ECMA
EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-35

CODE EXTENSION TECHNIQUES

3rd Edition – January 1982

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

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 techniques 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 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 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 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 character 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 codes, remaining in a 7-bit environment, and making use of code extension techniques;
- the structure of a family of 8-bit codes, remaining compatible with the 7-bit structure;
- 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.

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
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:

Bit combination

An ordered set of bits used for representation of characters.

Byte

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

Character

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

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.

Code extension

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

Code Table

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

Control Character

A control function the coded representation of which consists of a single bit combination.

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.

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.

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.

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.

Final character

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

Graphic character

A character, other than a control function, that has a visual representation normally handwritten, printed or displayed and that has a coded representation consisting of one or more bit combinations.

Intermediate character

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

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.

Position

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

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.

Version of the 7-bit code

A version of the 7-bit code is a code table in which all options left open by Table 1 of ECMA-6 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.

Version of the 8-bit code

A version of the 8-bit code is a code table in which all options left open by Table 1 of ECMA-43 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 and row numbers 0 to 9 of an 8-bit code. 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 02/00.

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 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.

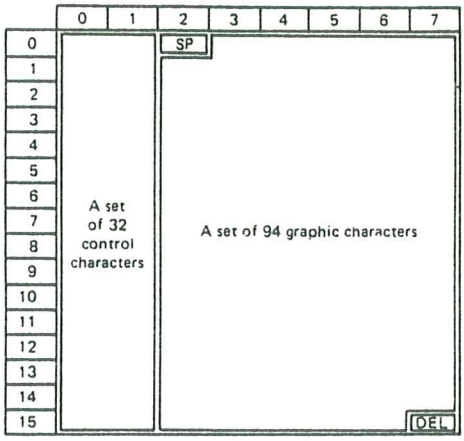


FIGURE 1

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 shall be regarded as constituting a new code outside the provision of this Standard.

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:

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

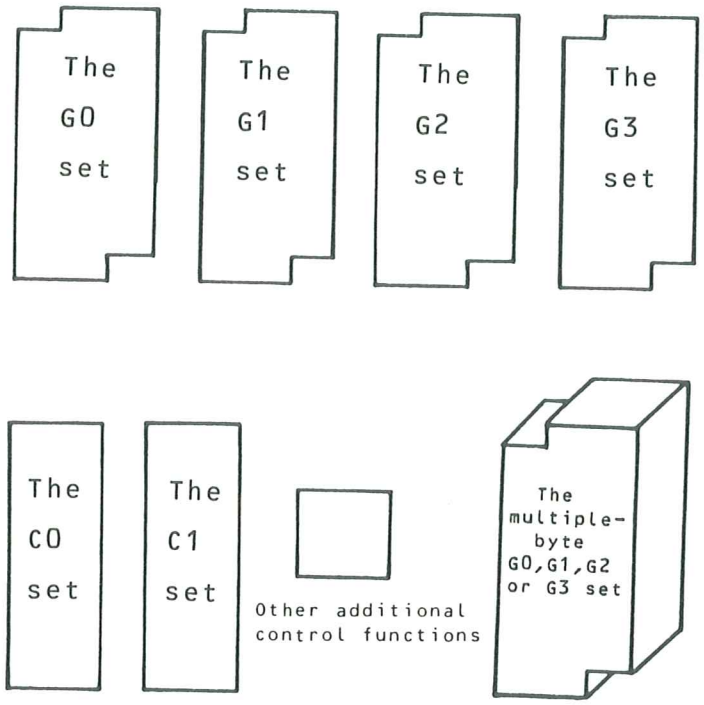


FIGURE 2

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 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,
  - 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).

- 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 functions 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 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 functions

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 coding of these functions.

Locking shift functions LS1R, LS2R, LS3R (see 7.2.2) are used in a 7-bit environment only when it is necessary to preserve their use for transformation between 7 and 8-bit environments (see 9.5).

5.1.8 Combination of graphic characters

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

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

ECMA-6 allows for the first of these two methods to represent accented characters. Sponsors of graphic character sets applying for registration under the provision of ISO 2375 are expected to identify any graphic characters in the set intended to be used in combination with other graphic

characters and to state the method by which this combination is to be achieved, including a clear indication of any characters in the set that are non-spacing. Any restriction on the use of BACKSPACE with any characters or the combination of spacing and non-spacing characters shall be clearly specified.

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

For use in a 7-bit environment the shift functions specified in this Standard are: SI, SO, LS2, LS3, SS2 and SS3.

5.2.1 Use of locking shift functions

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

The shift functions SO, LS2 or LS3 each invoke an additional set of 94 graphic characters: G1, G2 and G3 respectively. Such a 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 previously invoked set.

The shift function 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 the locking shift functions:

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

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

c) those included in any escape sequence;

d) the one following SS2 or SS3.

The character SPACE occurs at position 2/0; it shall not be assigned to any position in the additional graphic set. However, characters other than SPACE and representing a space of different size or usage may be assigned to other positions in any set of graphic characters.

At the beginning of any information interchange the shift status shall be defined by use of one of the locking shift functions (but see 8 and Fig. 10). If a particular set has already been invoked, use of the corresponding shift function has no effect.

