ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-35

FOR

EXTENSION OF THE 7-BIT CODED CHARACTER SET

December 1971

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-35

FOR

EXTENSION OF THE 7-BIT CODED CHARACTER SET

December 1971

BRIEF HISTORY

In July 1970 the ECMA Committee on Codes (TC1) issued the 3rd Edition of their Standard ECMA-6 for the 7-bit Coded Character Set. In parallel with this revision, work on extension techniques of this character set was in progress, a first draft was prepared and contributed to ISO/TC97/SC2, where it was discussed together with other proposals. At their October 1971 meeting in Tokyo, SC2 finally agreed on a Draft International Standard for Extension Techniques for the 7-bit Coded Character Set, which is technically identical to the present Standard ECMA-35, passed at the General Assembly of Dec. 2-3, 1971.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

1. INTRODUCTION

In the 7-bit code table (Standard ECMA-6) characters have been allocated to all 128 bit combinations.

If more characters are required, they can be obtained in a number of standard ways:

- remaining in a 7-bit environment and making use of code extension techniques;
- increasing the number of bits to 8;
- increasing the number of bits to 8 and applying code extension techniques.

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

- minimizing the risk of conflict between systems required to interoperate,
- permitting advance provision for code extension in the design of equipment,
- providing standardized methods of calling up internationally agreed sets of characters,
- allowing the unambiguous interchange of data between 7-bit and 8-bit environments, etc.

2. SCOPE

2.1 This Standard ECMA-35 specifies methods of extending the 7-bit code, either remaining in a 7-bit environment or increasing to an 8-bit environment. The description of procedures is contained in interrelated sections dealing 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.

2.2 While the 7-bit code of ECMA-6 is the standard code for general information interchange, an 8-bit code as described in this standard shall be used for information interchange within an 8-bit environment. Such an 8-bit code is also recommended for file storage.

2.3 It is not the intention of this Standard to suggest that all instances of its application accommodate all of its provisions. However, it is intended that, when code extension procedures are used, the applicable part of this Standard are to be followed.

2.4 Code extension techniques are classified and some classes given a structure in this Standard. Specific assignments of bit patterns associated with the designation of the classes are to be made in accordance with ISO Standard 2375.

2.5 Code extension techniques are designed to be used for data 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 formated for fixed record processing may have undesirable results.

3. DEFINITIONS AND NOTATIONS

3.1 Definitions

In this Standard, the following terms are used with the following special meanings:

3.1.1 To designate: to nominate a set of characters which may be brought into effect subsequently.

3.1.2 To invoke: to make the nominated set available for use; it is one stage nearer to the ultimate use of a single character of the set but it is still only a preparatory stage to that end.

3.1.3 To represent:

(i) to use a code combination with the meaning defined in a set that has been designated and invoked,

(ii) to use an escape sequence with the meaning of an additional control character.

3.2 Notations

In this Standard the following notations are used:

The bits of a 7-bit combination:		b_7	b_6	b_5	b_4	b_3	b_2	b_1
The bits of an 8-bit combination:	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1
Bit weight for column & 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 a_8-a_5 (or b_7-b_5) and the row number is the decimal equivalent of bits a_4-a_1 (or b_4-b_1), giving to these bits, the weights shown above. In representing the decimal equivalents, the convention is to append a leading zero to the column number for 8-bit columns 00 to 09. As an example the position of the character 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.

4. EXTENSION OF THE 7-BIT CODE REMAINING IN A 7-BIT ENVIRONMENT

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

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

This Standard does not describe the use of the character DATA LINK ESCAPE which is reserved for the provision of additional transmission controls; the use of this character is specified in other ECMA Standards (see ECMA-16, ECMA-24).

4.1 Choosing a Basic Set

Within a country, an organization or a system, there will be a need to choose a basic set of characters as the normal implementation of the 128 positions of the code table.

There are three types of basic sets, with various levels of compatibility with ECMA-6.

(1) an international version permitted by ECMA-6,

(2) a national version permitted by ECMA-6,

(3) a "compatible" variant.

NOTE: A "compatible" variant is defined to be a set which is compatible with ECMA-6 in as much as:

1) columns 0 and 1 contain only controls;
2) the ten TC's, NUL, SO, SI, CAN, SUB, ESC and DEL remain unaltered in their meanings and positions in the code table;
3) columns 2 to 7 are used for graphics only (apart from DEL);
4) graphics of ECMA-6 are not moved to other positions. A non-latin alphabet containing graphics which are also included in the latin alphabet is not subject to this rule.

In addition it is possible that a certain application may require for the basic set a non-"compatible" set. If so, it would be necessary to include the characters ESCAPE, SHIFT-OUT and SHIFT-IN in order to provide the facilities of Code Extension, described in this Standard.

Most basic sets consist of 128 characters but some character sets may consist of more than 128 characters by using two or more 7-bit combinations (see 4.3.3.2.7). Such sets are regarded as "compatible" variants if the four points of the note above are satisfied. Any reference in this document to a basic set should be recognized as including a multiple byte set as referred to above.

This Standard defines the method of identifying which basic set is effective. Most applications require the use of only one basic set. In this case the set is identified either by two appropriate ESCAPE sequences, one for the controls (see 4.3.3.2.1) and one for the graphics (see 4.3.3.2.4) or by agreement between the interchanging parties.

4.2 Extension of the Graphic Set by means of the Characters SHIFT-OUT and SHIFT-IN

4.2.1 Use of SO and SI

4.2.1.1 The character SHIFT-OUT (SO) and the character SHIFT-IN (SI) are used exclusively for extension of the graphic set, i.e. columns 2 to 7 of the code table excluding position 7/15. However, the character SPACE (position 2/0) is shift insensitive.

4.2.1.2 The character SO invokes an alternative set of not more than 94 graphics that are to replace the graphic characters of the basic set excluding the character SPACE (pos. 2/0). Graphic characters need not be assigned to all the positions of the alternative set nor, except as specified in 4.2.1.5, need all the graphic characters of the alternative set be different from the graphic characters of the basic set. Characters in the SHIFT-OUT set unchanged from the SHIFT-IN set appear twice in an 8-bit code constructed according to 7.1.

