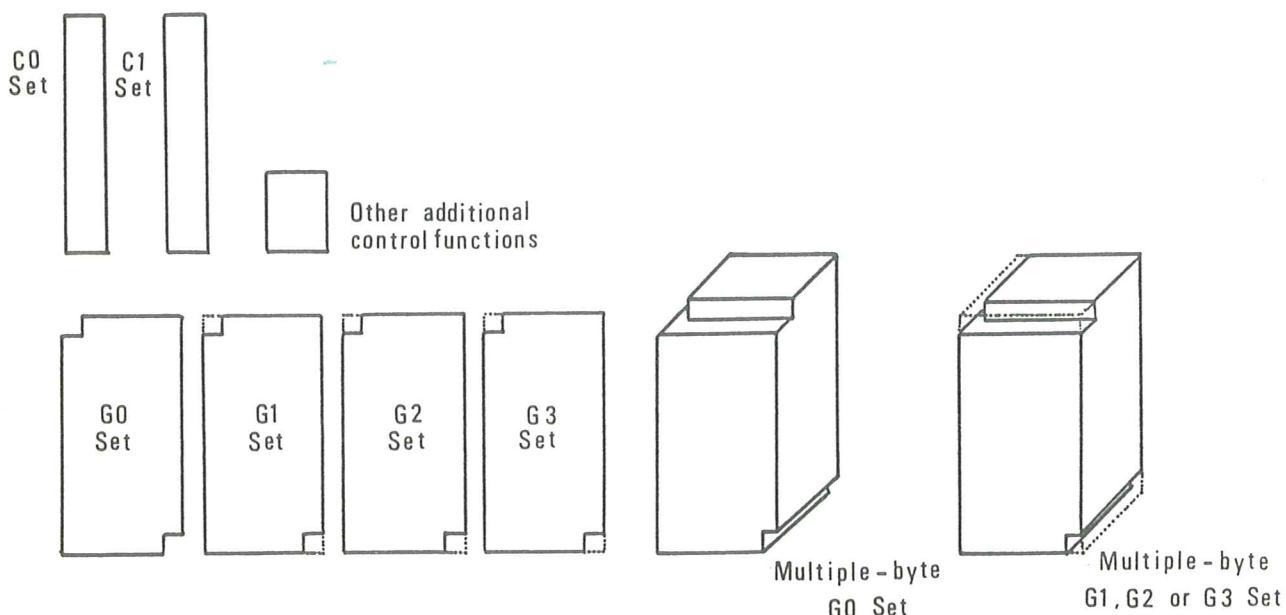
#### 5.1.4 Elements of code extension

Many applications 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 functions;
- other additional single control functions;
- G0 set: a set of 94 graphic characters (bit combinations 2/1 to 7/14), a multiple-byte set may also function as a G0 set;
- G1, G2, G3 sets: additional sets of 94 graphic characters (bit combinations 2/1 to 7/14) or of 96 graphic characters (bit combinations 2/0 to 7/15); a multiple-byte set may also function as a G1, G2 or G3 set.

##### NOTE 1

*It is intended that, if they are used, a set of control characters and a set of graphic characters which are compatible with ECMA-6 (see 5.1.5) are assigned to the C0 set and the G0 set respectively.*



#### 5.1.5 Compatibility

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

- a) A version according to ECMA-6.
- b) A compatible variant of ECMA-6, i.e. a 7-bit code which is compatible with ECMA-6 inasmuch as:
  - columns 0 and 1 contain only control characters;
  - 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;
  - SPACE, DELETE and one or two sets of 94 graphic characters are allocated to columns 2 to 7; sets of 96 graphic characters, multiple-byte sets and shift functions other than SO, SI are not used,
  - graphic characters of ECMA-6 are not moved to other positions (a non-latin alphabet containing graphic characters which are also included in the Latin alphabet is not subject to this rule).
- c) Other 7-bit codes structured as in 5.1.1. Such a code may contain 94 graphic characters, 96 graphic characters and/or multiple-byte sets in columns 2 to 7. To be able to provide the facilities of code extension of this Standard, the control characters ESCAPE, SHIFT-OUT and SHIFT-IN shall 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 control 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 the control character DATA LINK ESCAPE which is reserved for the provision of additional transmission control functions. The use of this character is specified in other standards.

#### 5.1.7 Other code extension characters

For use within a 7-bit environment, this Standard includes provision of some additional shift functions which are not included in ECMA-6.

LOCKING-SHIFT TWO	LS2
LOCKING-SHIFT THREE	LS3
SINGLE-SHIFT TWO	SS2
SINGLE-SHIFT THREE	SS3

See Appendix B for the coded representation of these functions.

Three additional locking-shift functions LS1R, LS2R, LS3R are specified in clause 7.2.1 which are used in a 7-bit environment only when it is necessary to preserve their use for transformation between 7-bit and 8-bit environments (see 9.2 and 9.4). When used in a 7-bit code, LS1R, LS2R and LS3R have the same effects as SO, LS2 and LS3, respectively.

#### 5.1.8 Combination of graphic characters

Some graphic sets may allow for the representation of additional graphic characters such as accented characters by the combination of two or more graphic characters in the same character position. Two methods of combining graphic characters in a single character position are provided for:

- graphic characters having implicit forward motion (spacing characters) used in conjunction with BACKSPACE or CARRIAGE RETURN;
- graphic characters having no implicit forward motion (non-spacing characters) used in combination with spacing graphic characters.

ECMA-6 allows for the first of these two methods to represent accented characters. Sponsors of graphic sets applying for registration under the provision of ISO 2375 are expected to identify any characters in the set that are non-spacing.

##### NOTE 2

A standard defining a character set should specify any restriction on combining characters, as this is not part of registration.

#### 5.2 Extension of the Graphic Set by Means of Shift Functions

For use in a 7-bit environment, the shift functions specified in this Standard are:

SO, SI, LS2, LS3, SS2 and SS3.

##### 5.2.1 Use of locking-shift functions

In a 7-bit environment, the functions SHIFT-OUT (SO), SHIFT-IN (SI), LOCKING-SHIFT TWO (LS2) and LOCKING-SHIFT THREE (LS3) shall be used exclusively for extension of the graphic set.

The shift functions SO, LS2 or LS3 shall each invoke an additional set of 94 or 96 graphic characters: G1, G2 and G3. If the set consists of 94 characters, it is invoked into positions 2/1 to 7/14, SPACE is invoked into

position 2/0 and DELETE is invoked into position 7/15; if the set consists of 96 characters, it is invoked into positions 2/0 to 7/15. 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 previously invoked set.

The shift function SI shall invoke the 94 graphic characters of the G0 set and cause positions 2/0 and 7/15 to take their normal meanings of SPACE and DELETE, respectively.

If a particular set is already invoked, use of the corresponding shift function has no effect.

The meanings of the following bit combinations shall not be affected by the occurrence of the locking-shift functions:

- a) those representing the control characters in columns 0 and 1;
- b) those included in any escape sequence;
- c) the one following SS2 or SS3.

The characters SPACE and DELETE shall occur at position 2/0 and 7/15, respectively, if and only if a set of 94 graphic characters is invoked. They shall not be assigned to any other positions in any set. However, characters other than SPACE and representing spaces of different sizes or usage may be assigned to any positions in any set of graphic characters or control functions.

At the beginning of any information interchange the shift status shall be defined by use of one of the locking-shift functions as specified in 5.4 (see also 8.).

#### 5.2.2 Use of single-shift functions

The single-shift functions SS2 and SS3 shall be used exclusively for extension of the graphic set. SS2 shall invoke one character from the last designated G2 set. SS3 shall invoke one character from the last designated G3 set.

These invocations alter the meaning of the immediately following bit combination only (but see 5.3.9) 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 one of those from 2/1 to 7/14 for a 94-character G2 or G3 set and 2/0 to 7/15 for a 96-character G2 or G3 set (see 9.3). The use of a single-shift function does not affect the current shift status established by a locking-shift function.