5.2.2 Use of single-shift functions

The single-shift functions SS2 and SS3 are used exclusively for extension of the graphic character 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 (but see 5.3.8) 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 (see 9.4). 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 character sets

Some applications require no more than three additional sets of 94 graphic 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. 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 character sets

If there is a requirement for more than three additional sets of graphic characters or for more than one graphic character 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. Each subsequent use of a shift function invokes 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 invokes the graphic characters of the set last designated for use by that shift function but does not affect the identity of any sets currently designated. A 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 character set is designated by an escape sequence the current shift status remains unaltered.

When a graphic character set is designated by an escape sequence and if that class of graphic character set is currently invoked, then the new set is also invoked.

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

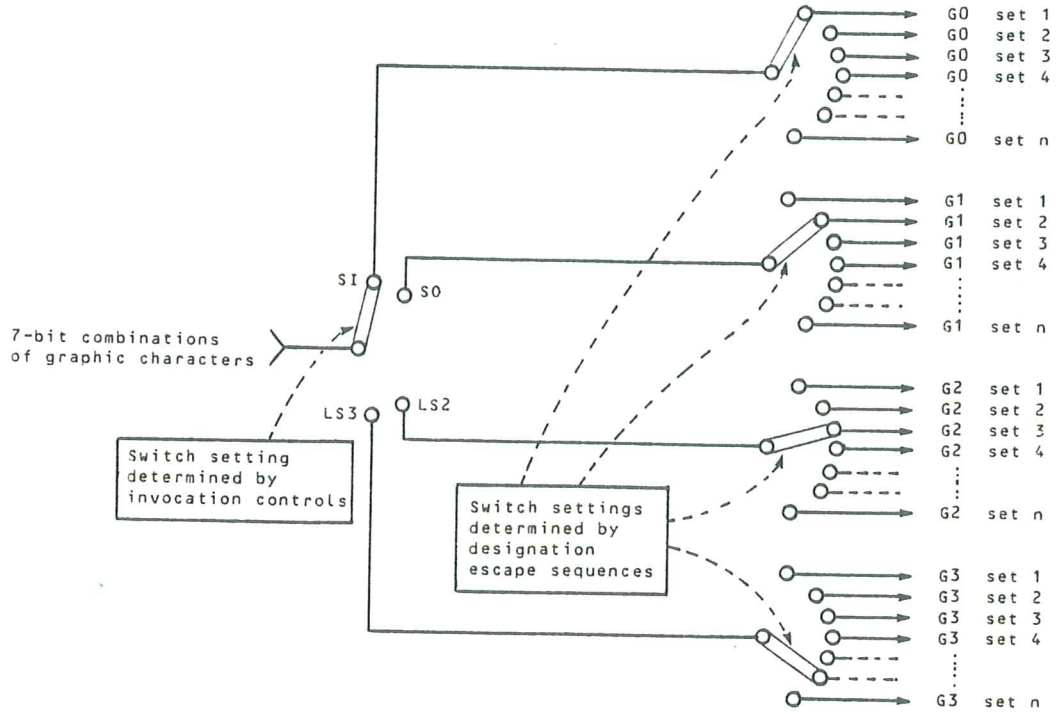


Figure 3

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, (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 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 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 bit combinations representing Intermediate characters.

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

NOTE 2

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

Intermediate characters are represented by the bit-combinations of column 2.

Final characters are represented by the bit-combinations of columns 3 to 7 excluding position 7/15.

NOTE 3

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

NOTE 4

In this Standard, any of these 79 Final characters is denoted by the symbol: F.

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.

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 are reserved for private use subject to the categorization outlined below.

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 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 shall be of 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.

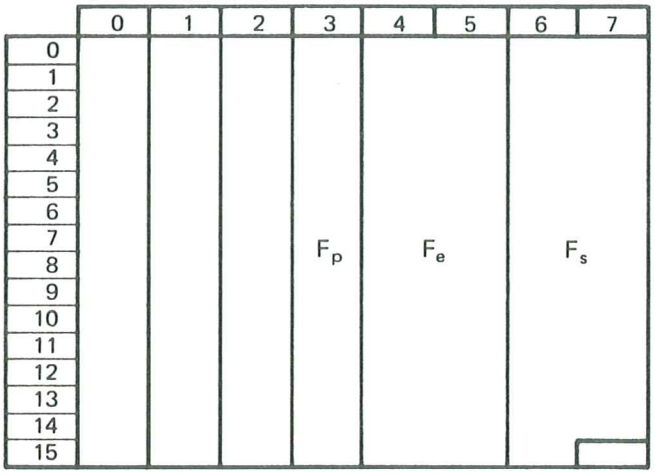


FIGURE 4

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.

NOTE 6

ESC Fs sequences are registered in the ISO International Register of Character Sets to be used with Escape Sequences, which is maintained by the Registration Authority. When any candidates for ESC Fs sequences have been approved by ISO/TC97/SC2 for standardization, the coding for the Final character, F_s will be assigned by the Registration Authority.

An ESC Fe sequence represents, depending on the Final character used, a single 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 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 5.3.12 (see Table 1 on page 19).