4.2.1.3 The character SI invokes the graphic characters of the basic set that are to replace the graphic characters of the alternative set.

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

1) Those corresponding to the 32 control characters in columns 0 and 1 and to the character DELETE (pos. 7/15);

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

3) Those included in any ESCAPE sequence.

4.2.1.5 The character SPACE occurs only at position 2/0; it shall not be assigned to any position in the alternative graphic set. These provisions do not preclude the assignment to positions in the basic set or in the alternative set of character equivalent to fixed SPACES other than the SPACE assigned to position 2/0.

4.2.1.6 At the beginning of any information interchange the shift status should be defined by a SI or SO character. When in SHIFT-IN status, a SI character has no effect, and, when in the SHIFT-OUT status a SO character has no effect.

4.2.2 <u>Unique SHIFT-OUT set</u>

Some applications require the use of only one alternative set of not more than 94 graphic characters. In such a case, that unique set is invoked by each use of the SO character.

The set is identified either by an appropriate ESC sequence as described in 4.3.3 or by agreement between the interchanging parties.

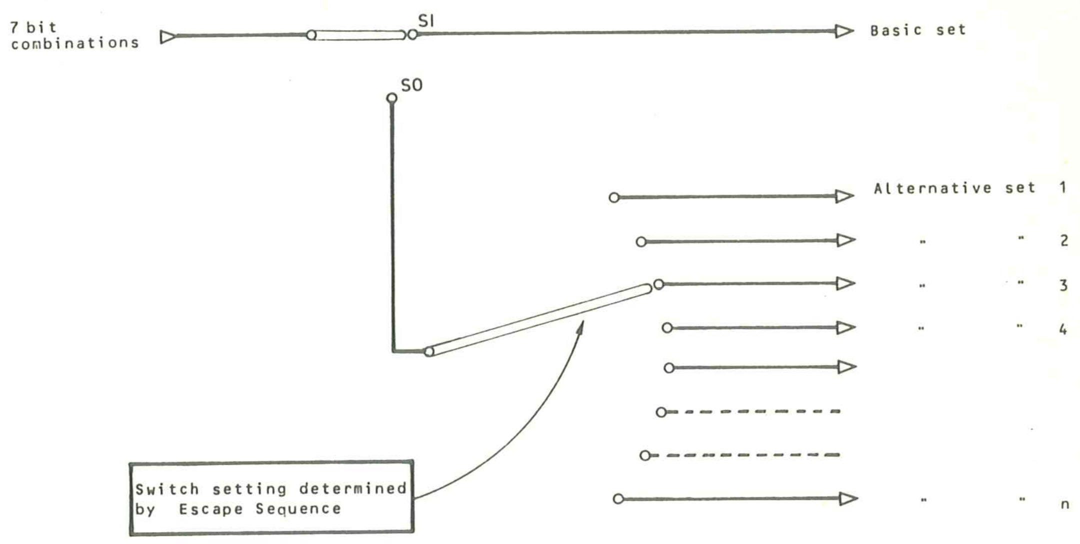
4.2.3 <u>Multiple SHIFT-OUT sets</u>

4.2.3.1 If two or more alternative graphic sets are required to coexist in a system the set to be used next is designated by the appropriate ESC sequence (see 4.3.3.2.5). That set can be invoked by the use of the SO character.

4.2.3.2 The use of the SI character re-instates the graphics of the basic set last designated, but does not affect the identity of the designated alternative set. The alternative set may be invoked any number of times by successive use of the SO character until it is superseded by another alternative set designated by another ESCAPE sequence.

It is not necessary to revert to the basic set by use of the SI character before changing from one alternative set to another by means of a further ESCAPE sequence. When the system is in the SHIFT-OUT state, the use of such a further ESCAPE sequence leaves the shift status unaltered, and the alternative set is invoked.

The following is the schematic representation of the above:



4.2.3.3 In some devices or systems it may be required to re-establish the SHIFT-IN state before designating a new SHIFT-OUT set by means of an ESCAPE sequence. This can be achieved by inserting a SI character just before the ESCAPE sequence which nominates the subsequent SHIFT-OUT set. This requirement must be agreed between any interchanging parties.

4.3 Code Extension by Means of ESCAPE Sequences

4.3.1 Purposes of ESCAPE Sequences

ESCAPE sequences provide single or sets of control functions other than for transmission control. Additional transmission control functions are obtained by the use of the character DATA LINK ESCAPE which is provided solely for this purpose. Its use is described in another Standard.

ESCAPE sequences are also used to designate sets of graphics, different uses of some or all of the 7-bit code combinations, and coded character sets with a number of bits other than 7.

Thus ESCAPE sequences are required to provide for example:

1) a single control not already in the code;

2) a set of controls not already in the code;

3) a set of graphics not already in the code;

4) a code structure different from that of the code.

4.3.2 Structure of ESCAPE Sequences

4.3.2.1 An ESCAPE sequence consists of two or more 7-bit combinations. The first is always the bit combination of the ESCAPE character and the last is always one of the Final characters (see 4.3.2.2). An ESCAPE sequence may also contain any number of 7-bit combinations representing intermediate characters (see 4.3.2.3).

4.3.2.2 The meaning of an ESCAPE sequence is determined by the 7-bit combination representing its intermediate characters if any, and by the 7-bit combination representing its Final character.

WARNING: Although in this Standard, ESCAPE sequences are described in terms of characters or positions of the code table of ECMA-6 the meaning of an ESCAPE sequence is determined only by its bit combinations and it is unaffected by any meaning previously assigned to those bit combinations, taken individually.

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

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

4.3.2.4 Final characters are the 79 characters of columns 3 to 7 of the 7-bit code table excluding the character DELETE (pos. 7/15).

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

4.3.2.5 Prohibited characters are the 32 control characters from columns 0 and 1 and the character DELETE (pos. 7/15).

Prohibited characters must not be used as either intermediate or final characters to construct an ESCAPE sequence.

As they may appear in an ESCAPE sequence in error, it may be necessary to provide methods of identifying such a situation to recover from it but this recovery is not subject to standardization.