#### 5.2.3 Unique additional graphic sets

Some applications require no more than three additional graphic sets of 94 or 96 characters that can be uniquely

identified as G1, G2 and G3 sets. These sets are designated by means of appropriate escape sequences as described in 5.3.7. to 5.3.10. As described in 5.4, such sequences may be omitted by agreement between interchanging parties. Any of these additional sets can then be invoked by means of the corresponding shift functions.

#### 5.2.4 Multiple graphic sets

If there is a requirement for more than three additional graphic sets or for more than one graphic set to be designated as either G0, G1, G2 or G3, it is necessary to designate the G0, G1, G2, G3 sets to be used next by means of the appropriate escape sequences as described in 5.3.7 to 5.3.10. Each subsequent use of a shift function shall invoke the corresponding currently designated set.

It is not necessary to revert to the G0 set by use of SI before designating a different set as G1, G2, G3 by means of an escape sequence.

The use of a shift function shall invoke the graphic characters of the set last designated for use by that shift function but shall not affect the identity of any sets currently designated. A designated set may be invoked any number of times by repeated use of the relevant shift function until it is superseded by another designating escape sequence.

When a further graphic set is designated by an escape sequence, the current shift status shall remain unaltered.

When a graphic set is designated by an escape sequence, and if that class of graphic set (i.e. G0, G1, G2 or G3) is currently invoked, then the new set shall also be invoked.

Figure 3 is a schematic representation of the designation and invocation processes described above.

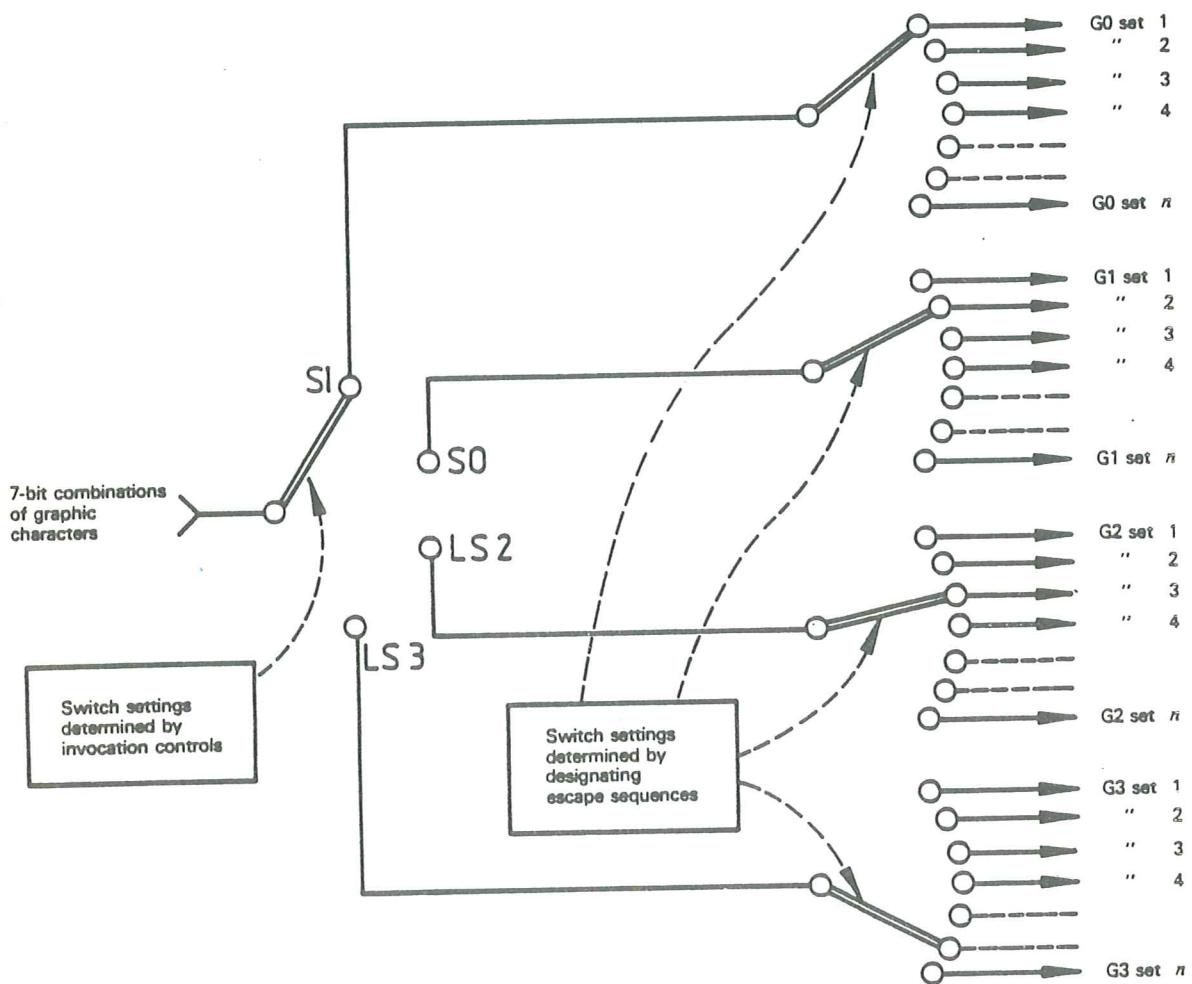


Figure 3 — Multiple graphic sets used with locking-shift functions

### 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 graphic characters, 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 function not already in the code;
- a set of control functions 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 shall consist of two or more 7-bit combinations. The first shall always be the bit combination representing ESCAPE and the last shall always be that representing the Final character. An escape sequence may also contain any number of 7-bit combinations representing Intermediate characters.

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

Intermediate characters are the 16 characters of column 2 of the 7-bit code table; they are 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; they are denoted by the symbol F.

#### NOTE 3

*Although, in this Standard, escape sequences are described in terms of characters or of positions in 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.*

The control characters in columns 0 and 1 and the character in position 7/15 shall not be used as either Intermediate or Final characters to construct an escape sequence.

#### NOTE 4

*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 sequences

The use of escape sequences is specified in this Standard. However, escape sequence with Final characters from column 3 are reserved for private use subject to the categorization outlined below. Escape sequences for private use are not subject to registration under ISO 2375.

#### NOTE 5

*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 registered 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 shall be of the form:

ESC F

Such escape sequences are used to represent additional control functions.

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

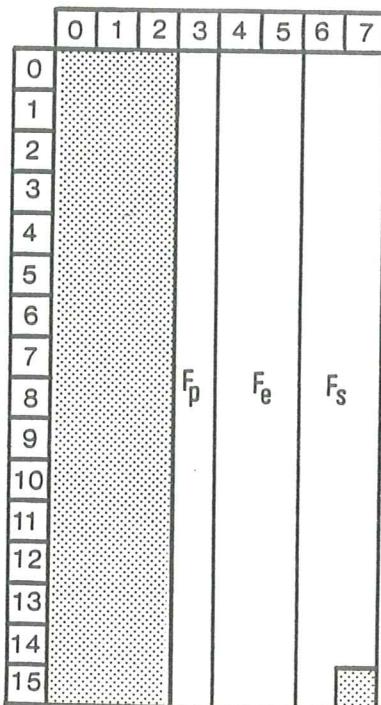


Figure 4 - Final character for two-character escape sequences

An ESC  $F_s$  sequence represents, depending on the Final character used, a single additional control function with a permanently assigned meaning. 31 Final characters of columns 6 and 7 are provided for this purpose.

#### NOTE 6

*ESC  $F_s$  sequences are registered in the ISO International Register of Character Sets to be used with Escape Sequences, which is maintained by the Registration Authority (see Appendix A). Any candidates for ESC  $F_s$  sequences must be approved by ISO/TC97/SC2 for registration. The coding for the Final character,  $F_s$  is assigned by the Registration Authority according to the procedure of ISO 2375.*

An ESC  $F_e$  sequence represents, depending on the Final character used, an individual control function of the currently designated C1 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 se-

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 F<sub>p</sub> 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 shall be of 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 5.3.13 (see Table 1).

