American National Standard

Adopted for Use by the Federal Government

FIPS PUB 1-2

See Notice on Inside Front Cover

code extension techniques for use with the 7-bit coded character set of american national standard code for information interchange

american national standards institute, inc.
1430 broadway, new york, new york 10018

X3.41-1974

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 1-2, Code for Information Interchange, Its Representations, Subsets, and Extensions. For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

ANSI®
X3.41-1974

American National Standard Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange

Secretariat

Computer and Business Equipment Manufacturers Association

Approved May 14, 1974

American National Standards Institute, Inc

American National Standard

An American National Standard implies a consensus of those substantially concerned with its scope and provisions. An American National Standard is intended as a guide to aid the manufacturer, the consumer, and the general public. The existence of an American National Standard does not in any respect preclude anyone, whether he has approved the standard or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standard. American National Standards are subject to periodic review and users are cautioned to obtain the latest editions.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of publication. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Printed in the United States of America

A2½M685/8

Foreword

(This Foreword is not a part of American National Standard Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange, X3.41-1974.)

American National Standard Code for Information Interchange (ASCII), X3.4-1968, provides coded representation for a set of graphics and control characters having general utility in information interchange. In some applications, it may be desirable to augment the standard repertory of characters with additional graphics or control functions.

ASCII includes characters intended to facilitate the representation of such additional graphics or control function by a process known as code extension. Although the basic nature of code extension limits the degree to which it may be standardized, there are advantages to adherence to certain standard rules of procedure. These advantages include minimized risk of conflict between systems required to interoperate, and the possibility of including advance provision for code extension in the design of general-purpose data handling systems.

A need has also developed for 8-bit codes for general information interchange in which ASCII is a subset. This standard provides a structure which will accommodate this need.

This standard was developed after extensive study of various potential applications and of trends expected in system design.

Suggestions for improvement of this standard will be welcome. They should be sent to the American National Standards Institute, 1430 Broadway, New York, N.Y. 10018.

This standard was processed and approved for submittal to ANSI by American National Standards Committee on Computers and Information Processing, X3. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

J. F. Auwaerter, Chairman
V. E. Henriques, Vice-Chairman
Robert M. Brown, Secretary

Organization Represented	*Name of Representative*
Addressograph Multigraph Corporation	A. C. Brown
	D. S. Bates (Alt)
Air Transport Association	F. C. White
American Bankers Association	M. E. McMahon
	John B. Booth (Alt)
American Institute of Certified Public Accountants	N. Zakin
	P. B. Goodstat (Alt)
	C. A. Phillips (Alt)
	F. Schiff (Alt)
American Library Association	J. R. Rizzolo
	J. C. Kountz (Alt)
	M. S. Malinconico (Alt)
American Newspaper Publishers Association	W. D. Rinehart
American Nuclear Society	D. R. Vondy
	M. K. Butler (Alt)
American Society of Mechanical Engineers	R. W. Rau
	R. T. Woythal (Alt)
Association for Computing Machinery	P. Skelly
	J. A. N. Lee (Alt)
	L. Revens (Alt)
	H. Thiess (Alt)
Association for Educational Data Systems	C. Wilkes
Association for Systems Management	A. H. Vaughan
Association of American Railroads	R. A. Petrash
Association of Computer Programmers and Analysts	T. G. Grieb
	G. Thomas (Alt)
Association of Data Processing Service Organizations	J. B. Christiansen
Burroughs Corporation	E. Lohse
	J. F. Kalbach (Alt)
Control Data Corporation	S. F. Buckland
	C. E. Cooper (Alt)
Data Processing Management Association	A. E. Dubnow
	D. W. Sanford (Alt)

Organization Represented	*Name of Representative*
Edison Electric Institute	R. Bushner
	J. P. Markey (Alt)
Electronic Industries Association	(Representation Vacant)
	A. M. Wilson (Alt)
General Electric Company	R. R. Hench
	J. K. Snell (Alt)
General Services Administration	D. L. Shoemaker
	M. W. Burris (Alt)
GUIDE International	T. E. Wiese
	D. Stanford (Alt)
Honeywell Information Systems Inc	T. J. McNamara
	Eric H. Clamons (Alt)
Institute of Electrical and Electronics Engineers, Communications Society	R. Gibbs
Institute of Electrical and Electronics Engineers, Computer Society	G. C. Schutz
	C. W. Rosenthal (Alt)
Insurance Accounting and Statistical Association	W. Bregartner
	J. R. Kerber (Alt)
International Business Machines Corporation	L. Robinson
	W. F. McClelland (Alt)
Joint Users Group	T. E. Wiese
	L. Rodgers (Alt)
Life Office Management Association	B. L. Neff
	A. J. Tufts (Alt)
Litton Industries	I. Danowitz
National Association of State Information Systems	G. H. Roehm
	C. Vorlander (Alt)
National Bureau of Standards	H. S. White, Jr
	J. O. Harrison (Alt)
National Cash Register Company	R. J. Mindlin
	Thomas W. Kern (Alt)
National Machine Tool Builders' Association	O. A. Rodriques
	E. J. Loeffler (Alt)
National Retail Merchants Association	I. Solomon
Olivetti Corporation of America	E. J. Almquist
Pitney-Bowes Inc	D. J. Reyen
	B. Lyman (Alt)
Printing Industries of America	N. Scharpf
	E. Masten (Alt)
Scientific Apparatus Makers Association	A. Savitsky
	J. French (Alt)
SHARE Inc	T. B. Steel, Jr
	R. H. Wahlen (Alt)
Society of Certified Data Processors	A. Taylor
	J. J. Martin (Alt)
Telephone Group	V. N. Vaughan, Jr
	Stuart M. Garland (Alt)
	J. C. Nelson (Alt)
UNIVAC, Division of Sperry Rand Corporation	M. W. Bass
	Charles D. Card (Alt)
U.S. Department of Defense	W. L. McGreer
	W. B. Rinehuls (Alt)
	W. B. Robertson (Alt)
Xerox Corporation	J. L. Wheeler

Subcommittee X3L2 on Codes and Character Sets, which developed this standard, had the following members:

Eric H. Clamons, Chairman

Arthur H. Beaver
Theodore R. Boosquet
John B. Booth
Royce L. Calloway
Charles D. Card
Clarence C. Chandler
L. J. Clingman
Blanton C. Duncan
T. F. Fitzsimmons
John G. Fletcher
Stuart M. Garland
Jerry Goldberg
Marjorie F. Hill
Thomas O. Holtey
William F. Huf
H. F. Ickes
William F. Keenan
Thomas W. Kern
John L. Little
Herbert S. Meltzer
Charles Navoichick

Contents

SECTION PAGE

American National Standard Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange

1. Introduction

In the establishment of American National Standard Code for Information Interchange (ASCII), X3.4-1968, a fundamental decision had to be made as to the size of the code. In making such a decision there is usually a conscious effort to avoid the most obvious problems with a code that is either too large or too small. Should the number of characters included be too small, many users will find their needs not accommodated and will be forced to adopt "parochial" codes for their applications. Should the number of characters be too large, many users will find the code disproportionately costly to implement, or untenably inefficient in transmission or storage, and will again be driven to the use of some other code. Thus, either extreme in code sizing will reduce the generality of the code, defeating the very purpose of standardization in this field.

The 7-bit size (128 characters) adopted for ASCII is thought to be near optimum at present with respect to the foregoing considerations. Nevertheless, there will be numerous applications with requirements that are not accommodated by a code of this size, or at least not by the specific characters assigned within it. Many of these applications can be served by the use of the standard code augmented by standard code extension procedures. Through such an approach, the user may be able to implement much of his system with standard hardware and software. More significantly, he will thereby be able to retain compatibility with other systems for the interchange of that information which can adequately be directly represented by the standard code.