4.3.3 Categories of ESCAPE Sequence

The use of two- or three-character ESCAPE sequences is specified in this Standard. ESCAPE sequences with 4 or more characters except those with final characters from column 3 are reserved for future standardization as the need may arise. The assignment and meaning of such combinations will be linked with the first intermediate character and will be interpreted as in 3-character ESCAPE sequences.

<u>WARNING</u>: The implementors of any private ESCAPE sequence described as such in this document are alerted to the fact that other implementors may give different meanings to the same ESCAPE sequences or may use different ESCAPE sequences to mean the same thing. Furthermore, such meanings may subsequently be standardized ESCAPE sequences.

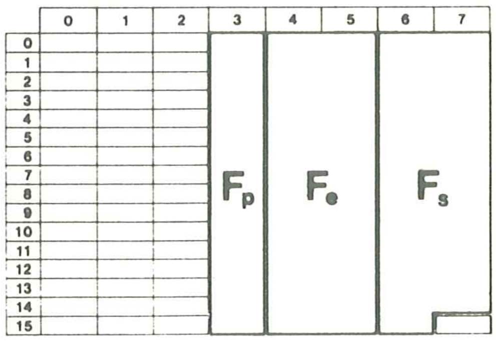
4.3.3.1 <u>Two-character ESCAPE sequences</u>

A two-character ESCAPE sequence takes the form:

<u>ESC</u> (F)

Such ESCAPE sequences are used to represent single additional controls.

The 79 two-character ESCAPE sequences are split into three types, depending on the Final character, as shown below:



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

An <u>ESC</u> (Fe) sequence represents, depending on the Final character used, an individual control out of a standardized set of 32 controls. 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 4.3.3.2.2 or by agreement between the interchanging parties. If more than one additional set of controls are required to coexist in a system the set to be used next is designated and invoked by the appropriate ESCAPE sequence.

NOTE: The use of ESC (Fe) sequences in relationship between 7- and 8-bit codes is described in section 8.2.2 of this Standard. These controls, represented in 7-bits by ESC (Fe) sequences are also represented as a set in an 8-bit code (see 7.1.2).

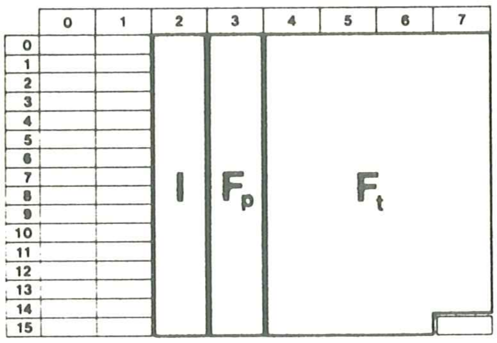
An ESC (Fp) sequence represents, depending on the Final character used, a single additional control without standardized meaning for use as required, subject to the prior agreement of the sender and the recipient of the data. The 16 Final characters of column 3 are provided for this purpose.

4.3.3.2 Three-character ESCAPE sequences

A 3-character ESCAPE sequence takes the form:

ESC (I) (F)

These sequences are split into two types according to their Final character as shown below.



ESC (I) (Ft) sequences are used for standardized controls. 63 Ft characters of columns 4-7 are provided for this purpose.

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

Both these types of 3-character ESCAPE sequences are grouped into classes according to their purpose, by means of their Intermediate characters, as listed below.

NOTE: In the following definitions of classes of 3-character ESCAPE sequences, intermediate characters are referred to by their column/row form (see 3.2). The column/row numbers are underlined to emphasize that they must be interpreted together and stand for one bit combination only.

4.3.3.2.1 ESCAPE sequences which define single additional control characters

ESC 2/3 (F) represents a single additional control character other than a transmission control.

4.3.3.2.2 ESCAPE sequences which define sets of 32 control characters for columns 0 and 1

ESC 2/1 (F) designates and invokes a set of 32 control characters into columns 0 and 1. To be "compatible" with ECMA-6 this set must have the following characteristic :

> NUL, SO, SI, CAN, SUB, ESC and the ten Transmission Controls remain unaltered in their meanings and their positions in the code table.

WARNING: Consideration should be given to the effect that changing the meaning of the control characters can have on equipment, when interchanging data. For example, the bit combination corresponding to "ESCAPE" will have the significance of ESCAPE to a system designed to respond to ESCAPE sequences.

4.3.3.2.3 ESCAPE sequences which define sets of 32 control characters for representation by ESC Fe sequences

ESC 2/2 (F) designates and invokes a set of 32 control characters additional to those of columns 0 and 1. Individual controls of such a set are represented in 7-bits by means of ESC Fe sequences (see 4.3.3.1). These sets may not contain transmission controls.

4.3.3.2.4 ESCAPE sequences which designate sets of 94 graphic characters for a basic set

ESC 2/8 (F) and ESC 2/12 (F) each designate and, if the system is in the SHIFT-IN state, invoke a set of 94 or less graphic characters for the basic set (see 4.1). If the system is in the SHIFT-OUT state, this set of graphics is not invoked until the next use of SHIFT-IN.

To be "compatible" with ECMA-6 such a set must contain no ECMA-6 graphics which have been moved to other positions. A non-latin alphabet containing graphics which are also included in the latin alphabet is not subject to this rule.

4.3.3.2.5 ESCAPE sequences which designate sets of 94 graphic characters for an alternative set

ESC 2/9 (F) and ESC 2/13 (F) each designate and, if the system is in the SHIFT-OUT state, invoke an alternative set of 94 or less graphic characters as specified in 4.2. If the system is in the SHIFT-IN state, this set of graphics is not invoked until the next use of SHIFT-OUT.

4.3.3.2.6 ESCAPE sequences for codes which require special interpretation

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

- a code with a number of bits other than 7, excluding those 8-bit codes structured in accordance with this Standard;
- a 7-bit code whose characteristics differ from those in this Standard.

The final character assignments are such that within the Ft and Fp groups the following classification occurs:

Final in Column	Broad Categorization
3	for private use of codes with any number of bits
4	a registered code of less than 7 bits
5	a registered code of 7 bits
6	a registered code of 8 bits
7	a registered code of more than 8 bits

4.3.3.2.7 ESCAPE sequences which define multiple-byte sets

ESC 2/4 (F) designates and, if the system is in SHIFT-IN state, invokes a set 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. Within such a set, each graphic character is represented by the same number of bytes. If the system is in the SHIFT-OUT state, this set of graphics is not invoked until the next use of SHIFT-IN.