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

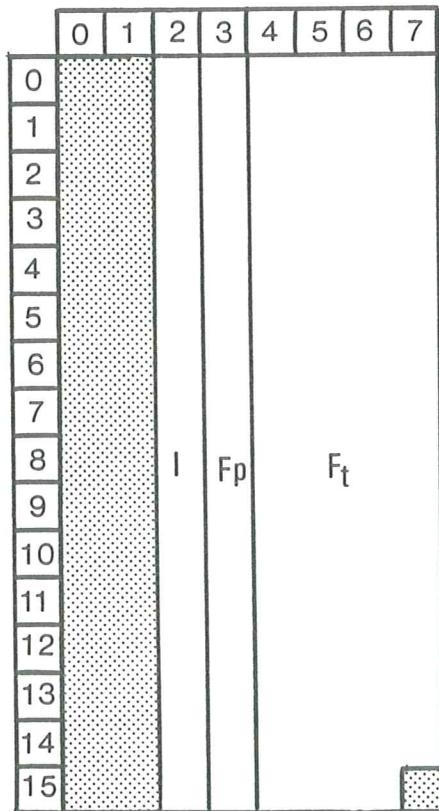


Figure 5 - Intermediate and Final characters for three-character escape sequences

ESC I Ft sequences are used for standardized purposes. 63 Ft characters of columns 4 to 7 are provided for this purpose. One such Final character, - that represented by bit combination 7/14 (when used with Intermediate characters 2/1, 2/2, 2/4, 2/8 to 2/11 or 2/13 to 2/15), shall indicate that the character set designated by the escape sequence is empty, i.e. that it does not contain any character. Bit combinations representing characters of a set which has been declared empty shall not be used. ESC I F<sub>p</sub> sequences are reserved for private use. Sixteen F<sub>p</sub> characters of column 3 are provided for this purpose.

#### 5.3.3.3 Escape sequences having four or more characters

An escape sequence having four or more characters shall be of the form

ESC I ... I F

where I ... I represents two or more characters.

Escape sequence having four or more characters shall be interpreted as follows:

- a) The first Intermediate character will indicate the same class of use as the Intermediate character of a three-character escape sequence.
- b) Except when the first Intermediate character is reserved for future standardization or is 2/0, 2/4, 2/5 or 2/6 (see 5.3.12, 5.3.9, 5.3.11 and 5.3.13 or 8, respectively), the second Intermediate character shall be used as follows:
  - 2/0 is reserved for the designation of DRCS sets (see 5.3.10);
  - 2/1 to 2/3 and any further Intermediate characters are available for registration of control functions and graphic character sets;
  - 2/4 to 2/15 are reserved for future standardization.
- c) All escape sequences that have a Final character of the F<sub>p</sub> type are reserved for private use and are not specified by this Standard.
- d) The use of 7/14 as a Final character to signify an empty set, as specified in 5.3.3.2, also applies to escape sequences having four or more characters.

#### 5.3.4 Single additional control functions

ESC 2/3 F represents a single additional control function determined by 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:

- a) inclusion of the ten transmission control characters;
- b) inclusion of the control characters NUL, SO, SI, CAN, SUB and ESC with their meanings and their positions in the 7-bit code table unaltered.

NOTE 7

*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 Fe

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 Fe sequences instead of a single bit-combination. A C1 set shall not include transmission control functions (see Note 8).

5.3.7 Sets of 94 graphic characters

ESC 2/8 F designates a set of 94 graphic characters as the G0 set. The designated set is invoked by SI.

ESC 2/9 F designates a set of 94 graphic characters as the G1 set. The designated set is invoked by SO.

ESC 2/10 F designates a set of 94 graphic characters as the G2 set. LS2 invokes the designated set and SS2 invokes one character from the designated set.

ESC 2/11 F designates a set of 94 graphic characters as the G3 set. LS3 invokes the designated set and SS3 invokes one character from the designated set.

5.3.8 Sets of 96 graphic characters

ESC 2/13 F designates a set of 96 graphic characters as the G1 set. The designated set is invoked by SO.

ESC 2/14 F designates a set of 96 graphic characters as the G2 set. LS2 invokes the designated set and SS2 invokes one character from the designated set.

ESC 2/15 F designates a set of 96 graphic characters as the G3 set. LS3 invokes the designated set and SS3 invokes one character from the designated set.

NOTE 8

When sets of characters are registered, a unique Final character is allocated to each set. In the case of control character sets, the series of Final characters for C0 sets and C1 sets are quite separate - i.e. a set is registered for use as either a C0 or a C1 set. In contrast, graphic sets are not registered as either G0, G1, G2 or G3 sets but as all four. They may be used in any of these ways by use of the appropriate Intermediate character as specified in 5.3.7, 5.3.8 and 5.3.9, except that a set of 96 characters cannot be used as a G0 set.

5.3.9 Set of graphic characters with multiple-byte representation

ESC 2/4 I F designates a set of graphic characters that are represented by two or more bytes, each corresponding to a bit combination in columns 2 to 7 (see Fig. 6).

ESC 2/4 2/8 F designates a multiple-byte graphic set as the G0 set. The designated set is invoked by SI.

ESC 2/4 2/9 F or ESC 2/4 2/13 F designates a multiple-byte graphic set as the G1 set. The designated set is invoked by S0.

ESC 2/4 2/10 F or ESC 2/4 2/14 F designates a multiple-byte graphic set as the G2 set. LS2 invokes the designated set and SS2 invokes one character from the designated set.

ESC 2/4 2/11 F or ESC 2/4 2/15 F designates a multiple-byte graphic set as the G3 set. LS3 invokes the designated set and SS3 invokes one character from the designated set.

As exception to these rules ESC 2/4 4/0, ESC 2/4 4/1 and ESC 2/4 4/2 designate multiple-byte sets as G0 sets, because they are already registered.

NOTE 9

The reason for these exceptions is that the first version of ECMA-35 allowed multiple-byte sets to be only G0 sets and used ESC 2/4 F to represent them.

A multiple-byte set designated by ESC 2/4 2/8 F, ESC 2/4 2/9 F, ESC 2/4 2/10 F or ESC 2/4 2/11 F consists of up to  $94^n$  characters each of which is represented by a sequence of n bit combinations in the range 2/1 to 7/14. A multiple-byte set designated by ESC 2/4 2/13 F, ESC 2/4 2/14 F, or ESC 2/4 2/15 F consists of up to  $96^n$  characters each of which is represented by a sequence of n bit combinations in the range 2/0 to 7/15.

Within such a set, each graphic character is represented by the same number of bytes.

If a single-shift function is used to invoke a character from a multiple-byte set, contrary to its normal usage the shift function will affect two or more bit-combinations to represent that character.

Escape sequences with 2/4 as the first Intermediate character and either 2/0 to 2/7 or 2/12 as the second Intermediate character are reserved for future standardization.

The Final character assignment is as follows:

Final character from column	Number of bytes
3	2 or more for private use
4 and 5	2
6	3
7	4 or more

In an escape sequence designating a multiple-byte set, a third Intermediate character in the range 2/1 to 2/3 and any further Intermediate characters can be used after the second Intermediate character, if more than 63 sets need to be registered.

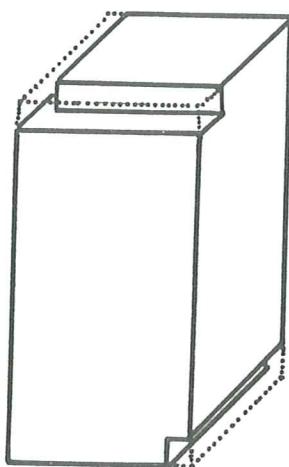


Figure 6 - Set of graphics with multiple-byte representation

#### 5.3.10 Dynamically Redefinable Character Sets (DRCS)

A DRCS is a set of graphic characters the visual appearance of which is specified and transmitted prior to the time of use. Such specification may be made explicitly or by a reference. These characters may be alphabetic, special symbols or picture element symbols. Once specified, a DRCS is regarded as a member of the repertoire of graphic character sets that can be designated by appropriate escape sequences as a G0, G1, G2 or G3 set.