2. General

2.1 Scope

2.1.1 This standard specifies methods of extending the 7-bit code remaining in a 7-bit environment or increasing to an 8-bit environment. The description of techniques is contained in four interrelated sections dealing respectively with:

(1) The extension of the 7-bit code remaining in a 7-bit environment

(2) The structure of a family of 8-bit codes

(3) The extension of an 8-bit code remaining in an 8-bit environment

(4) The relationship between the 7-bit code and an 8-bit code

2.1.2 While ASCII is the agreed-upon code for information interchange, an 8-bit code as described in this standard is provided for information interchange within an 8-bit environment.

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

When two systems with different implementations of code extension techniques are required to communicate with one another, they shall do so using only the code extension techniques they have in common.

2.1.4 Code extension techniques are classified, and some classes are given a structure in this standard. Other assignments of bit combinations associated with the designation of the classes will be made in accordance with the procedures for the registration of characters and character sets given in a related standard currently under development. Specific assignments of bit combinations to relate individual codes with their invocation or designation will also be made in accordance with that standard.

2.1.5 Code extension techniques are designed to be used for data to be processed serially in a forward direction. Use of these techniques in strings of data that are processed other than serially in a forward direction or included in data formatted for fixed record processing may have undesirable results.

2.2 **Purpose**

2.2.1 American National Standard Code for Information Interchange (ASCII), X3.4-1968, specifies the representation of 128 characters. Additionally, it allows the representation of several other graphics by the combination of two graphic characters with the Backspace control character. In many instances, ASCII lacks either controls or graphics to sufficiently satisfy the needs of an application. These needs may be satisfied by means of code extension, which is the subject of this standard.

This standard is intended to present a review of the salient structure of ASCII and then build upon that structure to describe various means of extending the control and graphic set of the code. It also describes structures and techniques to construct or formalize codes related to ASCII. These related codes are structured so as to allow application-dependent usage without preventing the interchangeability of their data.

2.2.2 The standard describes the following:

(1) The structure of ASCII

(2) Extension of the 7-bit code, remaining in a 7-bit environment, and making use of code extension techniques

(3) Increasing the number of bits to 8, yet retaining a structure compatible with the 7-bit structure

(4) Increasing the number of bits to 8 and applying similar code extension techniques

2.2.3 In order to use identical techniques in each of the cases mentioned in 2.2.2, and to facilitate conversion between them, standard rules are necessary. This has the advantage of:

(1) Reducing the risk of conflict between systems required to interoperate

(2) Permitting provision for code extension in the design of systems

(3) Providing standardized methods of calling into use agreed-upon sets of characters

(4) Allowing the interchange of data between 7-bit and 8-bit environments

2.2.4 This standard also describes the structure of families of codes that are related to ASCII by their structure.

2.3 **Application**. Characters and character sequences conforming to this standard are used in ASCII or ASCII-related information interchange where additional control or graphic characters, or both, not in ASCII must be provided.

2.4 **Conformance**. Recorded or transmitted data are in conformance with this standard if all those bit patterns contained in the data stream have exactly, and only, the meanings specified in this standard and in a related standard currently under development.

WARNING: Products may provide or process bit patterns, code extension procedures, classes of Escape sequences, and code relationships not specified in this standard. Products may have the capability to record or transmit data that contain contiguous bits or bytes, the meaning of which is not intended as those specified in this standard.

A user application program may cause nonconforming bit patterns to be transmitted in interchange data or require the use of bit patterns, code extension procedures, classes of Escape sequences, and code relationships not as specified in this standard for further processing of the application data.

The use of such product features or user applications may render the data in nonconformance with this standard.

3. Definitions

In this standard, the following definitions shall apply:

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

byte. A bit string that is operated upon as a unit and whose size 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 relationships 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 character whose occurrence in a particular context initiates, modifies, or stops a control function.

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

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, of which the

first is the bit combination corresponding to the Escape character.

Final character. The character whose bit combination terminates an Escape sequence.

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

Intermediate character. A character whose bit combination occurs between the Escape character and the Final character in an Escape sequence, consisting of more than two bit combinations.

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.

represent. To use a prescribed bit combination with the meaning of a character in a set of characters that has been designated and invoked. To use an Escape sequence with the meaning of an additional control character.

4. Notational Conventions

4.1 General. The primary concept of code extension is to allow the meaning associated with a bit pattern to be changed in a discrete and orderly way. As a result, some shorthand notations are introduced to disassociate bit patterns and classes of bit patterns from the meanings in any one code set. A number of notational conventions have been established in this document to simplify the expressions for code extension.

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 sum of the decimal equivalent of bits b_7-b_5 (or a_8-a_5), and the row number is the sum of the decimal equivalent of bits b_4-b_1 (or a_4-a_1), giving to these bits the weights given in 4.2.

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

4.2 Character Representation. 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 and row reference:		*Column*				*Row*		
		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0

4.3 Character Notations. Characters may be represented as follows:

Mnemonics:	ENQ	ETB
Denotation:		
(Location in code table [column/row]):	0/5	1/7
Bit pattern (byte):	0000101	0010111

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

5. Extension of the 7-Bit Code Remaining in a 7-Bit Environment

5.1 Introduction

5.1.1 The Structure of the 7-Bit Code. The 7-bit code table of ASCII consists of an ordered set of controls and graphic characters grouped (see Fig. 1) as follows:

(1) The set of thirty-two control characters allocated to columns 0 and 1 (hereafter referred to as the C0 set)

(2) The Space character in position 2/0, which may be regarded either as a control character or a nonprinting graphic character

(3) The set of ninety-four graphic characters allocated to columns 2-7 (hereafter referred to as the G0 set)

(4) The Delete character in position 7/15

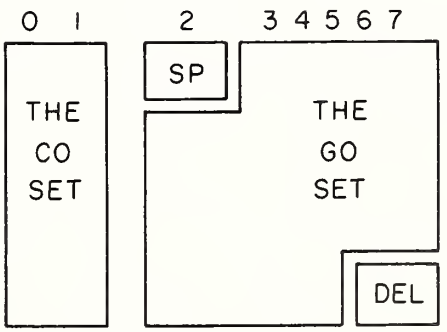


Fig. 1
The Structure of the 7-Bit Code

5.1.2 Extension by Substitution. In many cases the provisions of ASCII will satisfy the requirements of an application. Other applications, however, will be satisfied by the use of a similarly structured code in which some of the characters of ASCII are substituted by other characters. Such a substitution may be regarded as a replacement of the C0 set or the G0 set, or both, accordingly as new controls or graphics, or both, are required (see Appendix B of American National Standard X3.4-1968).

5.1.3 Extension by Increasing the Repertory of Characters. This standard provides for the addition of characters to the 128 provided by the structure of the 7-bit code in the following ways:

(1) Additional single controls

(2) Additional sets of thirty-two control characters (referred to as C1 sets)

(3) Additional sets of ninety-four graphic characters (referred to as G1 sets)

(4) Sets of more than ninety-four graphic characters, each represented by more than one byte (these sets function as G0 sets)

Many applications will require combinations of the aforementioned facilities. These facilities are shown in Fig. 2.

5.1.4 Code Extension Characters. In the 7-bit code, the following characters are provided for the purpose of code extension:

Escape character:	ESC
Shift-Out character:	SO
Shift-In character:	SI
Data Link Escape character:	DLE

This standard does not describe the use of the Data Link Escape character, which is reserved for the provision of additional transmission controls; the use of this character is specified in American National Standard Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links, X3.28-1971.