4.3.3.2.8 Announcement of Extension Facilities

ESC 2/0 (F) announces the extension facilities used in conjunction with data which follows. Sequences allocated to this class are included in section 8.

4.3.3.2.9 <u>ESCAPE sequences without assigned meanings</u>

ESCAPE sequences which have not been assigned meanings are reserved for future standardization.

4.4 Pictorial Representation

Table 1 in the annexes summarizes in a schematic form the standard means of extension available within a 7-bit environment.

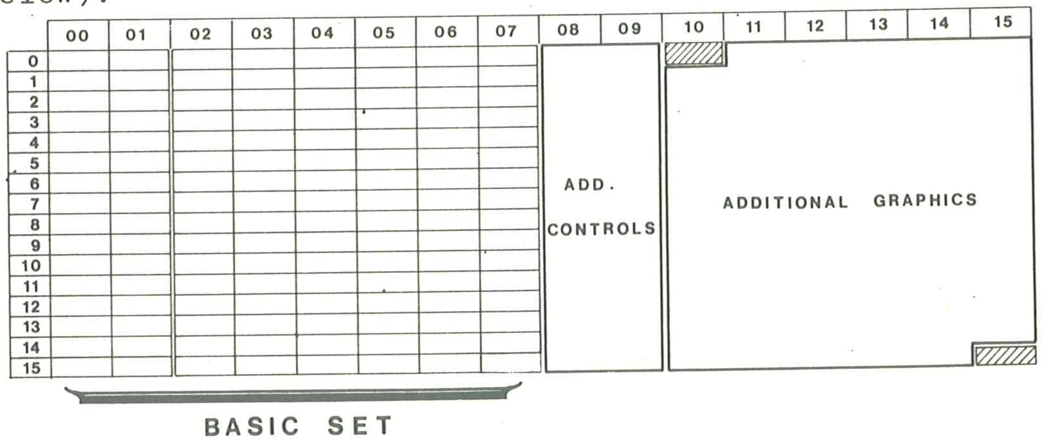
5. STRUCTURE OF A FAMILY OF 8-BIT CODES

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 basic set are assigned to the 128 bit combinations whose eighth bit is ZERO.

In this way, the 7-bit code table of the Standard ECMA-6 or any other "compatible" table, forms a defined and integral part of the 8-bit code table. The 128 additional bit combinations, whose eighth bit is ONE, are available for further assignment.

5.1 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 below).



<u>NOTE:</u> The columns of an 8-bit code table are numbered 00, 01,..., 15 to distinguish them from the columns of the 7-bit code table which are numbered 0, 1,..., 7.

Columns ØØ to 07 of this array contain 128 character positions which are in one-to-one correspondence with the 128 characters of the basic set defined in 4.1. Their coded representation is the same as in the 7-bit environment with the addition of an eighth more 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.

If control characters are required in addition to those of the basic set, they can only be placed in code positions of columns 08 and 09.

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 as for 7-bit codes includes the use of the character DATA LINK ESCAPE and is the subject of other ECMA Standards (ECMA-16, ECMA-24).

If graphic characters are required in addition to those of the basic set, they can only be placed in the code positions of columns 10 to 15, excluding positions 10/0 and 15/15. The SPACE character shall not be assigned to any position other than 2/0. These provisions do not preclude the assignment to bit combinations of an 8-bit code of characters equivalent to fixed SPACES other than the SPACE assigned to position 2/0.

In this Standard no meaning is assigned to position 10/0. The code position 15/15 (bit pattern 1111 1111) has a meaning equivalent to the character DELETE (pos. 7/15) of the 7-bit code. To facilitate compatibility with the 7-bit code, position 07/15 in an 8-bit code (bit pattern 0111 1111) shall always retain the meaning DELETE as in ECMA-6.

5.2 <u>The Family Concept</u>

The 8-bit code table includes a defined basic set of 128 characters together with the character in position 15/15.

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 controls can be selected for columns 08 and 09;
- a set of 94 additional graphics can be selected for columns 10 to 15 (excluding positions 10/0 and 15/15).

There are standard methods for identifying which selections of sets of controls and graphics have been allocated to the 8-bit table; these are described in Section 7.1.

6. DISTINGUISHING 7-BIT AND 8-BIT ORIGINATED DATA

The characters of the 7-bit code may be implemented in 7-bit media in the normal manner and may also be implemented in 8-bit media by the addition of an eighth bit which is ZERO. Both representations have equal validity and the relationship between them is simple. Similarly, the characters of the 8-bit code may be implemented in 7-bit media as well as in 8-bit media. Both representations have equal validity.

Code extension techniques have exactly the same meaning in a 7-bit or 8-bit environment but the use of SHIFT-OUT, SHIFT-IN and of ESC Fe sequences is not normal in an 8-bit code. It may therefore be useful to be able to distinguish between these two situations in an 8-bit environment if they are likely to occur together.

Identification that 7-bit code data follows is achieved by the occurrence of the sequence ESC 2/0 4/2 which is reserved permanently for this meaning only. This sequence has the same meaning in a 7- or 8-bit environment.

Identification that true 8-bit coded data follows is achieved by the occurrence of the sequence ESC 2/0 4/3, which is permanently assigned for this purpose and cannot have any other meaning. Its use is restricted to within an 8-bit environment.

7. EXTENSION OF THE 8-BIT CODE REMAINING IN AN 8-BIT ENVIRONMENT

Methods of extending an 8-bit code described in this Standard have been purposely limited to methods compatible with those used to extend the 7-bit code, and simple to use therein. This philosophy has been adopted to facilitate conversion of data represented in 7- and 8-bit environments.

Characters for extension purposes are included in the basic set of an 8-bit code. Their meanings and their positions in the code table are identical to those of the 7-bit code. They are:

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

This Standard does not describe the use of the character DATA LINK ESCAPE which is reserved for the provision of additional transmission controls; the use of this character is specified in other ECMA Standards (ECMA-16, ECMA-24).

The use of the characters SO and SI and of ESC Fe sequences is normally forbidden in an 8-bit environment (but see 7.1.3 and 7.1.2 for special circumstances).