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

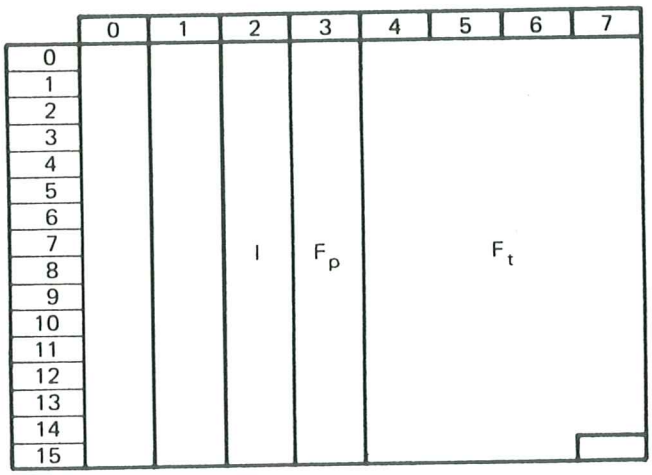


FIGURE 5

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

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

5.3.4 Single additional control 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:

- 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 (see Note 8).

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. LS2 invokes the designated set and SS2 invokes one character from the designated set.

ESC 2/11 F and ESC 2/15 F designate sets of 94 graphic characters which will be used as the G3 set. LS3 invokes the designated set and SS3 invokes one character from the designated set.

NOTE 7

There is a single repertoire of sets of 94 graphic characters, 126 of which may be designated by three-character escape sequences. The first 63 of these are identified by the Final character and one of the Intermediate characters 2/8 to 2/11. The Intermediate character designates the set as a G0, G1, G2 or G3 set respectively. A further 63 sets are identified by the Final character and one of the Intermediate characters 2/12 to 2/15, similarly used.

NOTE 8

Concerning paragraphs 5.3.6 and 5.3.7, 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 character sets are not registered as either G0, G1, G2 or G3 sets but as all four. They may be used in any of these four ways by use of the appropriate Intermediate character.

5.3.8 Sets of graphic characters with multiple-byte representation

ESC 2/4 F and ESC 2/4 I F designate 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.

ESC 2/4 F and ESC 2/4 2/12 F designate the multiple-byte graphic character sets which will be used as the G0 set. The designated set is invoked by SI.

ESC 2/4 2/9 F and ESC 2/4 2/13 F designate the multiple-byte graphic character sets which will be used as the G1 set. The designated set is invoked by SO.

ESC 2/4 2/10 F and ESC 2/4 2/14 F designate the multiple-byte graphic character sets which will be used as the G2 set. LS2 invokes the designated set and SS2 invokes one character from the designated set.

ESC 2/4 2/11 F and ESC 2/4 2/15 F designate the multiple-byte graphic character sets which will be used as the G3 set. LS3 invokes the designated set and SS3 invokes one character from the designated set.

Within such a set, each graphic character is represented by the same number of bytes. See Figure 6 below.

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

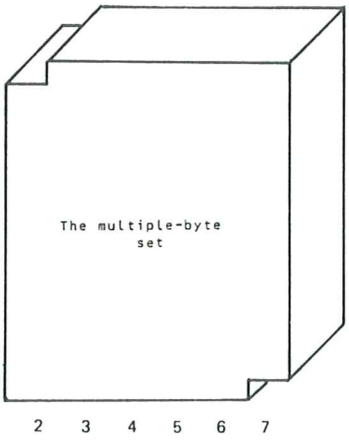


FIGURE 6

5.3.9 Dynamically Redefinable Character Sets (DRCS)

A DRCS is a set of graphic characters the exact shape of which is specified and transmitted at the time of use. Such characters may be alphabetic, special symbols or picture element symbols. Once loaded, a DRCS is regarded as a member of a library that can be designated by appropriate escape sequences as G0, G1, G2 or G3 sets.

ESC I 2/0 F designates such a set, where I shall be a bit combination from 2/8 to 2/15 to indicate whether the set is to be used as a G0, G1, G2 or G3 set, in the same way as defined in paragraph 5.3.7.

126 i.e. 2 x 63 sets may be identified by means of such four-character sequences. This should be enough for most requirements but a third or more Intermediate characters can be inserted between the 2/0 character and the Final character if more sets are needed.

Sequences with 2/0 as the second Intermediate character and with 2/0 to 2/7 as the first Intermediate characters 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 shall be a bit combination from 2/8 to 2/15 to indicate whether the set is to be used as a G0, G2 or G3 set, in the same way as defined in paragraph 5.3.7.

NOTE 9

This class of escape sequence is exceptional because the allocation of final (and possibly Intermediate characters) is not done by the Registration Authority (see ISO 2375) but by the user. It is recommended that Final characters be allocated sequentially, starting with 4/0.

NOTE 10

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.

5.3.10 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 5.3.13.

5.3.11 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.

5.3.12 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.

5.3.13 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 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 Intermediate character 2/1 to 2/15 and any additional Intermediate characters will be associated with the Final character to permit additional entities within the class defined by the first Intermediate character.