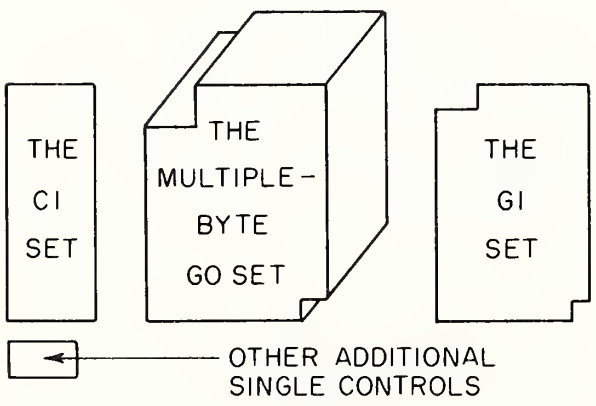


Fig. 2
Structure for the Extended 7-Bit Code

5.1.5 Compatibility. For purposes of interchange, various levels of compatibility that may be preserved among extension facilities are identified as follows:

(1) A set that is compatible with ASCII inasmuch as:

(a) Columns 0 and 1 contain only control characters

(b) Columns 2–7 are used for graphic characters only (apart from DEL)

(c) The ten transmission control characters and NUL, SO, SI, CAN, SUB, ESC, and DEL remain unaltered in their meanings and in their positions in the code table

(d) Graphics of ASCII are not moved to other positions (A non-Latin alphabet containing graphics that are also included in the Latin alphabet is not subject to this rule.)

(2) Other sets structured as in 5.1.1

To be able to provide the facilities of code extension given in this standard, the Escape, or the Shift-Out and Shift-In characters, or both, shall remain unaltered in their meanings and their positions in the code table.

5.2 Extension of the Graphic Set by Means of the Shift-Out and Shift-In Characters

5.2.1 Use of SO and SI. The Shift-Out character SO and the Shift-In character SI are used exclusively for extension of the graphics.

The character SO invokes an alternative set of ninety-four graphics, the G1 set. This set replaces the graphic characters of the G0 set. Graphic characters need not be assigned to all the positions of the alternative set, nor need all the graphic characters of the alternative set be different from the graphic characters of the G0 set.

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

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

(1) Those corresponding to the control characters in columns 0 and 1 and the Delete character in position 7/15

(2) The one corresponding to the Space character in position 2/0

(3) Those included in any Escape sequence

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

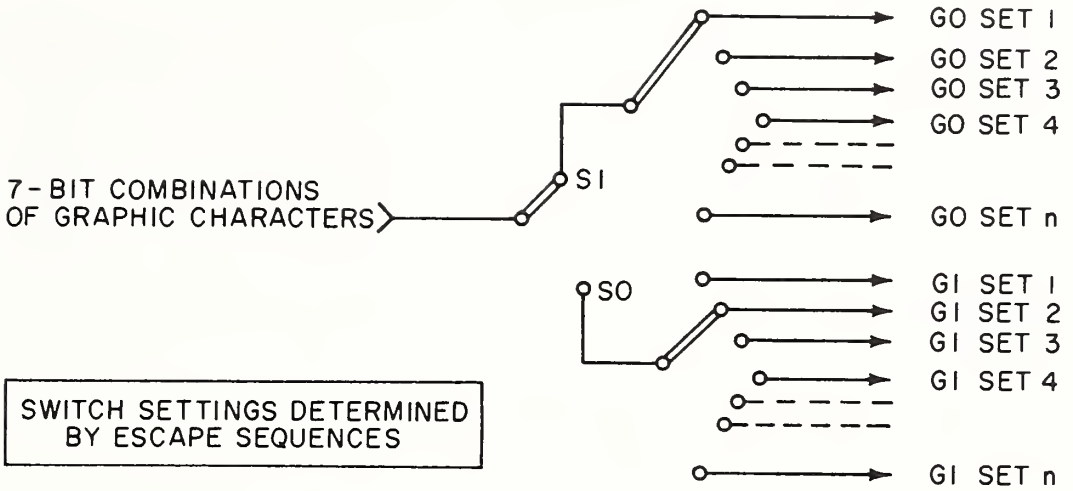


Fig. 3
Schematic Illustration of the Shift-Out and Shift-In Concept

At the beginning of any information interchange, the shift status should be defined by SI or SO. When in the Shift-In state, SI has no effect; and when in the Shift-Out state, SO has no effect.

5.2.2 Unique Shift-Out Set. Some applications require the use of only one alternative set of ninety-four graphic characters. In such a case, that unique set is invoked by each use of SO. The set is identified either by an appropriate ESC sequence as described in 5.3.7 or by agreement between or among the interchanging parties.

5.2.3 Multiple Shift-Out Sets. 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. That set can then be invoked by the use of SO.

The use of SI reinvokes the graphics of the G0 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 SO until it is superseded by another alternative set designated by another Escape sequence.

It is not necessary to revert to the G0 set by use of SI 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.

A schematic representation of the aforementioned is given in Fig. 3.

WARNING: In some devices or systems there may be a requirement to reestablish the Shift-In state before designating a new Shift-Out set by means of an Escape sequence. This can be achieved by inserting SI before the Escape sequence which designates the subsequent Shift-Out set. Such a requirement shall be agreed upon between or among any interchanging parties.

5.3 Code Extension by Means of Escape Sequences

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

Thus, Escape sequences are required to provide, for example:

(1) A single control character not already in the code

(2) A set of control characters not already in the code

(3) A set of graphic characters not already in the code

(4) A code of different structure

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

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

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

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

NOTE: In this standard, any one of these sixteen Intermediate characters is denoted by the symbol "I."

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

NOTE: In this standard, any one of these seventy-nine Final characters is denoted by the symbol "F."

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

The thirty-three prohibited characters shall not be used as either Intermediate or Final characters to construct an Escape sequence.

As these prohibited characters may appear in an ESC sequence in error, it may be necessary within an application to provide methods of identifying such a situation and of recovering from it, but this is not covered in this standard (see Section A1 of Appendix A).

5.3.3 Categories of Escape Sequences. The categories of Escape sequences are specified in this standard. However, Escape sequences with Final characters from column 3 are reserved for private use subject to the categorization outlined in 5.3.3.1 and 5.3.3.2.

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

5.3.3.1 Two-Character Escape Sequences. A two-character Escape sequence takes the form ESC F. Such Escape sequences are used to represent single additional characters. The seventy-nine two-character Escape sequences are split into three types, depending on the Final character, as shown in Fig. 4.

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

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

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

5.3.3.2 Three-Character Escape Sequences. A three-character Escape sequence takes the form ESC I F.

All types of three-character Escape sequences are grouped into categories according to their purpose, by means of their Intermediate characters, as shown in 5.3.4 through 5.3.10 (see Table 1).

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

ESC I Ft sequences are used for standardized purposes. The sixty-three Ft characters of columns 4-7 are provided for this purpose.

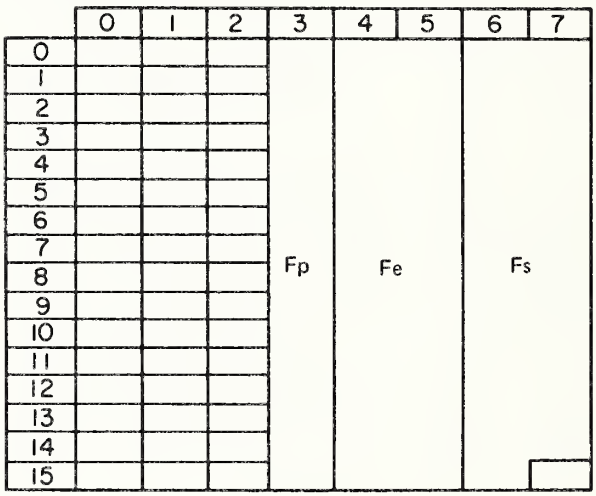


Fig. 4
Portions of the Code Table Used in Two-Character Escape Sequences

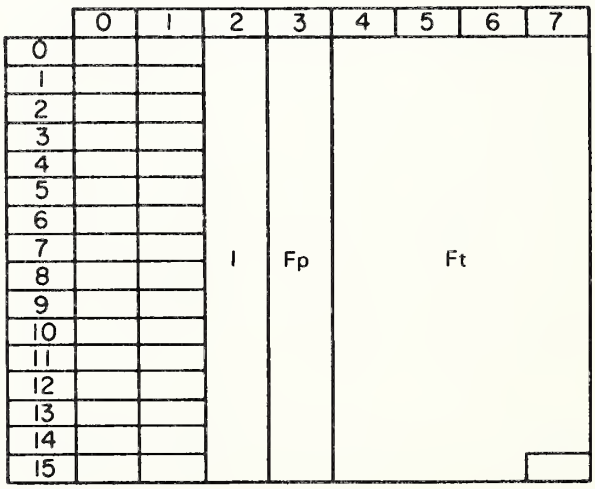


Fig. 5
Portions of the Code Table Used in Three-Character Escape Sequences

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

5.3.4 Single Additional Characters. ESC 2/3 F represents, depending on the Final character used, a single additional character.