The character ESCAPE is used in an 8-bit code in exactly the same way as in the 7-bit code to construct similar ESCAPE sequences (see 4.3). Thus characters from columns 08 to 15 cannot be used in an ESCAPE sequence.

7.1 Defining the 8-bit Code

As described in 5.2, the code table can be considered as split into three parts:

- the basic set
- the set of additional controls
- the set of additional graphics.

7.1.1 The basic set, as described in 4.1, must be defined by means of an ESC 2/1 (F) sequence and an ESC 2/8 (F) or an ESC 2/12 (F) sequence. These sequences designate and invoke the same character sets as in the 7-bit environment (4.3.3.2.2 and 4.3.3.2.4). These sets occupy the first 128 positions of the code table and have no effect on columns 08 to 15.

7.1.2 The set of additional controls required for columns 08 and 09 must be defined by means of an ESC 2/2 (F) sequence. This sequence designates and invokes the same set of controls as in the 7-bit environment (see 4.3.3.2.3).

The individual controls of such a set are represented in a 7-bit code by a 2-character ESCAPE sequence whose final character is from the corresponding row of column 4 or 5 of the 7-bit table. When using a true 8-bit code there is no need and it is contrary to the intention of this Standard to represent these controls by such 2-character ESCAPE sequences. However, it is recognized that in some transitory situations it may be desirable to keep 7-bit originated data in an 8-bit environment. In that case $a_8 = 0$ is merely added to the 7-bit combinations of the ESCAPE sequence and data is represented in a 7-bit oriented form.

7.1.3 The set of additional graphics required for columns 10 to 15 must be defined by means of an ESC 2/9 (F) or an ESC 2/13 (F) sequence. Such a sequence designates and invokes the same set of graphics which it designates as a SHIFT-OUT set in the 7-bit environment (see 4.3.3.2.5).

In an 8-bit code as specified in this Standard, the meaning of the characters SO and SI is not defined. However, it is recognized that in some transitory situations, it may be desirable to keep 7-bit originated data in an 8-bit environment. In that case $a_8 = 0$ is merely added to the 7-bit combinations and data is represented in a 7-bit oriented form; the characters SO and SI then retain the same meaning as in the 7-bit environment.

7.2 Code Extension by Means of ESCAPE Sequences

Once the 8-bit code is defined according to the rules specified in 7.1, code extension means are available, making use of the following ESCAPE sequences.

7.2.1 2-character ESCAPE sequences

Two-character ESCAPE sequences have the same structure as in the 7-bit environment (see 4.3.3.1).

- ESC (Fs) sequences represent single additional controls with the same meaning they have in the 7-bit environment.
- ESC (Fe) sequences are normally forbidden in an 8-bit environment (but see 7.1.2 for special circumstances).
- ESC (Fp) sequences represent single additional controls with the same meaning they have in the 7-bit environment.

7.2.2 3-character ESCAPE sequences

Three-character ESCAPE sequences have the same structure as in the 7-bit environment (see 4.3.3.2).

The following sequences are available in an 8-bit code for extension purposes:

- ESC 2/0 (F) sequences announce the extension facilities used in conjunction with data which follows (see 8).
- ESC 2/1 (F) sequences designate and invoke sets of controls for columns 00 and 01 (see 4.3.3.2.2).
- ESC 2/2 (F) sequences designate and invoke sets of controls for columns 08 and 09 (see 4.3.3.2.3).
- ESC 2/3 (F) sequences represent single additional controls with the same meaning they have in the 7-bit environment (see 4.3.3.2.1).
- ESC 2/4 (F) sequences designate and invoke sets of graphic characters that are represented by two or more bit combinations from column 02 to 07 (see 4.3.3.2.7).
- ESC 2/5 (F) sequences retain the same meaning they have in the 7-bit environment (see 4.3.3.2.6).
- ESC 2/8 (F)
- ESC 2/12 (F)

  These sequences designate and invoke sets of graphic characters for columns 02 to 07 (see 4.3.3.2.4).
- ESC 2/9 (F)
- ESC 2/13 (F)

  These sequences designate and invoke sets of graphics for columns 10 to 15 (see 4.3.3.2.5).

7.3 Pictorial Representation

Table II in the annexes summarizes in a schematic form the standard means of extension available within an 8-bit environment.

8. RELATIONSHIP BETWEEN THE 7-BIT CODE AND AN 8-BIT CODE

Transformation between 7-bit and 8-bit codes depends on which facilities of code extension are included in the application (see also 6).

8.1 Announcement of Extension Facilities Used

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

In the table below, the facilities used for each of the four final characters 4/1, 4/2, 4/3 and 4/4 are indicated. It should be noted that in a 7-bit environment, data announced by a sequence ESC 2/0 4/4 has the same form as data announced by a sequence ESC 2/0 4/2.

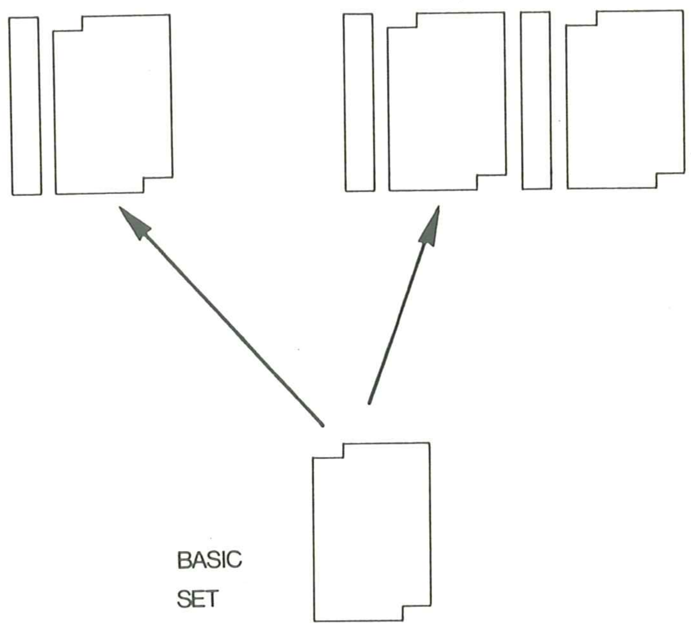
The announcer ESC 2/0 4/4 is provided for those interchange situations in which it is agreed to differentiate between 7-bit and 8-bit originated data in the 7-bit environment.