ESC I 2/0 F designates such a set, where I shall be in the range 2/8 to 2/11 to indicate whether a 94-character set is to be used as a G0, G1, G2 or G3 set respectively

or in the range from 2/13 to 2/15, to indicate whether a 96-character set is to be used as a G1, G2 or G3 set respectively, in the same way as defined in 5.3.7 and 5.3.8.

126, i.e.  $2 \times 63$  sets may be identified by means of such four character sequences. This should be enough for most requirements but one or more additional Intermediate characters can be inserted between the second Intermediate character and the Final character if more sets are needed.

Escape sequences with 2/0 to 2/7 or 2/12 as the first and 2/0 as the second Intermediate character are reserved for future standardization.

Multiple-byte graphic sets may also be dynamically redefinable. ESC 2/4 I 2/0 F designates such a set, where I takes the same values (2/8 to 2/11 or 2/13 to 2/15) and has the same meanings as in the preceding paragraphs. Further Intermediate characters can be used if more than 63 sets are needed to be identified.

NOTE 10

*This class of escape sequences is exceptional because the allocation of Final (and possibly Intermediate) characters is not done by the Registration Authority (see Appendix A) but by the user. It is recommended that Final characters be allocated sequentially, starting with 4/0.*

NOTE 11

*The need for this particular escape sequence as distinct from the normal three-character sequence used to represent registered sets is that it implies exact description of the shape or font of the characters.*

5.3.11 Other coding systems

ESC 2/5 F and ESC 2/5 I F (except ESC 2/5 4/0) designate and invoke a coding system different from that of this Standard, not necessarily a character code.

The escape sequence ESC 2/5 4/0 is allocated to, and recommended for, use by such other coding systems for returning to the coding system of this Standard. ESC 2/5 4/0 restores the state of the coding system to that at the time of invocation of the other coding system, that is announcers, designated and invoked control and graphic character sets. Whether or not other states, e.g. the active position, are restored is outside the scope of this Standard.

In order to indicate whether the other coding system uses the return escape sequence ESC 2/5 4/0, the designating escape sequences are divided into the following categories:

ESC 2/5 F

shall mean that the other coding system uses ESC 2/5 4/0 to return,

ESC 2/5 2/15 F

shall mean that the other coding system does not use ESC 2/5 4/0 to return (it may have an alternate means to return or none at all).

In the first case, a second Intermediate character in the range 2/1 to 2/3 and any further additional Intermediate characters can be used, if more than 63 coding systems need to be registered.

In the second case, any further Intermediate characters can be used, if more than 63 coding systems need to be registered.

Escape sequences with 2/5 as the first Intermediate character and either 2/0 or 2/4 to 2/14 as the second Intermediate character are reserved for future standardization.

The above facility provides a means for switching between ISO 2022 coding systems and other coding systems when it is not performed at a higher level (see 10.).

#### 5.3.12 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 8.

#### 5.3.13 Revision of Registered Sets

Appendix A of this Standard refers to the International Register of Coded Character Sets to be Used with Escape Sequences established under the provisions of ISO 2375. ESC 2/6 F, when used, shall immediately precede a designating escape sequence and indicate a revision of a registered set. F will specify the revision number 1 to 63 by taking values 4/0 to 7/14 respectively. Revisions are only permitted to add a character or characters to a set and shall be submitted to the Registration Authority as required by ISO 2375, pointing out that the submission is a revision of a registered set. If the revision is not upwards compatible with the previous version, then an entirely new designating escape sequence shall be allocated.

##### NOTE 12

*The combined use of the "revision number" escape sequence and of the original designating escape sequence facilitates the recognition by older devices or systems of newer versions of character sets.*

Escape sequence with 2/6 as the first Intermediate character and any further Intermediate characters are reserved for future standardization.

#### 5.1.14 Three-character escape sequences without assigned meanings

The escape sequences ESC 2/7 F and 2/12 F have not been assigned meanings and are reserved for future standardization.

### 5.3.15 Summary of assignments of Intermediate characters

Table 1 summarizes the assignments of the Intermediate characters in the escape sequences. The shaded area denotes the combinations reserved for future standardization.

SECOND INTERMEDIATE CHARACTER									
FIRST INTERMEDIATE CHARACTER		CODE POSITION							
CATEGORY		CODE POSITION		CODE POSITION		CODE POSITION		CODE POSITION	
2/0		2/1	2/4	2/5	2/8	2/9	2/10	2/11	2/13
		2/2	2/3	2/6					2/14
		2/7							2/15
2/0	Announcers								
2/1		C0 set							
2/2	Control functions	C1 set							
2/3			Single control functions						
2/4	Graphic characters			Multiple-byte sets					
2/5				Other coding systems					
2/6					Revision of Registration				
2/7									
2/8					G0 set				
2/9					G1 set				
2/10					G2 set				
2/11					G3 set				
2/12									
2/13	94 Graphic characters				G1 set				
2/14					G2 set				
2/15					G3 set				

5.4 Initial Designation and Invocation

At the beginning of any information interchange, except where interchanging parties have agreed otherwise, all designations shall be defined by use of the appropriate escape sequences, and the shift status shall be defined by the use of the appropriate locking-shift functions. Interchanging parties who agree not to use such designators are warned that they may thereby reduce their capability to interchange data subsequently.