5.3.5 Sets of Thirty-Two Control Characters for Columns 0 and 1. ESC 2/1 F designates and invokes the C0 set of thirty-two 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 shall be included in a C0 set.

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

(1) Inclusion of the ten transmission control characters

(2) Inclusion of the characters NUL, SO, SI, CAN, SUB, and ESC with their meanings and their position in the code table unaltered

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

5.3.6 Sets of Thirty-Two Control Characters for Representation by ESC Fe. ESC 2/2 F designates and invokes the C1 set of thirty-two control characters without affecting the C0 set. Individual control characters of such a set are represented by means of ESC Fe sequences rather than single-character bit combinations. A C1 set shall not include transmission control characters.

5.3.7 Sets of Ninety-Four Graphic Characters. ESC 2/8 F and ESC 2/12 F designate sets of ninety-four graphic characters that 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 ninety-four graphic characters that will be used as the G1 set. The designated set is invoked by SO.

The two aforementioned groups of graphic sets together make up a single repertory of graphic sets which may be designated to either of two available positions, G0 or G1. No significance is attached to the groupings other than that their existence allows more such sets of ninety-four graphic characters to be defined within the scope of three-character Escape sequences. There are, therefore, 126 such sets possible for standardization without requiring further extension (see 5.3.12).

5.3.8 Codes That Require Special Interpretation. ESC 2/5 F designates and invokes a code that requires special interpretation, such as:

(1) A code with a number of bits other than seven, excluding those 8-bit codes structured in accordance with this standard

(2) 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 (columns 3-7), the following classification occurs:

Final in Column	*Broad Categorization*
3	Private code with any number of bits
4	Code of less than 7 bits
5	Code of 7 bits
6	Code of 8 bits
7	Code of more than 8 bits

5.3.9 Sets of Graphics with Multiple-Byte Representation. ESC 2/4 F designates sets of graphic characters that are represented by two or more bytes, each corresponding to a bit combination in columns 2-7, apart from positions 2/0 and 7/15. The designated set is invoked by SI and is therefore regarded as a G0 set. Within such a set, each graphic character is represented by the same number of bytes as shown in Fig. 6.

5.3.10 Announcement of Extension Facilities. ESC 2/0 F announces the extension facilities used in conjunction with data that follow. The use of these sequences is specified in Section 8.

5.3.11 Three-Character Escape Sequences without Assigned Meanings. The three-character Escape sequences ESC 2/6 F, ESC 2/7 F, ESC 2/10 F, ESC 2/11 F, ESC 2/14 F, and ESC 2/15 F have not been assigned meanings and are reserved for future standardization.

5.3.12 Escape Sequences Having Four or More Characters. Escape sequences having four or more characters shall be interpreted according to the following:

(1) The first Intermediate character shall indicate the category of usage identical with three-character Escape sequences in 5.3.11