  The second Intermediate character 2/0 is reserved for the designation of DRCS sets - see 5.3.9.

  In an escape sequence having 2/4 has the first Intermediate character, special meaning is given to the second Intermediate character as specified in 5.3.8.

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

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

5.4 Omission of Escape Sequences

At the beginning of any information interchange, except where interchanging parties have agreed otherwise, all designations will be defined by use of the appropriate escape sequences, and the shift status will be defined by 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 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.

TABLE 1

Code Position	Bits of Intermediate characters $b_7 b_6 \ldots b_1$	Category	Grouping	See clause
2/0	0 1 0 0 0 0 0	Announcers		5.3.11
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	GRAPHIC CHARACTERS	Multiple-byte sets	5.3.8
2/5	0 1 0 0 1 0 1	Complete codes		5.3.10
2/6	0 1 0 0 1 1 0	Reserved for future standardization		5.3.12
2/7	0 1 0 0 1 1 1			
2/8	0 1 0 1 0 0 0	GRAPHIC CHARACTERS *(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 11

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.

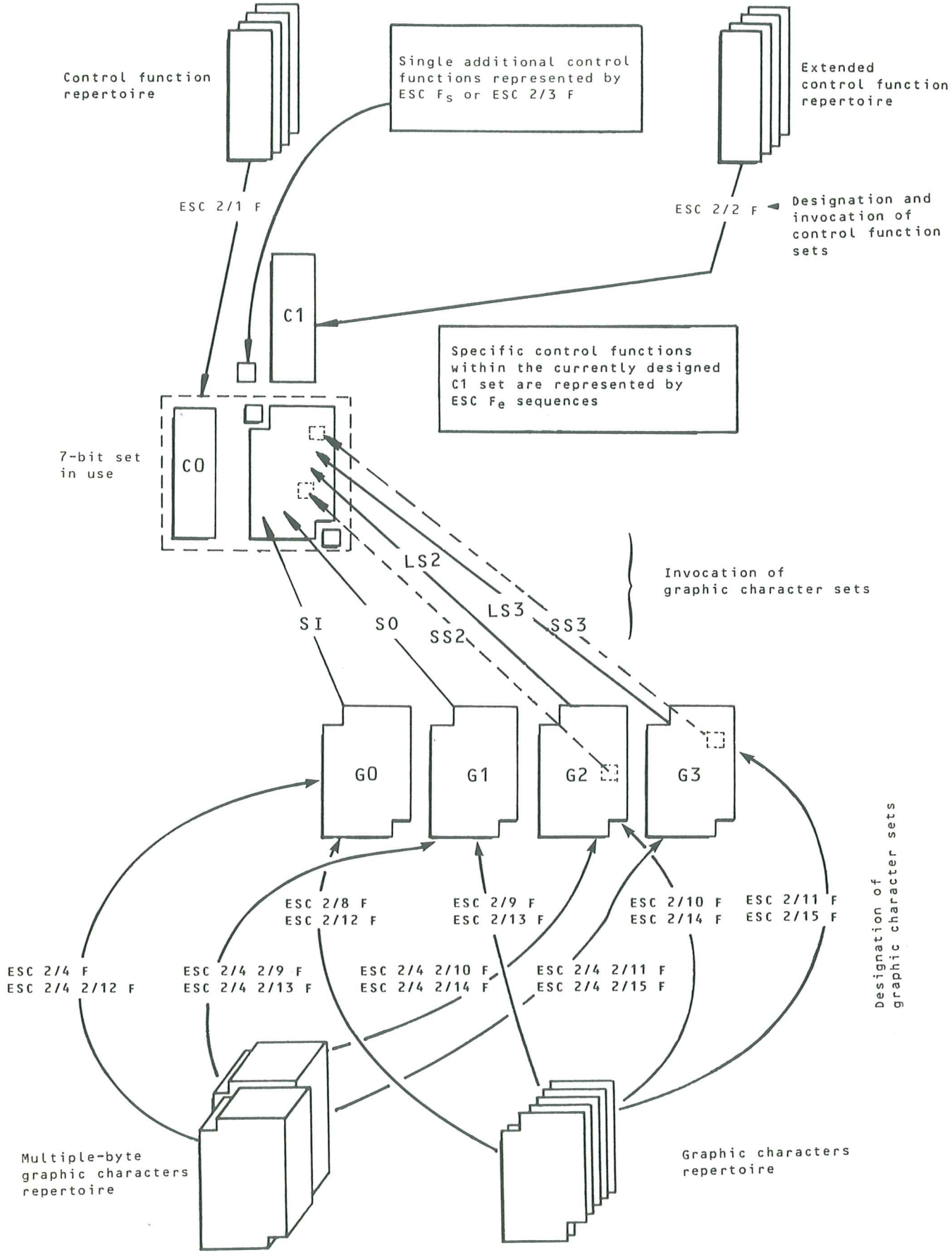

Control function repertoire
Single additional control functions represented by ESC F_S or ESC 2/3 F
Extended control function repertoire
ESC 2/1 F
ESC 2/2 F
Designation and invocation of control function sets
C1
Specific control functions within the currently designed C1 set are represented by ESC F_e sequences
7-bit set in use
C0
LS2
LS3
SS3
SS2
SI
SO
Invocation of graphic character sets
G0
G1
G2
G3
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 character sets
ESC 2/4 F
ESC 2/4 2/12 F
ESC 2/4 2/9 F
ESC 2/4 2/13 F
ESC 2/4 2/10 F
ESC 2/4 2/14 F
ESC 2/4 2/11 F
ESC 2/4 2/15 F
Multiple-byte graphic characters repertoire
Graphic characters repertoire

FIGURE 7

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 00 to 15 contains 256 code positions (see Figure 8).