5.5 Pictorial Representation of Code Extension in 7-bit Environment

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

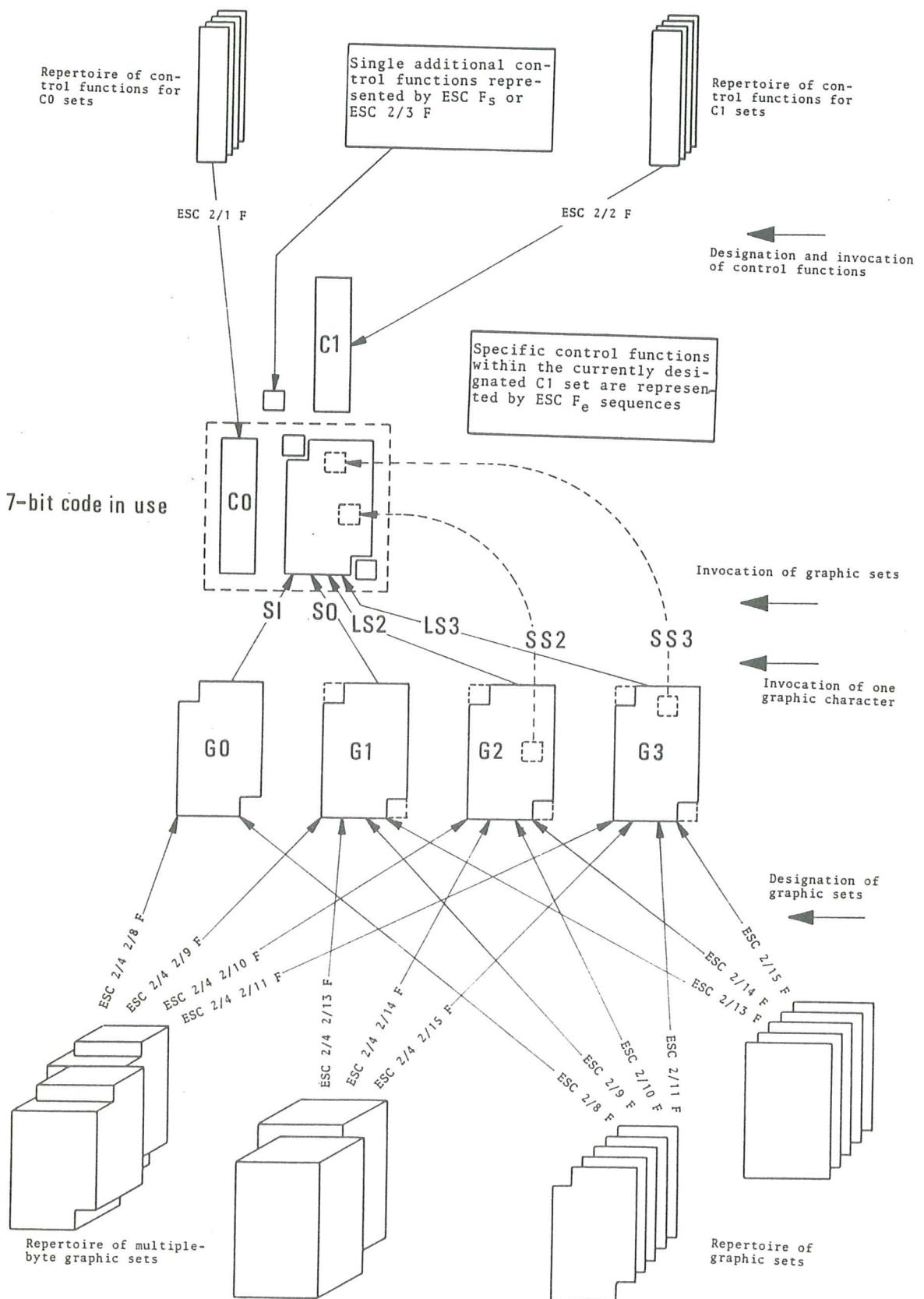


Figure 7 - Code extension in 7-bit environment  
(showing all shift functions)

## 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 set to 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 set to 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 Fig. 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 representation is ONE.

Columns 08 and 09 are provided for control characters and columns 10 to 15 for graphic characters.

The control characters in columns 08 and 09 of an 8-bit code shall not include transmission control characters.

### 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) a set of 32 additional control characters can be selected for columns 08 and 09;
- b) a set of 94 or 96 additional graphic characters can be selected for columns 10 to 15. If a set of 94 graphic characters is invoked in columns 10 to 15, positions 10/0 and 15/15 shall not be used.

There are standard techniques for identifying selections of sets of control characters and graphic characters for 8-bit codes. These techniques are described below.

## 7. 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.

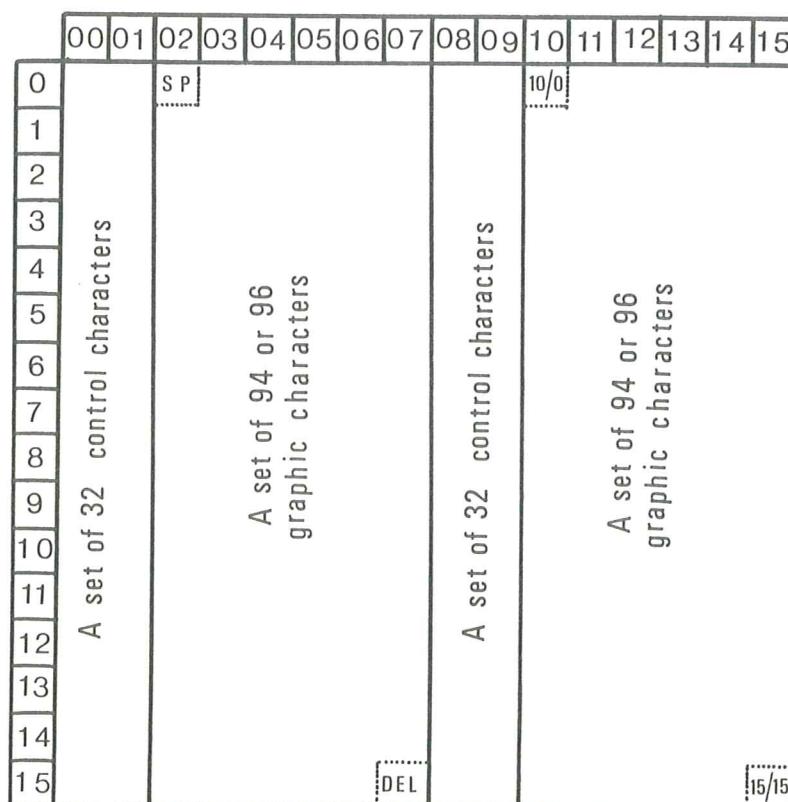
The character ESCAPE shall be used in an 8-bit code in exactly the same way as in the 7-bit code to construct escape sequences. The meanings of these sequences are not altered in an 8-bit code. All characters in columns 08 to 15 are excluded from assignment in escape sequences and any occurrences of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

### 7.1 Elements of Code Extension in an 8-Bit Environment

These elements, shown in Fig. 2, are as follows:

Set	Description	Columns occupied
C0	32 control characters	00 - 01
C1	32 control characters	08 - 09
G0	94 graphic characters	02 - 07
G1	94 or 96 graphic characters	02 - 07 or 10 - 15
G2	94 or 96 graphic characters	02 - 07 or 10 - 15
G3	94 or 96 graphic characters	02 - 07 or 10 - 15

The C0 and C1 sets shall be designated and invoked by the same escape sequences as in the 7-bit environment (see 5.3.5 and 5.3.6). The G0, G1, G2 and G3 sets shall be designated by the same escape sequences as in the 7-bit environment (see 5.3.7 to 5.3.10).



## 7.2 Extension of the Graphic Set by Means of Shift Functions

The shift functions in this Standard for use in an 8-bit environment are:

LS0, LS1, LS1R, LS2, LS2R, LS3, LS3R, SS2 and SS3.

See Appendix B for the coded representation of these functions.

### 7.2.1 Use of locking-shift functions

In an 8-bit environment there are seven locking-shift functions used exclusively for graphic set extension.

With the exception of LS0 which invokes only 94-character sets, each of the six others invokes an additional set of 94 or 96 graphic characters into columns 02 to 07 or into columns 10 to 15. The seven locking-shift functions are:

Function	Set invoked	Columns affected
LOCKING-SHIFT ZERO (LS0)	G0	02 to 07
LOCKING-SHIFT ONE (LS1)	G1	02 to 07
LOCKING-SHIFT ONE RIGHT (LS1R)	G1	10 to 15
LOCKING-SHIFT TWO (LS2)	G2	02 to 07
LOCKING-SHIFT TWO RIGHT (LS2R)	G2	10 to 15
LOCKING-SHIFT THREE (LS3)	G3	02 to 07
LOCKING-SHIFT THREE RIGHT (LS3R)	G3	10 to 15

If a particular set is already invoked the use of the corresponding shift functions has no effect.

The meanings of the following bit combinations shall not be affected by the occurrence of these locking-shift functions:

- a) those representing the control characters in columns 00, 01, 08 and 09;
- b) those included in any escape sequence;
- c) that following SS2 or SS3.

The characters SPACE and DELETE shall occur at position 02/0 and 07/15, respectively, if and only if a set of 94 graphic characters is invoked into columns 02 to 07. They shall not be assigned to any position in any set. However, characters other than SPACE and representing spaces of different sizes or usage, may be assigned to any positions in any set of graphic characters or control functions.

At the beginning of any information interchange the shift status shall be defined by use of the locking-shift functions as specified in 5.4 (see also clause 8 and Table 2).

### 7.2.2 Use of single-shift functions

Use of the single-shift functions in an 8-bit code is identical with their use in a 7-bit code (see 5.2.2). Only bit combinations from columns 02 to 07 are permitted to follow

SS2 or SS3. Except for the situation described in 9.3, bit combinations from columns 10 to 15 shall not follow SS2 or SS3. The use of a single-shift function does not affect the current status established by one or more of the locking-shift functions.

### 7.3 Code Extension by Means of Escape Sequences

When an 8-bit code has been established in accordance with 7.1, code extension shall be achieved by means of escape sequences as described below.