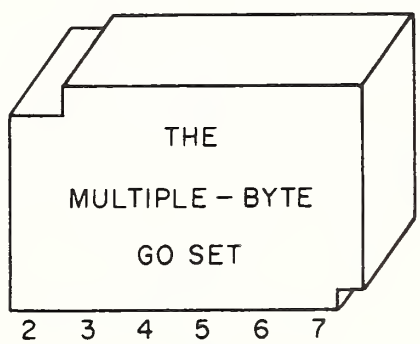


Fig. 6
The Multiple-Byte G0 Set

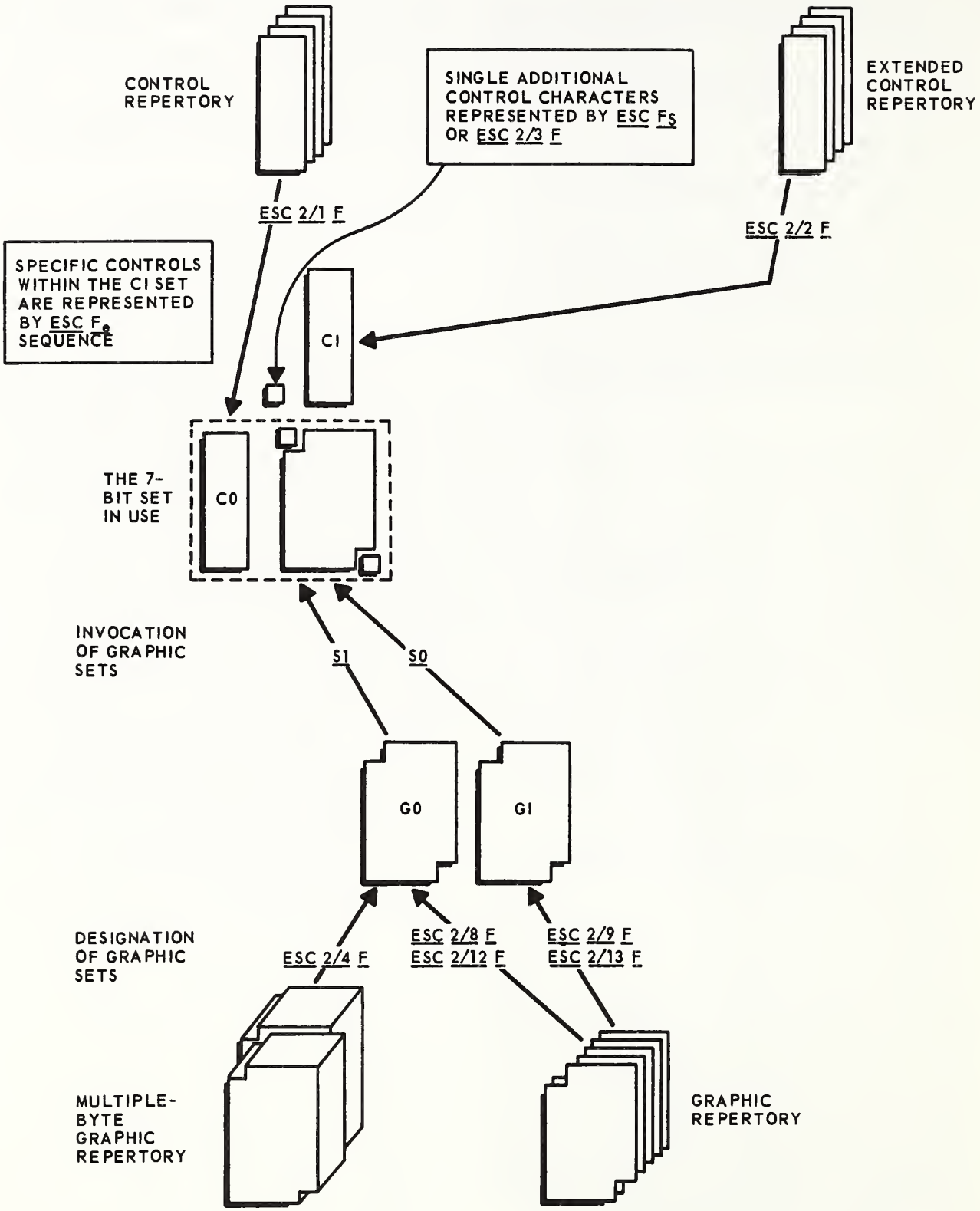


Fig. 7
7-Bit Code Extension Structure Using C0, C1, G0, and G1 Sets

Table 1
Summary of Assignments of Three-Character Escape Sequences

Column/ Row	Bits of Intermediate b_7 b_6 . . . . b_1	Category	Grouping	Subsection
2/0	0 1 0 0 0 0 0	Announcers		5.3.10
2/1	0 1 0 0 0 0 1	CONTROLS	C0 set	5.3.5
2/2	0 1 0 0 0 1 0		C1 set	5.3.6
2/3	0 1 0 0 0 1 1	Single characters		5.3.4
2/4	0 1 0 0 1 0 0	GRAPHICS	Multiple-byte sets	5.3.9
2/5	0 1 0 0 1 0 1	Codes requiring special interpretation		5.3.8
2/6	0 1 0 0 1 1 0	Reserved for future standardization		5.3.11
2/7	0 1 0 0 1 1 1			
2/8	0 1 0 1 0 0 0	GRAPHICS*	G0 set	5.3.7
2/9	0 1 0 1 0 0 1		G1 set	
2/10	0 1 0 1 0 1 0		Reserved for future standardization	5.3.11
2/11	0 1 0 1 0 1 1			
2/12	0 1 0 1 1 0 0		G0 set	5.3.7
2/13	0 1 0 1 1 0 1		G1 set	
2/14	0 1 0 1 1 1 0		Reserved for future standardization	5.3.11
2/15	0 1 0 1 1 1 1			

*There is a single repertory of sets of ninety-four graphic characters. Any member of the repertory may be designated as either a G0 or G1 set. Four designating Escape sequences, two for G0 and two for G1, are provided for designating members of the repertory.

(2) The second and any additional Intermediate characters shall be associated with the Final character to permit additional entities within the category defined by the first Intermediate character

(3) All Escape sequences having four or more characters whose final character is of the Ft type are reserved for standardization

(4) All Escape sequences whose final character is of the Fp type (private) are not to be the subject of standardization

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

5.4 Pictorial and Tabular Representations. Fig. 7 summarizes in pictorial form the standard means of code extension within a 7-bit environment.

Table 1 summarizes in tabular form the assignment of Intermediate characters in Escape sequences.

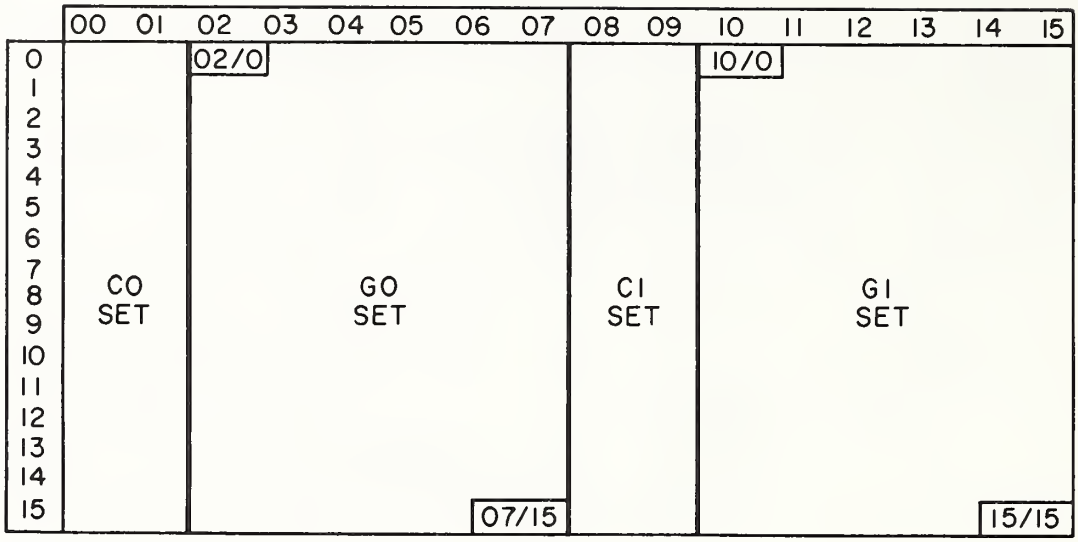


Fig. 8
The 8-Bit Code Table

6. Structure of a Family of 8-Bit Codes

6.1 General. 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 128 characters of the 7-bit code, the set as defined under 5.1, form a defined and integral part of an 8-bit code that is structured in accordance with this document. The 128 additional bit combinations, whose 8th bit is 1, are available for future assignment.

6.2 The 8-Bit Code Table. A 16 X 16 array of columns numbered 00–15 and rows numbered 0–15 contains 256 code positions (see Fig. 8). The 8-bit code table consists of an ordered set of controls and graphic characters grouped as follows (see Fig. 8):

(1) A set of thirty-two control characters allocated to columns 00 and 01 (C0 set)

(2) A set of ninety-four graphic characters allocated to columns 02–07, subject to the exception of positions 02/0 and 07/15 (G0 set)

(3) The Space character in position 02/0, which may be regarded either as a control character or a non-printing graphic character

(4) A set of thirty-two control characters allocated to columns 08 and 09 (C1 set)

(5) A set of ninety-four graphic characters allocated to columns 10–15, subject to the exception of positions 10/0 and 15/15

The control characters in columns 08 and 09 of an 8-bit code shall not include transmission control characters. Provision of data transmission capability for 8-bit codes includes the use of the Data Link Escape character and is covered in American National Standard Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links, X3.28-1971.

6.3 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:

(1) A set of thirty-two additional control characters can be selected for columns 08 and 09

(2) A set of ninety-four additional graphic characters can be selected for columns 10–15 (excluding positions 10/0 and 15/15)

There are standard techniques for identifying selections of sets of controls and graphics for 8-bit codes. These techniques are described in Sections 7, 8, and 9.

7. The Use of Code Extension in an 8-Bit Code

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

The Escape character is used in an 8-bit code in exactly the same way as in the 7-bit code to construct ESC sequences. Except as provided in 7.2, the meanings of these sequences are not altered in an 8-bit code. All characters in columns 08–15 are excluded from assign-