A pictorial representation of this table is shown on page 19 and 20.

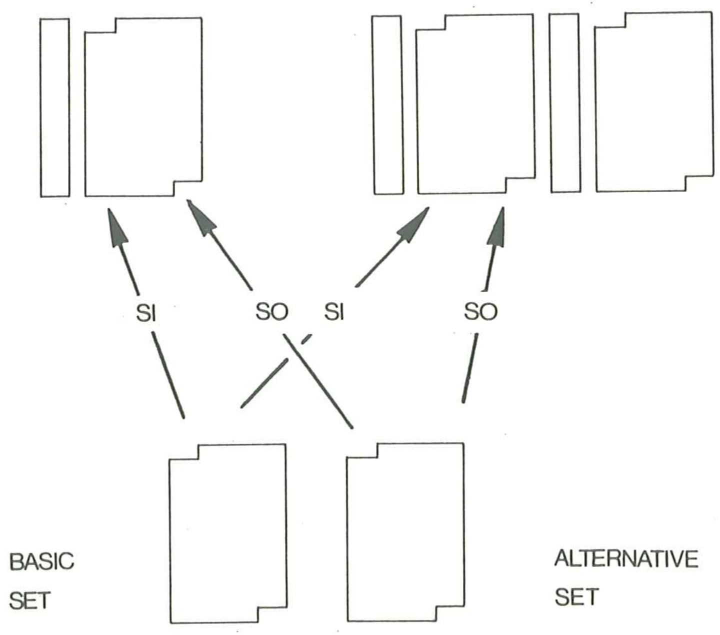
Final Characters	Facilities Utilized
4/1	One graphic set only is to be used. The ESCAPE sequence which designates this set also invokes it into columns 2 to 7. SI and SO are not to be used. In an 8-bit environment, columns 10 to 15 are not used.
4/2	Two graphic sets are to be used. In both 7 and 8-bit environments SI invokes the basic set into columns 2 to 7 and SO invokes the alternative set into columns 2 to 7. In an 8-bit environment columns 10 to 15 are not used.
4/3	Two graphic sets are to be used in an 8-bit environment only. The designating ESCAPE sequences also invoke the basic set and the alternative set into columns 02 to 07 and columns 10 to 15 respectively. SI and SO are not to be used.
4/4	Two graphic sets are to be used. In a 7-bit environment, SI invokes the basic set into columns 2 to 7 and SO invokes the alternative set into columns 2 to 7. In an 8-bit environment, the designating ESCAPE sequences also invoke the basic set and the alternative set into columns 02 to 07 and 10 to 15 respectively; SI and SO are not to be used. The transformation is described in 8.2

Final Character	7-Bit Environment	8-Bit Environment

<u>4/1</u>

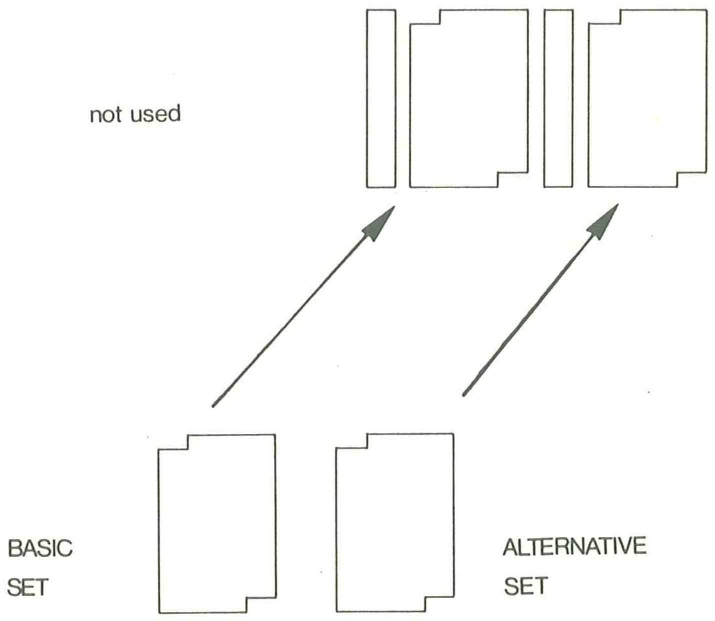


<u>4/2</u>

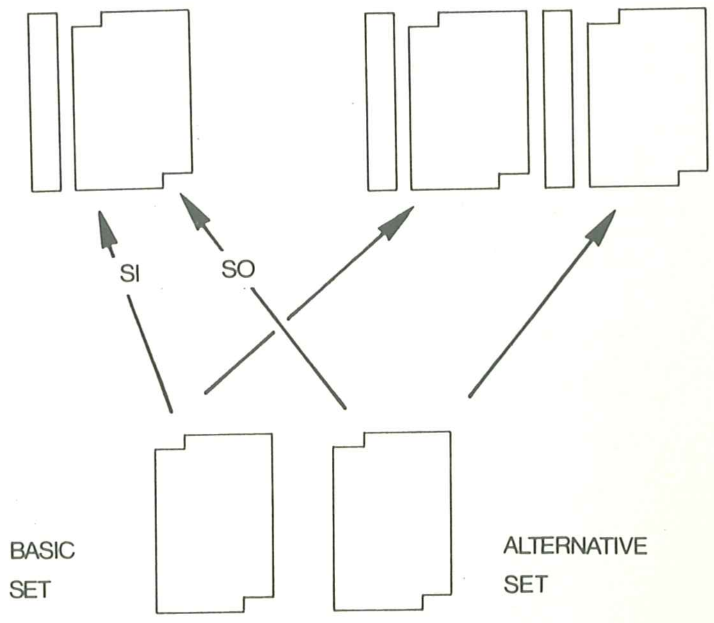


Final Character	7-Bit Environment	8-Bit Environment

4/3



4/4



8.2 Correspondence between the 7-bit Code and an 8-bit Code

The correspondence described below applies to data announced by the sequence ESC 2/0 4/4. In this case the occurrence, in an 8-bit environment, of either characters SI, SO or of the combinations 10/0, 15/15 or of ESC (Fe) sequences are conditions for which no handling procedures are prescribed in this Standard.