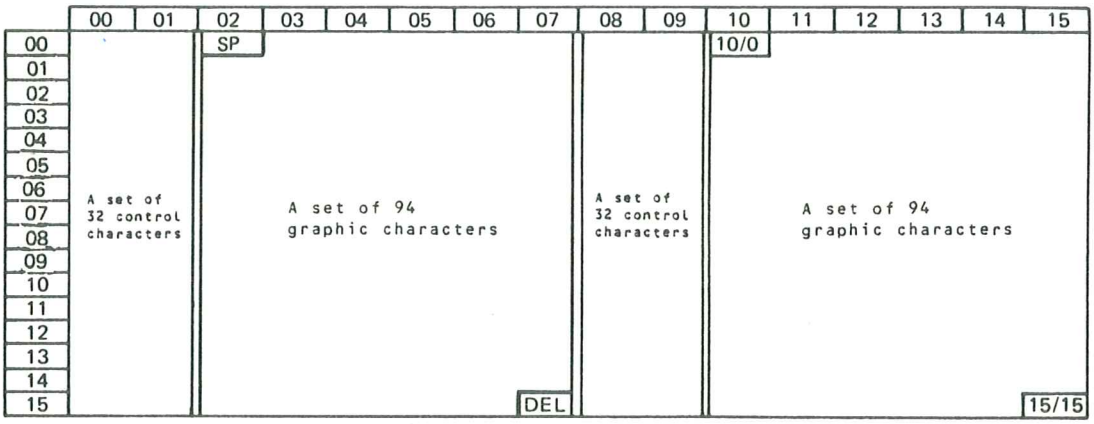


FIGURE 8

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 10/00 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 10/00 and 15/15).

There are standard techniques for identifying selections of sets of control functions and graphic characters 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.

Due to this compatibility not all extension facilities described for use in a 7-bit environment have been repeated in this section, see for example 5.3.9, 5.3.10, 5.3.11 or 5.4.

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 The Elements of Code Extension in an 8-bit Environment

These elements, shown in Figure 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 graphic characters	02-07 or 10-15
G2	94 graphic characters	02-07 or 10-15
G3	94 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).

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.

Coding for them is shown in Appendix B.

7.2.1 Use of single-shift functions

Use of the single-shift functions in an 8-bit code is identical to their use in a 7-bit code (see 5.2.2). The bit combination following SS2 or SS3 one from columns 02 to 07 except positions 02/00 and 07/15. For the use of single-shift functions with multiple-byte sets, see 5.3.8. All bit combinations of columns 08 to 15 shall not follow SS2 or SS3 so as to maintain compatibility between 7-bit and 8-bit environments (see 9.4). The use of a single-shift function does not affect the current status established by one or more of the locking-shift functions.

7.2.2 Use of the locking shift functions

In an 8-bit environment there are seven locking shift functions used exclusively for graphic set extension. Each invokes an additional set of 94 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 - 07
LOCKING SHIFT ONE	(LS1)	G1	02 - 07
LOCKING SHIFT ONE RIGHT	(LS1R)	G1	10 - 15
LOCKING SHIFT TWO	(LS2)	G2	02 - 07
LOCKING SHIFT TWO RIGHT	(LS2R)	G2	10 - 15
LOCKING SHIFT THREE	(LS3)	G3	02 - 07
LOCKING SHIFT THREE RIGHT	(LS3R)	G3	10 - 15

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

- those corresponding to the control characters in columns 00 and 01, 08 and 09 and position 07/15;
- the one corresponding to the character SPACE in position 02/00;
- those corresponding to positions 10/00 and 15/15;
- those included in any escape sequence;
- the one following SS2 or SS3.

The character SPACE occurs at position 02/00; it shall not be assigned to any position in the additional graphic sets. However characters other than SPACE and representing spaces of different size or usage, may be assigned to other positions in any set of graphic characters.

At the beginning of any information interchange the shift status shall be defined by use of the locking shift functions (but see 8 and Figure 10). If a particular set has already been invoked, use of the corresponding shift function has no effect.

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_S sequences represent single additional control functions with the same meaning they have in the 7-bit environment (see Note 6).

The use of ESC F_e sequences in an 8-bit environment is contrary to the intention of this Standard but should they occur in special circumstances (see Fig. 10) 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 having four or more characters

These escape sequences shall have the same structure and meaning as in the 7-bit environment (see 5.3.13).