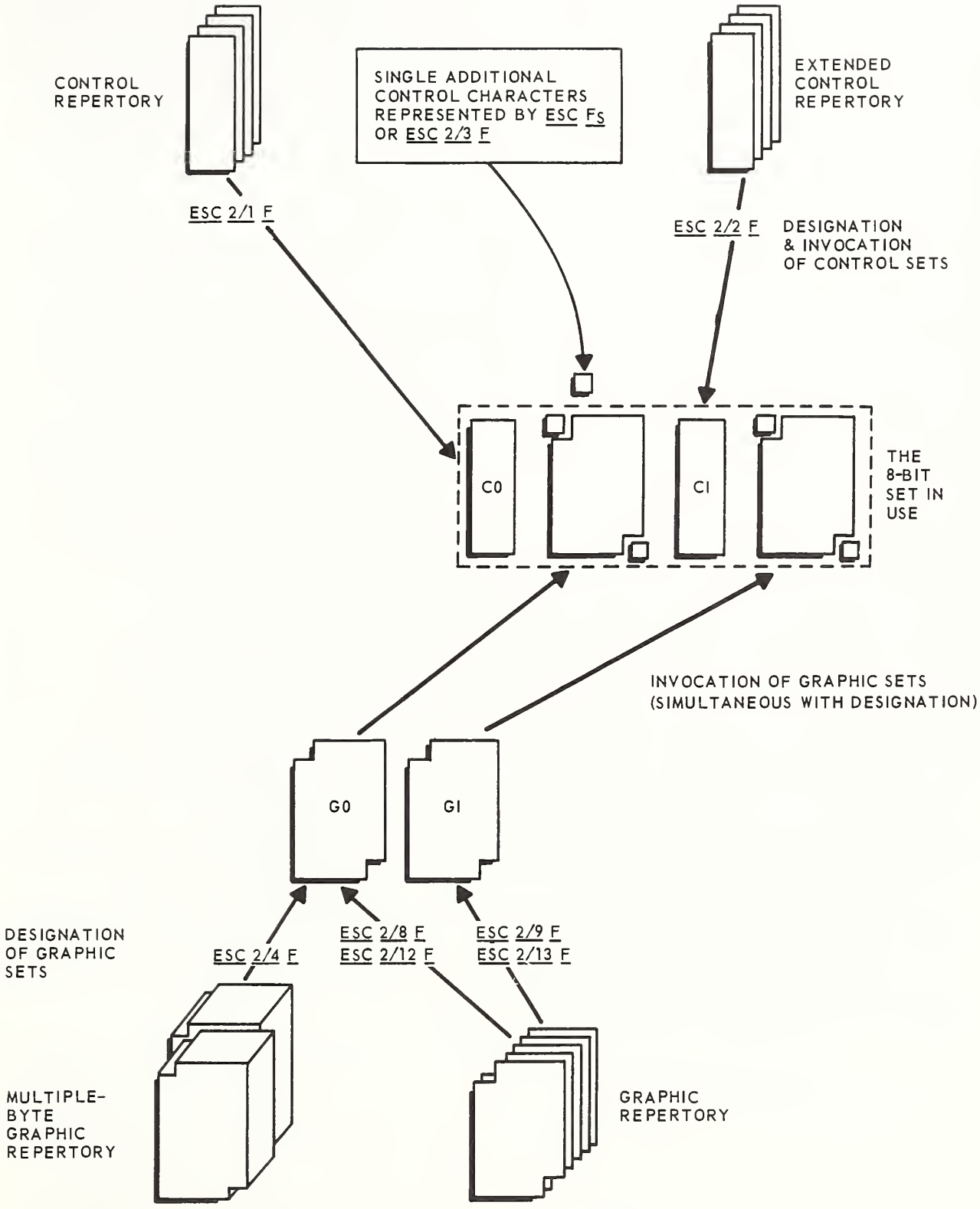


Fig. 9
8-Bit Code Extension Structure Using C0, C1, G0, and G1 Sets

ment 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.

7.2 Defining an 8-Bit Code. As described in Section 6, the code table can be considered as having four main parts: the C0 control set, the G0 graphic set, the C1 control set, and the G1 graphic set.

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

The C0 and G0 sets are designated and the C0 set is invoked as in the 7-bit environment (see 5.3.5 and 5.3.7).

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

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

7.3 Code Extension by Means of Escape Sequences. Once the 8-bit code is established in accordance with 7.2, code extension means are available, making use of Escape sequences as described therein and in 7.3.1 through 7.3.3.

7.3.1 Two-Character Escape Sequences. Two-character Escape sequences have the same structure as in the 7-bit environment (see 5.3.2).

7.3.1.1 ESC Fs sequences represent single additional controls with the same meaning they have in the 7-bit environment.

7.3.1.2 The use of ESC Fe sequence in an 8-bit environment is contrary to the intention of this standard but, should they occur, their meaning is the same as in the 7-bit environment.

7.3.2 Three-Character Escape Sequences. Three-character Escape sequences have the same structure as in the 7-bit environment (see 5.3.2).

7.3.3 Escape Sequences Having Four or More Characters. These sequences have the same structure and meaning as in the 7-bit environment (see 5.3.2 and 5.4).

7.4 Pictorial Representation. Fig. 9 summarizes in pictorial form the standard means of extension available in an 8-bit environment.

8. Announcement of Extension Facilities Used

The class of three-character Escape sequences ESC 2/0 F is used in data interchange to announce the code extension facilities utilized in the data that follow. Subject to agreement between or among the interchanging parties, such an announcing sequence may be omitted. The Final character of the announcing sequence indicates the facilities used for representing graphic sets in 7- and 8-bit environments and the number of bits used as given in Table 2.

9. Relationship between 7-Bit and 8-Bit Codes

9.1 Transformation between 7-Bit and 8-Bit Codes. Transformation between 7-bit and 8-bit codes depends on which facilities of code extension are included in the application. Examples of such transformations are given in Fig. B7 and B8 in flowchart form.

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

Table 2
Announcer Sequences and Meanings*

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

*A pictorial representation of these cases is shown in Fig. 10.

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

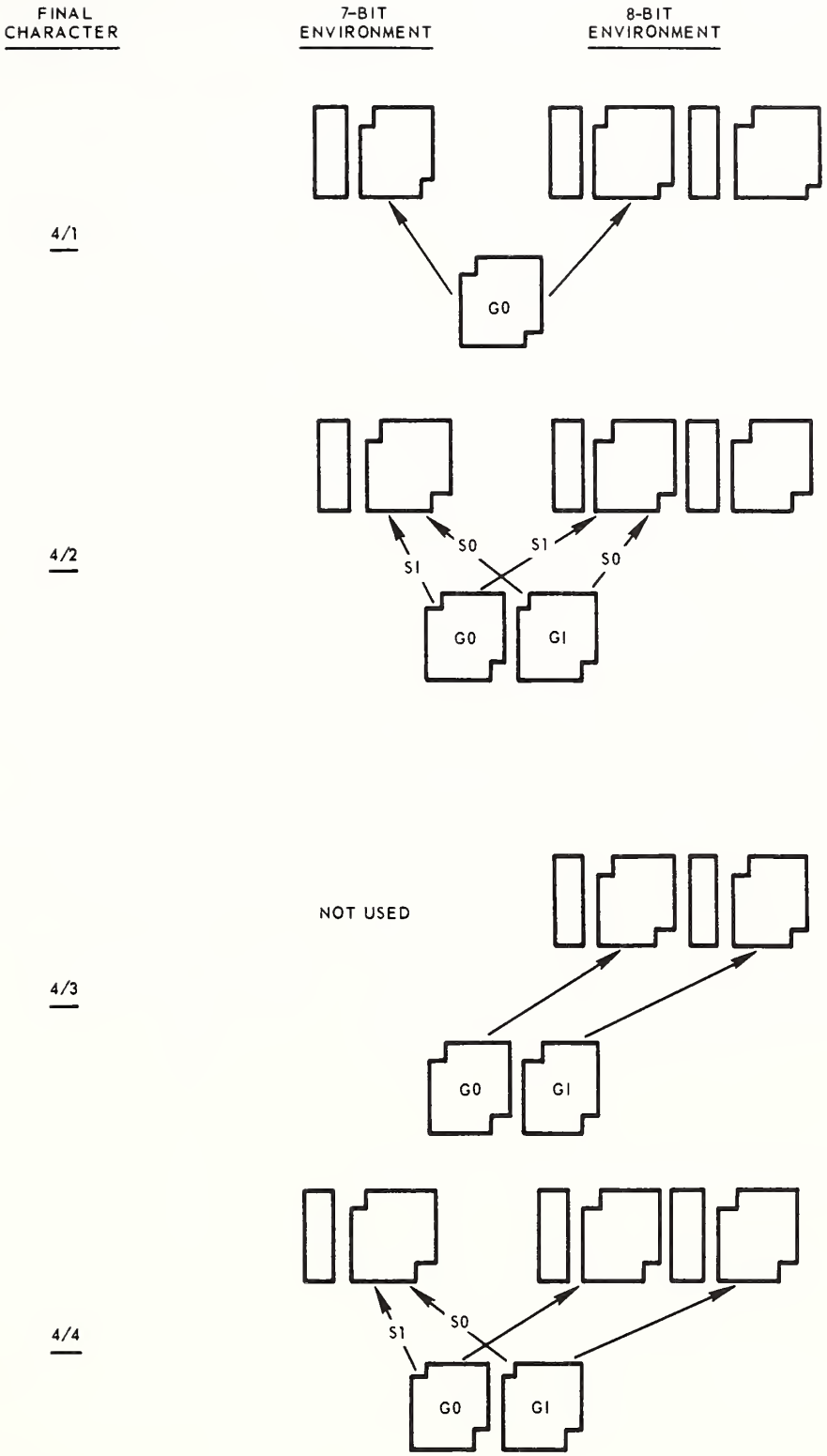


Fig. 10
Pictorial Representation of the Announcer Facilities

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

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

9.3 Representation of Positions 10/0 and 15/15 in a 7-Bit Environment. No meaning is assigned to positions 10/0 and 15/15 in this standard. If there is a requirement to represent these positions in a 7-bit environment, a private Escape sequence shall be used.