#### 7.3.1 Two-character escape sequences

Two-character escape sequences shall have the same structure as in the 7-bit environment (see 5.3.3.1).

ESC F<sub>S</sub> sequences represent single additional control functions with the same meaning they have in the 7-bit environment. (See note to 5.3.3.1).

The use of ESC F<sub>E</sub> sequences in an 8-bit environment is contrary to the intention of this Standard but should they occur in special circumstances (see Table 2) their meaning is the same as in the 7-bit environment.

#### 7.3.2 Three-character escape sequences

Three-character escape sequences shall have the same structure and meaning as in the 7-bit environment (see 5.3.3.2).

#### 7.3.3 Escape sequences with four or more characters

These escape sequences shall have the same structure and meaning as in the 7-bit environment (see 5.3.3.3).

#### NOTE 13

*The escape sequences to designate and invoke coding systems other than that specified in this Standard (see 5.3.11), those to designate multiple-byte graphic sets (see 5.3.9) and those to designate dynamically redefinable character sets (see 5.3.10) have the same structure and meaning as in a 7-bit environment.*

### 7.4 Sets of Graphic Characters with Multiple-byte Representation

In an 8-bit environment, as in a 7-bit environment, multiple-byte graphic character sets may be designated and invoked as G0, G1, G2 or G3 sets (see 5.3.9). A graphic character of such a multiple-byte set is represented by two or more bytes, all of which consist of bit combinations either from columns 02 to 07, or from columns 10 to 15, depending upon where the multiple-byte set has been invoked. Thus the 8th bit (b<sub>8</sub>) of each byte in such a multiple-byte representation shall be uniformly either ZERO or ONE.

#### NOTE 14

*If the 8th bit (b<sub>8</sub>) of bytes in a given multiple-byte representation is different 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.*

NOTE 15

The transformation between 7-bit and 8-bit codes (see clause 9) is not affected by the occurrence of multiple-byte graphic character sets.

7.5 Compatibility

An 8-bit code will be considered to be compatible with this Standard if columns 00 to 07 comply with 5.1.5 a) or b) and

- columns 08 and 09 contain only control characters,
- columns 10 to 15 are used for graphic characters only.

In order to provide the code extension facilities of this Standard, the character ESCAPE and the shift functions required must remain unaltered in their meanings and their coded representations (see Appendix B).

7.6 Pictorial Representation of Code Extension in 8-bit Environment

Fig. 9 summarizes, in a schematic form, the standard means of code extension available in an 8-bit environment.

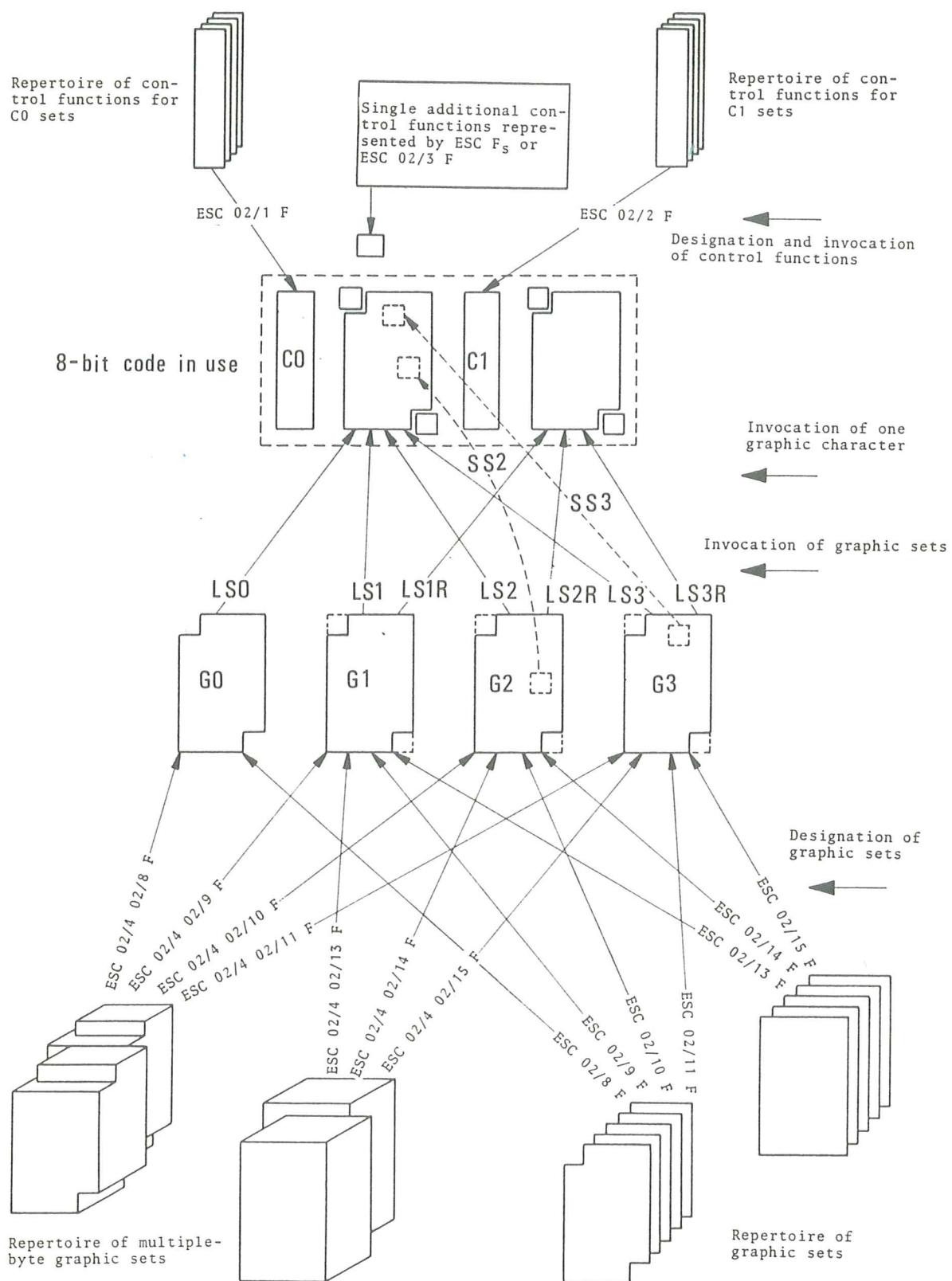


Figure 8 - Code extension in 8-bit environment  
(showing all shift facilities)

## 8. ANNOUNCEMENT OF EXTENSION FACILITIES USED

### 8.1 General

At the beginning of an information interchange, it may be required to announce the code extension facilities used in the data which follows. If such announcement is to be embedded within the character coded information, one or more of the class of three-character escape sequences ESC 2/0 F shall be used. Subject to agreement between the interchanging parties, such an announcement sequence may be omitted. The Final character of the announcing sequence indicates the facilities used for representing graphic sets and some control sets in 7-bit and 8-bit environments. The Final characters used for this purpose are listed in Table 2 together with a description of the facilities to be used and a pictorial representation where appropriate.

Escape sequences with 2/0 as the first Intermediate character and any further Intermediate characters are reserved for future standardization.

### 8.2 Restrictions

The announcers 4/1, 4/3 and 4/4 shall not be used in combination with the announcers 5/0, 5/2, 5/3, 5/4, 5/5, 5/6 and 5/7.

The announcers 4/12, 4/13 and 4/14 shall not be used together with any other announcer.