NOTE 12

The escape sequences to designate and invoke complete codes (see 5.3.10) those to designate multiple-byte graphic sets (see 5.3.8) and those to designate dynamically redefinable character sets (see 5.3.9) 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 5.3.8). A graphic character of such a multiple-byte set is represented by two or more bytes, all of which consist of a bit combination either in columns 02 to 07, apart from position 02/00 and 07/15 or in columns 10 to 15, apart from position 10/0 and 15/15. Thus the 8th bit (b8) of each byte in a given multiple-byte set must be uniformly the same (either ZERO or ONE).

NOTE 13

If the 8th bit (b8) 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 14

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

7.5 Compatibility

An 8-bit code will be considered as compatible with this Standard if columns 00 to 07 comply with paragraph 5.1.5 of this Standard and

- columns 08 and 09 contain only control characters,
- columns 10 and 15 (excluding positions 10/00 and 15/15) are used for graphic characters only (see Figure 8 and ECMA-43).

In order to provide the code extension facilities of this Standard, the shift characters required must remain unaltered in their meanings and their positions in the code table (see Appendix B).

7.6 Pictorial Representation

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

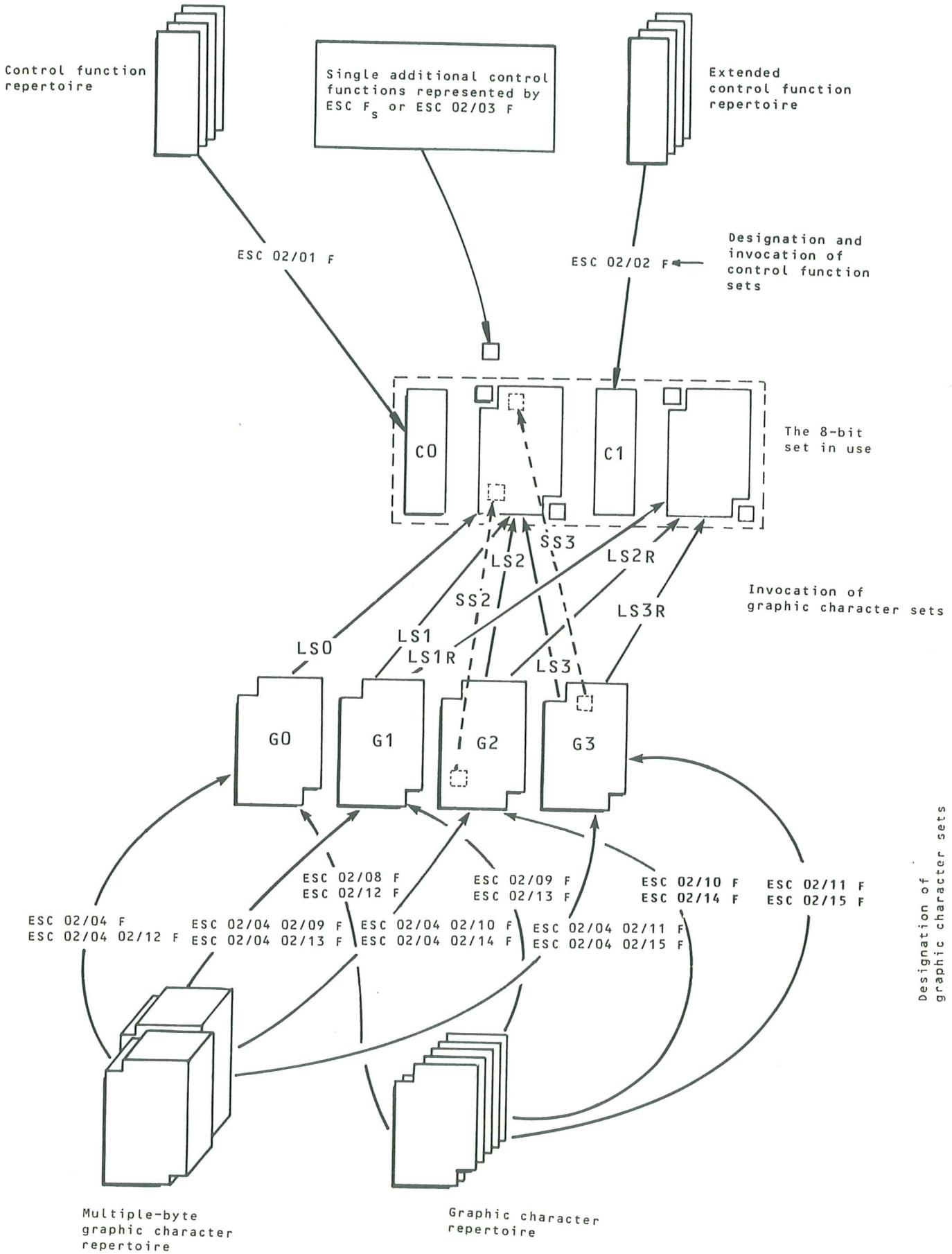

Control function repertoire
Single additional control functions represented by ESC F_s or ESC 02/03 F
Extended control function repertoire
ESC 02/01 F
ESC 02/02 F
Designation and invocation of control function sets
C0
C1
The 8-bit set in use
LS2
SS3
LS2R
SS2
LS3R
Invocation of graphic character sets
LS0
LS1
LS1R
LS3
G0
G1
G2
G3
ESC 02/08 F
ESC 02/12 F
ESC 02/09 F
ESC 02/13 F
ESC 02/10 F
ESC 02/14 F
ESC 02/11 F
ESC 02/15 F
ESC 02/04 F
ESC 02/04 02/12 F
ESC 02/04 02/09 F
ESC 02/04 02/13 F
ESC 02/04 02/10 F
ESC 02/04 02/14 F
ESC 02/04 02/11 F
ESC 02/04 02/15 F
Designation of graphic character sets
Multiple-byte graphic character repertoire
Graphic character repertoire