8.2.1 Columns 00 to 07

An 8-bit coded character in columns 00 to 07 is equivalent to the 7-bit coded character from the corresponding row of columns 0 to 7 respectively.

The relationship between the 7-bit coded character and the 8-bit coded character is:

$$a_8 = 0$$
$$a_7 = b_7$$
$$a_6 = b_6$$
$$a_5 = b_5$$
$$a_4 = b_4$$
$$a_3 = b_3$$
$$a_2 = b_2$$
$$a_1 = b_1$$

8.2.2 Columns 08 and 09 (additional control characters)

An 8-bit coded character in column 08 or 09 is equivalent to a 7-bit coded ESCAPE sequence consisting of ESC followed by the character from the corresponding row of columns 4 and 5 respectively. In other words, there is an exact equivalence between the 32 characters in columns 08 and 09 of the 8-bit code table and the 32 two-character ESC (Fe) sequences in the 7-bit code (see 4.3.3.1).

The relationship between the second character of the ESCAPE sequence and the 8-bit coded character is:

$$a_8 = 1$$
$$a_7 = 0, \qquad b_7 = 1$$
$$a_6 = \qquad b_6 = 0$$
$$a_5 = \qquad b_5$$
$$a_4 = \qquad b_4$$
$$a_3 = \qquad b_3$$
$$a_2 = \qquad b_2$$
$$a_1 = \qquad b_1$$

8.2.3 Columns 10 to 15 (additional graphic characters)

A string of 8-bit coded characters from columns 10 to 15 is equivalent to a string of 7-bit coded characters consisting of the character SO followed by the characters from the corresponding rows from columns 2 to 7 respectively. In other words, there is an exact equivalence between the 94 characters of columns 10 to 15 of the 8-bit code table (excluding the characters in positions 10/0 and 15/15) and the 94 characters of the SO set in the 7-bit environment (see 4.2).

If a control character in column 00 or 01 of the 8-bit code appears in the string, the corresponding control character in column 0 or 1 respectively of the 7-bit code will appear in the corresponding position of the equivalent string, i.e. control characters in columns 0 and 1 of the 7-bit code retain their normal meaning after SO. Similarly ESCAPE sequences are unaffected by a preceding SHIFT-OUT character.

The relationship between an 8-bit coded additional graphic character and the 7-bit coded character following the SO character is:

$$a_8 = 1$$

$$\left.\begin{aligned} a_7 &= b_7 \\ a_6 &= b_6 \end{aligned}\right) \quad \text{(where } b_7 \text{ and } b_6 \text{ are not both ZERO)}$$