10. Specific Meaning of Escape Sequences

The meanings of individual Escape sequences are not specified in this standard. Instead, their meanings will be specified using the procedures for the registration of characters and character sets given in a related standard currently under development which are to be followed in preparing and maintaining a register of Escape sequences and their meanings. These registration procedures do not apply to Escape sequences reserved for private use.

Appendixes

(These Appendixes are not a part of American National Standard Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange, X3.41-1974, but are included for information purposes only.)

Appendix A
Implementation Considerations

A1. Appearance of Control Characters in Escape Sequences

A1.1 Although this standard precludes the assignment of control characters from columns 0 and 1 in Escape sequences, it does not necessarily preclude the occurrence of control characters in Escape sequences. System designers must be aware that additional standards may have to be considered if the occurrence of control characters is to be taken into account. Of particular concern is American National Standard Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links, X3.28-1971. Under the provisions of that standard, communication controls may be interjected into Escape sequences as a part of the communication control procedures.

A1.2 In interpreting data streams, the communication control characters and communication control sequences formed with the Data Link Escape (DLE) character as well as any associated error check characters (Block Check characters or Cyclic Redundancy Check characters), if present, should not be allowed to affect the meanings of Escape sequences in which they occur. The meaning of the Escape sequences is to be reckoned as if the communication controls had been purged. Of course, this doctrine cannot be applied when the occurrence of a communication control would have the effect of terminating a portion of a message prior to the transmission of the Final character of the affected Escape communication control characters and Data Link Escape sequences and the possibility of their occurring with validity in an Escape sequence at the input to the receiving portion of the system. The occurrence of a communication control character or character sequence formed with DLE which would not validly occur in the data stream should be considered as an error condition. This Appendix does not prescribe the means of recovery from this error condition. For suggested examples of the preceding, see Table A1.

Table A1
Suggested Validity of Communication Control Characters in Sequences

Communication Control Character or Character Sequence	May Validly Occur in an Escape Sequence at a Receiver
SOH	Yes
STX	Yes
ETX	No
EOT	No
ENQ	No
ACK	No
DLE	Yes
NAK	No
ETB	Yes
SYN	Yes
DLE SOH	Yes
DLE STX	Yes
DLE ETX	No
DLE EOT	No
DLE ETB	Yes
DLE =	Yes
DLE 0	No
DLE 1	No
DLE 2	No
DLE 3	No
DLE 4	No
DLE 5	No
DLE 6	No
DLE 7	No
DLE SYN	Yes

A2. Interactions When Operating in Multiprogramming and Multiprocessing Environments

The requirements and restrictions that must be satisfied in either a multiprogramming or multiprocessing environment are not increased over those required in a simple batch environment by the introduction of data that include Escape sequences. These requirements and restrictions include the serial examination of the data to ensure that all Escape sequences are interpreted, and recovery procedures to restart after error take into account the serial dependency of the data. Also, that provision is made that no characters in Escape sequences are skipped or encountered in an incorrect sequence.

The last requirement precludes access being per-

mitted to stored data at any point where the Escape sequence is not recognized, or to magnetic tape using read backward if the Escape sequences were only provided for read forward.

A3. Unique Interchange by Prior Agreements

In order that the code represented by a character string can be unambiguously identified, the character string should be preceded by a designating Escape sequence. However, under certain conditions, this Escape sequence can be omitted as illustrated in the following examples:

(1) If only ASCII or an agreed-upon 8-bit code is used, it is not required to designate these codes.

(2) If ASCII or an ASCII variant is used with a single alternative graphic set, it is not required to use designating Escape sequences.

(3) If interchange is only in another code, such interchange is beyond the scope of this standard. Any necessary conventions are subject to the agreement of the parties to the interchange.

However, it is a recommended practice that the designating Escape sequence be included in every interchange in order to preserve the generality of the application and avoid unnecessary limitations to the arena of interchangeability of the data.

A4. System Startup and Restart

When a link is established or reinitialized, in the absence of any other agreement between or among the exchanging parties, the basic code in effect is assumed to be ASCII (7-bit environment). If the interchange is to be in another code where no prior agreement between or among the exchanging parties exists, appropriate Escape sequences must be used.

A5. Future Work on Code Extension

This Appendix contains additional code extension techniques that have been discussed. This is one approach to the problem of accommodating more than two pages of graphics, if required to do so.

A6. Additional Invokers and Announcers

A6.1 Definitions of Additional Invokers

A6.1.1 ESI. ESI is a mnemonic for the control character Extended Shift-In. This control is used in 8-bit extended codes, and its ESC Fe counterpart is used in some 7-bit extended codes to invoke the last designated G2 graphic set.

A6.1.2 ESO. ESO is a mnemonic for the control character Extended Shift-Out. This control is used in 8-bit extended codes and its ESC Fe counterpart is used in some 7-bit extended codes to invoke the last designated G3 graphic set.

A6.2 Categories. These categories are primarily intended for 8-bit codes, but the facility for their use is available to the application designer or user in 7-bit working where full understanding of the implications exists.

ESC 2/10 F and ESC 2/14 F designate sets of ninety-four graphic characters which will be used as the G2 set. The designated set is invoked by ESI or the Escape sequence ESC 4/15, when appropriate.

A6.3 Future Announcers. See Table A2.

Table A2
Future Announcers

Final Character	Facilities Used
4/5	The G0 set of graphics is to be used. In both 7- and 8-bit environments, the Escape sequence which designates this set also invokes it into columns 2–7. SI and SO are not to be used. The C1 set of controls is to be used also. In an 8-bit environment, individual control characters of the C1 set of controls will be represented by ESC Fe sequences and not by the bit patterns of columns 08 and 09. Thus, in an 8-bit environment, columns 08–15 are not used.
4/6	The G0 and G1 sets of graphics are to be used. In both 7- and 8-bit environments, SI invokes G0 into columns 2–7 and SO invokes G1 into columns 2–7. The C1 set of controls is to be used also. In an 8-bit environment, individual control characters of the C1 set of controls will be represented by ESC Fe sequences and not by the bit patterns of columns 08 and 09. Thus, in an 8-bit environment, columns 08–15 are not used.

NOTE: Additional announcers would be required to cover G2 and G3 sets of graphics.

Appendix B
Illustration of Implementation

B1. General

The implementation of the extension of the 7-bit code, described in this standard, depends on the practical need of the implementor or user of the code extension. Implementations may differ in the following aspects and still remain in accordance with this standard:

(1) Including all or parts of the control sets and graphic sets

(2) Identifying the sets by Escape sequences, or having an agreement

(3) Dropping the superfluous invokers or not

(4) Examining the structure of an Escape sequence or not

(5) Handling the character DEL

(6) Combinations of the aforementioned

In this Appendix there are shown a few of the possible implementations that may help the user to a better understanding. A summary of Fig. B1 through B8 is given in Table B1. There are three types of figures: structure, interpretation, and transformation.

Table B2 shows the symbols used in the flowcharts (see Fig. B4 through B8) and their associated meaning. Table B3 shows the items used in the flowcharts, their meaning, and the possible values.