FIGURE 9

8. ANNOUNCEMENT OF EXTENSION FACILITIES USED

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 (or ESC 02/00 F in an 8-bit environment) 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 Figure 10 together with a description of the facilities to be used and a pictorial representation where appropriate.

NOTE 15

For the sake of simplicity the Final characters are identified in Fig. 10 by their positions in a 7-bit environment. In an 8-bit environment the same positions, in 8-bit notation, apply.

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 7-bit and 8-bit originated data.

NOTE 17

An example of the sequences which might be used in a 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 02/00 05/00
ESC 02/00 05/02
ESC 02/00 05/07
ESC 02/00 05/10

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. Identification of those facilities is achieved by use of the Announcement Sequence defined in 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, b_8 is set to ZERO.

9.3 Representation of Positions 10/00 and 15/15 in a 7-Bit Environment

No meaning is assigned to positions 10/00 and 15/15 in this Standard. If there is a requirement to represent these posi-

tions in a 7-bit environment, a private escape sequence shall be used. If a "complete code" (5.3.10) allocates meanings to 10/00 and 15/15 it may require special attention.

9.4 Interaction of Shift Characters

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 7.2.1 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.

9.5 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 is allocated to indicating that such retention is desired, or has been preserved.

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 shall 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 Standard shall be allocated by the ISO Coding Committee (ISO/TC97/SC2) using the same procedures mentioned in 10.1.

Figure 10 - Announcement of Extension Facilities (see Note 15)

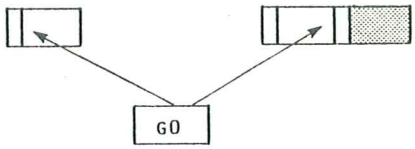
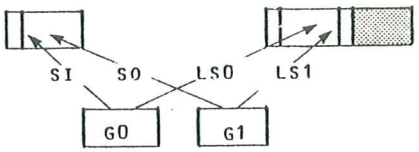
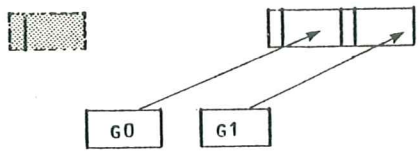
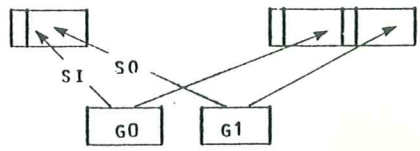
Final Character	Facilities utilized	7-bit environment / 8-bit environment
4/1	The G0 set only shall be used. The escape sequence which designates this set also invokes it into columns 2 to 7. No shift functions shall be used. In an 8-bit environment, columns 10 to 15 are not used.	
4/2	The G0 and G1 sets shall be used. In 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 encironment only. The designating sequences also invoke the G0 and G1 sets into columns 02 to 07 and 10 to 15 respectively. No shift functions shall 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 02 to 07 and 10 to 15 respectively; no shift functions shall be used.	
4/5	Full preservation of shift functions is maintained when transforming data between 7-bit and 8-bit environments.	see 9.5
4/6	The C1 set shall be used. In both a 7-bit and an 8-bit environment, each C1 control shall be represented by an ESC F_e sequence.	see 5.3.3.1 and 7.3.1

Figure 10 (cont'd)