$$a_5 = b_5$$

$$a_4 = b_4$$

$$a_3 = b_3$$

$$a_2 = b_2$$

$$a_1 = b_1$$

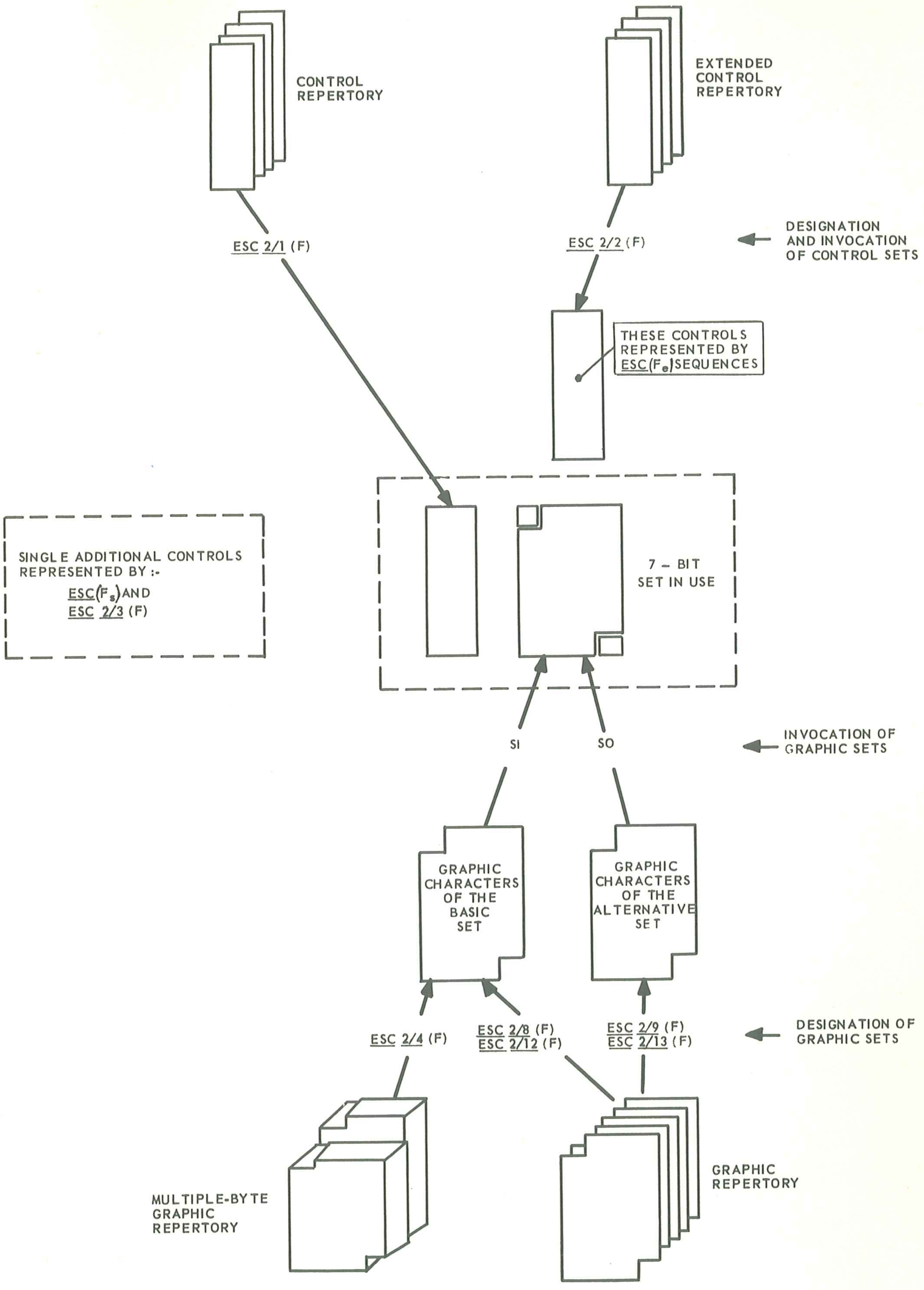

CONTROL REPERTORY
EXTENDED CONTROL REPERTORY
ESC 2/1 (F)
ESC 2/2 (F)
DESIGNATION AND INVOCATION OF CONTROL SETS
THESE CONTROLS REPRESENTED BY ESC(Fe)SEQUENCES
SINGLE ADDITIONAL CONTROLS REPRESENTED BY :- ESC(Fs)AND ESC 2/3 (F)
7 - BIT SET IN USE
SI
SO
INVOCATION OF GRAPHIC SETS
GRAPHIC CHARACTERS OF THE BASIC SET
GRAPHIC CHARACTERS OF THE ALTERNATIVE SET
ESC 2/4 (F)
ESC 2/8 (F)
ESC 2/12 (F)
ESC 2/9 (F)
ESC 2/13 (F)
DESIGNATION OF GRAPHIC SETS
MULTIPLE-BYTE GRAPHIC REPERTORY
GRAPHIC REPERTORY

Table I-Code extension in a 7-bit environment

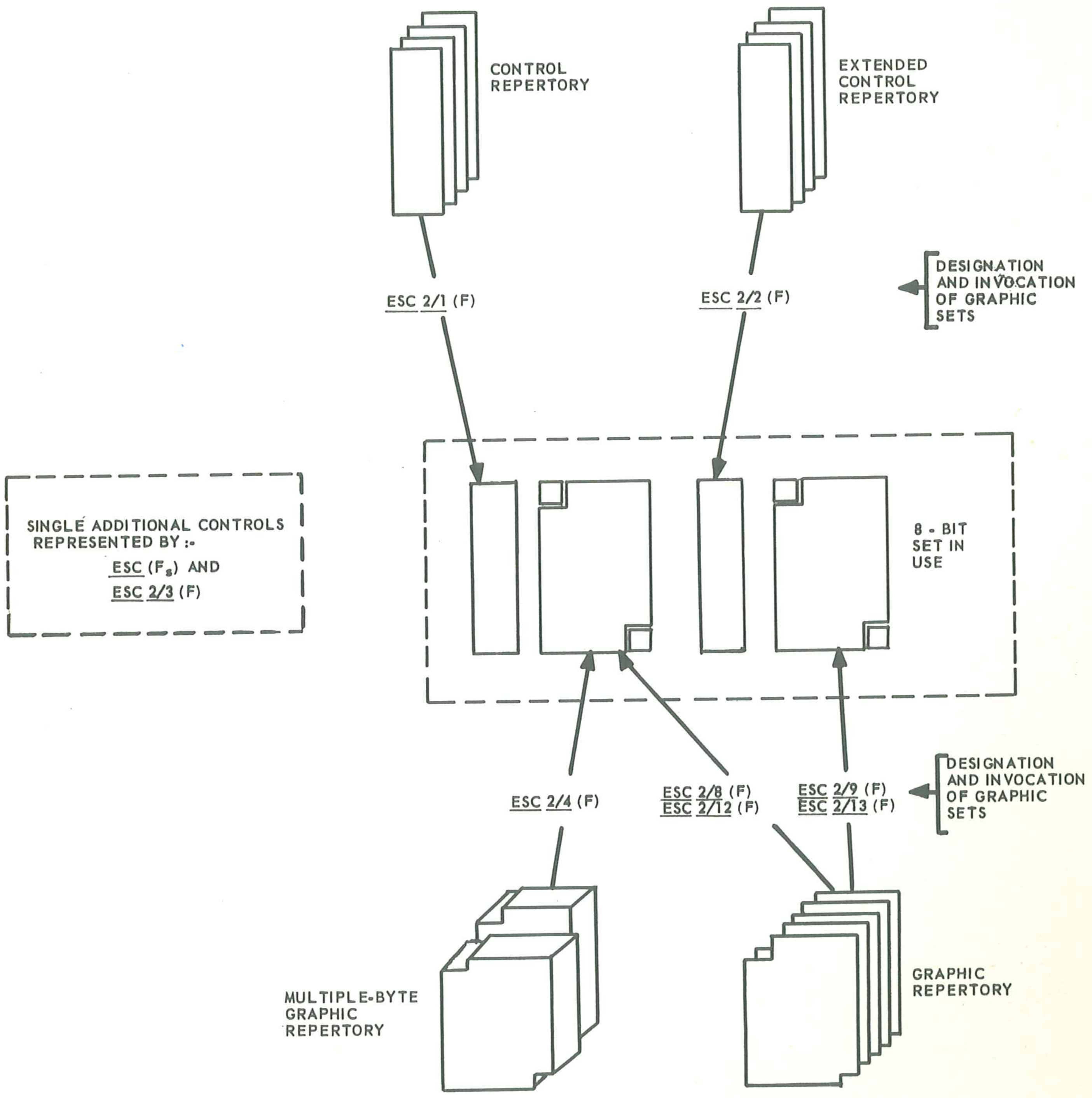

CONTROL REPERTORY
EXTENDED CONTROL REPERTORY
ESC 2/1 (F)
ESC 2/2 (F)
DESIGNATION AND INVOCATION OF GRAPHIC SETS
SINGLE ADDITIONAL CONTROLS REPRESENTED BY :-
ESC (Fs) AND
ESC 2/3 (F)
8 - BIT SET IN USE
ESC 2/4 (F)
ESC 2/8 (F)
ESC 2/12 (F)
ESC 2/9 (F)
ESC 2/13 (F)
DESIGNATION AND INVOCATION OF GRAPHIC SETS
MULTIPLE-BYTE GRAPHIC REPERTORY
GRAPHIC REPERTORY

Table II-Code extension in an 8-bit environment