#### NOTE 16

*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 sequences are provided for those interchange situations in which it is required to differentiate in the 7-bit environment between data originating from two types of 8-bit environment, viz. those having G1 in columns 2-7 and those having G1 in columns 10-15.*

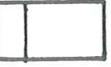
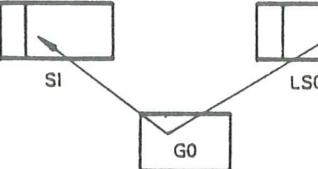
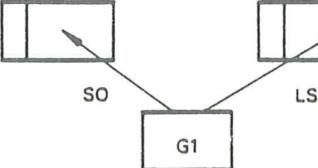
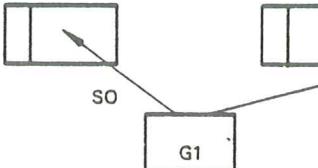
#### NOTE 17

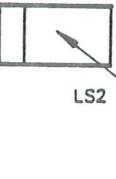
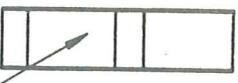
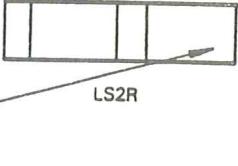
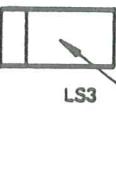
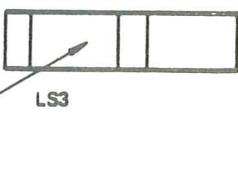
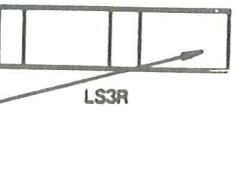
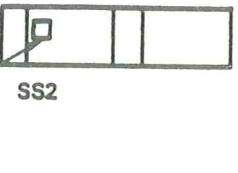
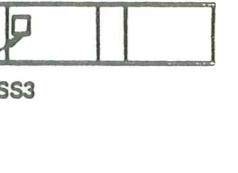
*An example of the sequences which might be used in an 8-bit environment to announce the use of G0, G1 and G3 sets with locking shifts and G2 with a single shift, is as follows:*

ESC 02/0 05/0  
ESC 02/0 05/2  
ESC 02/0 05/7  
ESC 02/0 05/10

TABLE 2

Final character	Facilities utilized	7-bit environment	8-bit environment
4/1	The G0 set shall be used. The escape sequence which designates this set also invokes it into columns 2 to 7. No locking-shift functions shall be used. In an 8-bit environment, columns 10 to 15 are not used. See 8.2.		
4/2	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 LS0 invokes G0 and LS1 invokes G1 into columns 02 to 07, while 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 02 to 07 and 10 to 15, respectively. No locking-shift functions shall be used. See 8.2.		
4/4	The G0 and G1 sets shall be used. In a 7-bit environment, SI invokes G0 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 02 to 07 and 10 to 15, respectively; no locking-shift functions shall be used. See 8.2.		
4/5	Full preservation of shift functions is maintained when transforming data between 7-bit and 8-bit environments.		See 9.4
4/6	The C1 set shall be used. In both a 7-bit and an 8-bit environment, each C1 control function shall be represented by the ESC F <sub>E</sub> sequence.		See 5.3.3.1 and 7.3.1

Final character	Facilities utilized	7-bit environment	8-bit environment
4/7	The C1 set shall be used. In a 7-bit environment, each C1 control function is represented by an ESC Fe sequence. In an 8-bit environment, each C1 control function is represented by a single bit combination from columns 08 and 09.	See 5.3.3.1	
4/8	All graphic character sets comprise 94 characters.	See 5.3.7	
4/9	The graphic character sets may comprise 94 and/or 96 characters.	See 5.3.7 and 5.3.8	
4/10	In a 7-bit or an 8-bit environment a 7-bit code is used.		
4/11	In an 8-bit environment an 8-bit code is used.		
4/12	Level 1 of ECMA-43 shall be used.		
4/13	Level 2 of ECMA-43 shall be used.		
4/14	Level 3 of ECMA-43 shall be used.		
5/0	In addition to any other category of graphic sets which may be used, the G0 set shall be used. It will be invoked by SI in a 7-bit environment and by LS0 in an 8-bit environment. See 8.2.		
5/2	In addition to any other category of graphic sets which may be used, the G1 set shall be used. It will be invoked by SO in a 7-bit environment and by LS1 in an 8-bit environment. See 8.2.		
5/3	In addition to any other category of graphic sets which may be used, the G1 set shall be used. It will be invoked by SO in a 7-bit environment and by LS1R in an 8-bit environment. See 8.2.		

Final character	Facilities utilized	7-bit environment	8-bit environment
5/4	In addition to any other category of graphic sets which may be used, the G2 set shall be used. It will be invoked by LS2 in both 7-bit and 8-bit environments. See 8.2.		
5/5	In addition to any other category of graphic sets which may be used, the G2 set shall be used. It will be invoked by LS2 in a 7-bit environment and by LS2R in an 8-bit environment. See 8.2.		
5/6	In addition to any other category of graphic sets which may be used, the G3 set shall be used. It will be invoked by LS3 in both 7-bit and 8-bit environments. See 8.2.		
5/7	In addition to any other category of graphic sets which may be used, the G3 set shall be used. It shall be invoked by LS3 in a 7-bit environment and by LS3R in an 8-bit environment. See 8.2.		
5/10	In addition to any other category of graphic sets which may be used, the G2 set shall be used. SS2 shall invoke a single character of this set in both 7-bit and 8-bit environments.		
5/11	In addition to any other category of graphic sets which may be used, the G3 set shall be used. SS3 shall invoke a single character of this set in both 7-bit and 8-bit environments.		

## 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. Identification of those facilities is achieved by use of the announcement sequences defined in clause 8.

### 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, bg is set to ZERO.

Some locking-shift functions, viz. LS1R, LS2R and LS3R have different effects in 7-bit and in 8-bit codes. When these shift functions are used in an 8-bit environment and when it is not obvious whether a 7-bit or an 8-bit code is employed an announcement sequence ESC 2/0 4/10 or ESC 2/0 4/11, respectively, shall be used to ensure unambiguous interpretation of the data.

### 9.3 Interaction of Shift Functions

If 7-bit coded data employing single-shift and locking-shift 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 5.2.2 and 7.2.2, 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 locking-shift function being inserted immediately after a single-shift character. This additional locking-shift function shall be disregarded, in so far as interpretation of the single-shift function is concerned, and the following bit combination shall be interpreted as representing a character from the G2 or the G3 set.

### 9.4 Preservation of Information on Retransformation

When transforming information originated in a 7-bit environment into an 8-bit environment, there is no difficulty in preserving the multiple use of different invokers. It is possible that in such a situation the greater facilities available in an 8-bit environment could be used to minimize the use of shift functions in that environment. This Standard does not specify a means of achieving that.

When transforming information originated in an 8-bit environment in which use has been made of the various shift functions defined in this Standard, there is likewise no difficulty in

representing the information in a 7-bit environment. However, if it is subsequently necessary to retransform that information back to an 8-bit environment, retaining the identical use of shift functions employed originally, it is necessary to keep a note of them during the 8-bit to 7-bit transformation. The announcement sequence ESC 2/0 4/5 indicates that such retention is desired, or has been preserved.

#### 10. RELATIONSHIP WITH PRESENTATION PROTOCOL

In an application which uses a Presentation Protocol or another general level of control, the beginning of a string of character-coded information is indicated according to the conventions of that protocol. The end of the string is sometimes indicated by a delimiter. This delimiter acts as an instruction to return from the coding method defined in this Standard to the coding method used in the presentation protocol. It is defined as follows.

##### CODING METHOD DELIMITER - CMD

A control function that delimits a string of data coded according to this Standard which switches to a general level of control.

Its coded representation is ESC 6/4.

##### NOTE 18

*This ESC F<sub>S</sub> sequence may be suitable for coding systems other than that of this Standard.*

The use of CMD is not mandatory if the higher level protocol defines means to delimit the string, for instance by specifying the length of the string.

#### 11. SPECIFIC MEANINGS OF ESCAPE SEQUENCES

In general 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 shall be followed in preparing and maintaining a register of escape sequences and their meanings.

Allocation of Final and, where necessary, Intermediate characters to all categories of escape sequences, except those reserved for future standardization and those for announcers, DRCS, ESC Fe and private use, is carried out by the Registration Authority in accordance with ISO 2375 (see Appendix A).