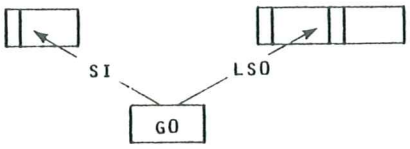
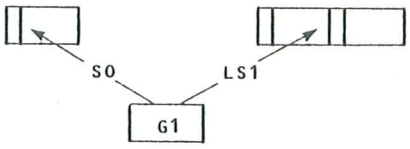
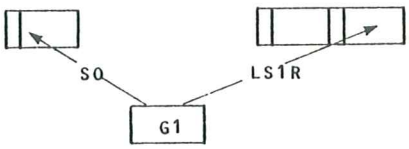
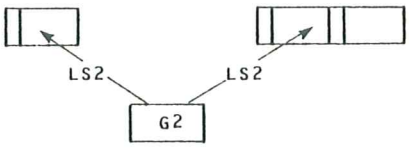
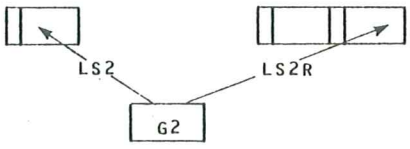
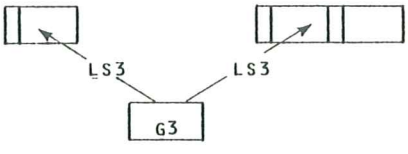
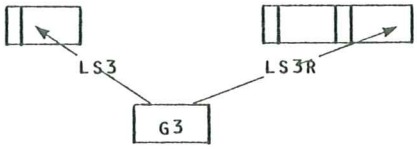
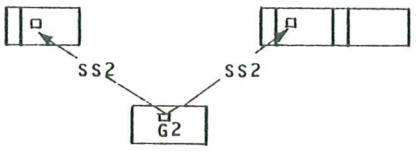
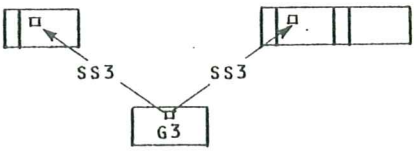
Final Character	Facilities utilized	7-bit environment / 8-bit environment
4/7	The C1 set shall be used. In a 7-bit environment, each C1 control is represented by an ESC F_e sequence. In an 8-bit environment each C1 control is represented by a single control character from columns 08 and 09.	see 5.3.3.1
5/0	In addition to any other category of graphic sets which may be used, the G0 set will be used. It will be invoked by SI in a 7-bit environment and by LS0 in an 8-bit environment.	
5/2	In addition to any other category of graphic sets which may be used, the G1 set will be used. It will be invoked by SO in a 7-bit environment and by LS1 in an 8-bit environment.	
5/3	In addition to any other category of graphic sets which may be used, the G1 set will be used. It will be invoked by SO in a 7-bit environment and by LS1R in an 8-bit environment.	
5/4	In addition to any other category of graphic sets which may be used, the G2 set will be used. It will be invoked by LS2 in both 7-bit and 8-bit environments.	
5/5	In addition to any other category of graphic sets which may be used, the G2 set will be used. It will be invoked by LS2 in a 7-bit environment and by LS2R in an 8-bit environment.	
5/6	In addition to any other category of graphic sets which may be used, the G3 set will be used. It will be invoked by LS3 in both 7-bit and 8-bit environments.	

Figure 10 (cont'd)

Final Character	Facilities utilized	7-bit environment / 8-bit environment
5/7	In addition to any other category of graphic sets which may be used, the G3 set will be used. It will be invoked by LS3 in a 7-bit environment and by LS3R in an 8-bit environment.	
5/10	In addition to any other category of graphic sets which may be used, the G2 set will be used. It will be invoked by SS2 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 will be used. It will be invoked by SS3 in both 7-bit and 8-bit environments.	

APPENDIX A

MAIN DIFFERENCES BETWEEN THE 2nd AND THE 3rd EDITION OF THIS STANDARD

A.1 G2 and G3 Sets

The concept of two additional graphic sets of 94 characters each was introduced. The necessary form of escape sequences for designating and invoking G2 and G3 graphic sets were added.

To invoke characters of these sets four locking shift functions and two single-shift functions were introduced.

With the use of G2 and G3 sets it became necessary to define locking shifts for G0 and G1 in an 8-bit environment.

These changes permit the designation of 376 (i.e. 4 x 94) graphic characters concurrently.

With the introduction of these new locking shift functions, further announcement sequences were added.

A.2 Designation of Complete 7-bit and 8-bit Codes

Clause 5.3.10 was revised so that the 7-bit and 8-bit codes designated and invoked by ESC 2/5 F sequences are no longer limited to codes requiring special attention. 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.

A.3 Multiple-byte Sets

Multiple-byte sets were permitted to be G1, G2 and G3 sets as well as G0 sets as formerly.

A.4 Designation of DRCS Sets

A new class of escape sequence was introduced to identify a set of graphic characters which is not specified until time of use (see 5.3.9).

A.5 Compatibility

All facilities specified in the 2nd Edition of ECMA-35 have been retained in the present 3rd Edition, thus maintaining upward compatibility.

APPENDIX B

SHIFT CONTROL FUNCTIONS

B.1 Coding

		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 F_S
LOCKING SHIFT TWO	LS2	ESC F_S	ESC F_S
LOCKING SHIFT TWO RIGHT	LS2R	-	ESC F_S
LOCKING SHIFT THREE	LS3	ESC F_S	ESC F_S
LOCKING SHIFT THREE RIGHT	LS3R	-	ESC F_S
SINGLE SHIFT TWO	SS2	ESC 4/14*	08/14
SINGLE SHIFT THREE	SS3	ESC 4/15	08/15

For each locking shift represented by an ESC F_S sequence, the same Final character will apply for 7- and 8-bit usage.

**If a single-byte representation of SS2 is required in 7-bit environment, it should be coded as 1/9.*

B.2 Shift Function Actions

Shift Function	Graphic Set which it invokes	Side of an 8-bit code to which the graphic set is invoked
S0	G1	7-bit working only
S1	G0	7-bit working only
LS0	G0	left
LS1	G1	left
LS1R	G1	right
LS2	G2	left
LS2R	G2	right
LS3	G3	left
LS3R	G3	right
SS2	G2	left (non-locking)
SS3	G3	left (non-locking)