## APPENDIX A

### THE INTERNATIONAL REGISTER OF CODED CHARACTER SETS TO BE USED WITH ESCAPE SEQUENCES

---

#### A.1 The International Register

The International Register contains character sets which have been registered in accordance with procedures specified in ISO 2375. Its purpose is to identify widely used character sets and associate with each a unique escape sequence by means of which it can be designated conveniently.

The publication of this Register promotes compatibility in international information interchange and avoids duplication of effort in developing application-oriented character sets. Registration provides an identification for a character set but implies nothing about its status; it may or may not be part of an international or a national standard, or of an application-oriented standard. However, when such a standard is issued subsequent to the registration of an escape sequence, it is appropriate for the escape sequence identifying the character set to be specified in the standard.

If it is desired to register a set, application should be made as required in ISO 2375. Any character set can be a candidate for registration so long as it satisfies the technical requirements of ISO 2022 and the formal requirements of ISO 2375. Its characteristics will determine the type of escape sequence which can be allocated to it.

#### A.2 The Registration Authority

The registration procedure and the maintenance of the Register is performed by an International Registration Authority. ECMA has been designated by the Council of the International Organization for Standardization, ISO, as the Registration Authority for ISO 2375.

ECMA carries out these duties as a free service to the international data processing community. It advises applicants on the requirements to be met by applications, circulating the applications in conformance with the procedures. It allocates the escape sequence and finally registers each character set with its specific escape sequence. After each registration owners of the Register receive the corresponding additional sheets of the Register.

## APPENDIX B

### SHIFT FUNCTIONS

#### B.1 Coded Representation

Shift function		7-bit environment	8-bit environment
SHIFT-OUT	SO	0/14	-
SHIFT-IN	SI	0/15	-
LOCKING-SHIFT ZERO	LS0	-	00/15
LOCKING-SHIFT ONE	LS1	-	00/14
LOCKING-SHIFT ONE RIGHT	LS1R	-	ESC 07/14
LOCKING-SHIFT TWO	LS2	ESC 6/14	ESC 06/14
LOCKING-SHIFT TWO RIGHT	LS2R	-	ESC 07/13
LOCKING-SHIFT THREE	LS3	ESC 6/15	ESC 06/15
LOCKING-SHIFT THREE RIGHT	LS3R	-	ESC 07/12
SINGLE-SHIFT TWO	SS2	ESC 4/14	08/14
SINGLE-SHIFT THREE	SS3	ESC 4/15	08/15

If a 7-bit single-byte representation of SS2 is required, it should be bit combination 1/9.

If it is required to represent LS1R, LS2R and LS3R in a 7-bit environment, ESC 7/14, ESC 7/13 and ESC 7/12, respectively, shall be used.

#### B.2 Action of the Shift Function

Shift function	Graphic characters invoked	Side of an 8-bit code to which the graphic set is invoked
SO	G1 set	7-bit working only
SI	G0 set	7-bit working only
LS0	G0 set	Left
LS1	G1 set	Left
LS1R	G1 set	Right
LS2	G2 set	Left
LS2R	G2 set	Right
LS3	G3 set	Left
LS3R	G3 set	Right
SS2	single character	- (non-locking)
SS3	single character	- (non-locking)

ANNEX C

SUMMARY OF THE ESCAPE SEQUENCES DEFINED IN THIS STANDARD

C.1 Notation

The list below shows the escape sequences defined in this Standard. The characters in parentheses show how the number of Intermediate characters will be increased when, for each type of escape sequences, the first series of Final characters will have been allocated.

In stands for one bit combinaison in the range 2/1, 2/2, 2/3.

Im stands for zero or more bit combinaisons in the range 2/0 to 2/15.

C.2 List of Escape Sequences

ESC	2/0	F		Announcers
ESC	2/1	(In Im)	F	C0 set
ESC	2/2	(In Im)	F	C1 set
ESC	2/3	(In Im)	F	Single additional control function
ESC	2/4	2/8 2/0	(Im) F	Multiple-byte 94-char. G0 DRCS
ESC	2/4	2/8	(In Im) F	Multiple-byte 94-char. G0 set*
ESC	2/4	2/9 2/0	(Im) F	Multiple-byte 94-char. G1 DRCS
ESC	2/4	2/9	(In Im) F	Multiple-byte 94-char. G1 set
ESC	2/4	2/10 2/0	(Im) F	Multiple-byte 94-char. G2 DRCS
ESC	2/4	2/10	(In Im) F	Multiple-byte 94-char. G2 set
ESC	2/4	2/11 2/0	(Im) F	Multiple-byte 94-char. G3 DRCS
ESC	2/4	2/11	(In Im) F	Multiple-byte 94-char. G3 set
ESC	2/4	2/13 2/0	(Im) F	Multiple-byte 96-char. G1 DRCS
ESC	2/4	2/13	(In Im) F	Multiple-byte 96-char. G1 set
ESC	2/4	2/14 2/0	(Im) F	Multiple-byte 96-char. G2 DRCS
ESC	2/4	2/14	(In Im) F	Multiple-byte 96-char. G2 set
ESC	2/4	2/15 2/0	(Im) F	Multiple-byte 96-char. G3 DRCS
ESC	2/4	2/15	(In Im) F	Multiple-byte 96-char. G3 set
ESC	2/5	4/0		Standard return means
ESC	2/5	(In Im)	F	Other coding systems with standard return**
ESC	2/5	2/15 (Im)	F	Other coding systems without standard return
ESC	2/6	F		Revisions
ESC	2/8	2/0 (Im)	F	94-char. G0 DRCS
ESC	2/8	(In Im)	F	94-char. G0 set
ESC	2/9	2/0 (Im)	F	94-char. G1 DRCS
ESC	2/9	(In Im)	F	94-char. G1 set
ESC	2/10	2/0 (Im)	F	94-char. G2 DRCS
ESC	2/10	(In Im)	F	94-char. G2 set
ESC	2/11	2/0 (Im)	F	94-char. G3 DRCS
ESC	2/11	(In Im)	F	94-char. G3 set
ESC	2/13	2/0 (Im)	F	96-char. G1 DKCS
ESC	2/13	(In Im)	F	96-char. G1 set
ESC	2/14	2/0 (Im)	F	96-char. G2 DRCS
ESC	2/14	(In Im)	F	96-char. G2 set
ESC	2/15	2/0 (Im)	F	96-char. G3 DRCS
ESC	2/15	(In Im)	F	96-char. G3 set
ESC	6/4			CODING METHOD DELIMITER (CMD)

\* Three escape sequences, namely ESC 2/4 4/0, ESC 2/4 4/1 and ESC 2/4 4/2, are also designating multiple-byte 94-characters G1 sets. See clause 5.3.9 of the Standard.

\*\* Except ESC 2/5 4/0 representing the standard return means.

## APPENDIX D

### MAIN DIFFERENCES BETWEEN THE 3rd EDITION (1982) AND THE PRESENT 4th EDITION OF THIS STANDARD

---

- D.1 The concept of graphic character set has been extended for G1, G2 and G3 sets to include 96 in addition to 94 graphic characters. This extension also applies to multiple-byte graphic character sets. As a consequence positions 10/0 and 15/15 are now available for use with 96-character sets.
- D.2 94 and 96-character sets are designated by different categories of escape sequences, but are invoked by the same shift functions.
- D.3 The concept "complete code" has been replaced by "coding system other than that of ECMA-35". The use of ESC 2/5 F sequences for designating and invoking such coding systems has been clarified. A standard means of returning from such coding systems to ECMA-35 has been provided for those cases where it is applicable.
- D.4 Two new announcers have been specified to distinguish between 7-bit codes and 8-bit codes in an 8-bit environment.
- D.5 A control function, CODING METHOD DELIMITER (CMD), has been introduced (see 10). It delimits strings of character-coded data and switches to a general level of control, such as a presentation protocol.
- D.6 A new Appendix A on registration has been introduced.
- D.7 A new Appendix C summarizes the escape sequences specified in this Standard.
- D.8 Various editorial improvements have been made.