B2. Structure

The pictorial representations of the structure of the 7-bit and 8-bit code extension (see Fig. B1 through B3) have all the same format as follows:

(1) At the top of the figure are the repertories of single controls and sets of controls

(2) Below that are the Escape sequences that are used for the designation and invocation of the control sets as the C0 and C1 set

(3) At the bottom of the figure is the repertory of graphic sets

(4) Above that are the Escape sequences used for designation of the graphic set as the G0 or G1 set

(5) Above that are the shift characters that are used for the invocation of the G0 or G1 sets

(6) All this comes together in the center to form the 7-bit set or 8-bit set in use

B3. Interpretation

The interpretation of data containing 7-bit or 8-bit code extension is shown by means of flowcharts (see Fig. B4 through B6). The interpretation is separate from the transformation.

B4. Transformation

The transformation of the data from 7-bit to 8-bit and from 8-bit to 7-bit is shown by means of flowcharts (see Fig. B7 and B8). The transformation is separate from the interpretation.

Table B1
Index to Figures of Appendix B

		Figures			
				Transformation	
Included Sets	Bits	Structure	Interpretation	7 to 8	8 to 7
C0, C1, G0, G1	7	B1	B4	B7	B8
C0, C1, G0, G1	8	B2	B5	–	–
C0, C1, G0, G1	7 and 8	B3	B6	–	–

Table B2
Meaning of Flowchart Symbols

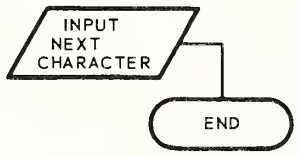
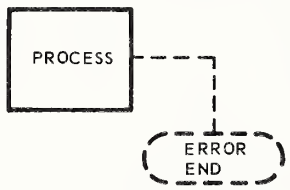
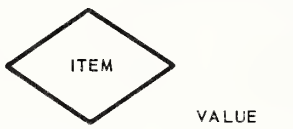
SYMBOL	NAME	MEANING
	INPUT	TO FETCH THE NEXT CHARACTER FROM THE INPUT CHARACTER STRING AND GO TO THE END IF THERE IS NO MORE CHARACTER AVAILABLE.
	PROCESS	TO EXECUTE THE DEFINED PROCESS. THERE MAY BE AN ERROR END BECAUSE OF LACK OF MORE CHARACTERS.
	DECISION	TO EXAMINE THE ITEM OF HAVING THE SPECIFIED VALUE.
	PREPARATION	TO SET THE ITEM TO THE SPECIFIED VALUE.
	OUTPUT	TO TAKE THE LAST CHARACTER (OUTPUT CHARACTER) OR THE SPECIFIED CHARACTER(S) OUTPUT 1/11, 4/14 OR THE MANIPULATED CHARACTER OUTPUT o8 DROPPED TO THE OUTPUT CHARACTER STRING.
	PREDEFINED PROCESS	TO EXECUTE THE APPROPRIATE PREDEFINED PROCESS
	CONNECTOR	ENTRY FROM, OR EXIT TO, ANOTHER PART OF THE FIGURE.

Table B3
Items Used in the Flowcharts

ITEM	MEANING	POSSIBLE VALUES	
		IN 7 BITS	IN 8 BITS
CHARACTER	THE BIT PATTERN OF THE LAST CHARACTER	0/0 TO 7/15	0/0 TO 15/15
COLUMN	THE COLUMN NUMBER OF THE LAST CHARACTER	0 TO 7	0 TO 15
STATUS 7	SHIFT-STATUS OF THE 7-BIT CODE	S1 S0 UNDEFINED	
STATUS LEFT	SHIFT-STATUS OF THE LEFT PART OF THE 8-BIT CODE		S1 UNDEFINED
STATUS RIGHT	SHIFT-STATUS OF THE RIGHT PART OF THE 8-BIT CODE		S0 UNDEFINED
BITS	NUMBER OF THE VALID BITS OF THE CHARACTER		7 8 UNDEFINED

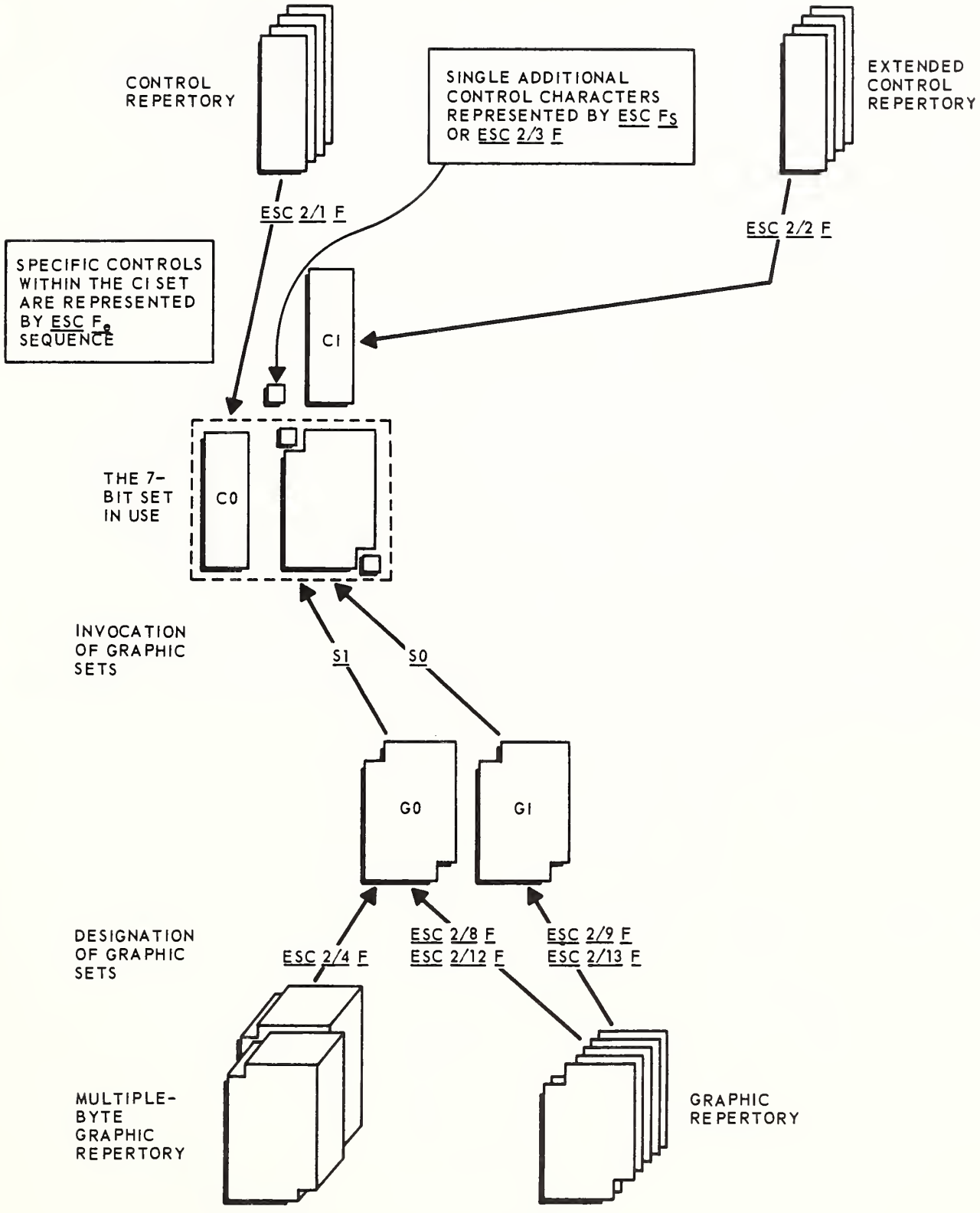


Fig. B1
7-Bit Code Extension Structure Using C0, C1, G0, and G1 Sets

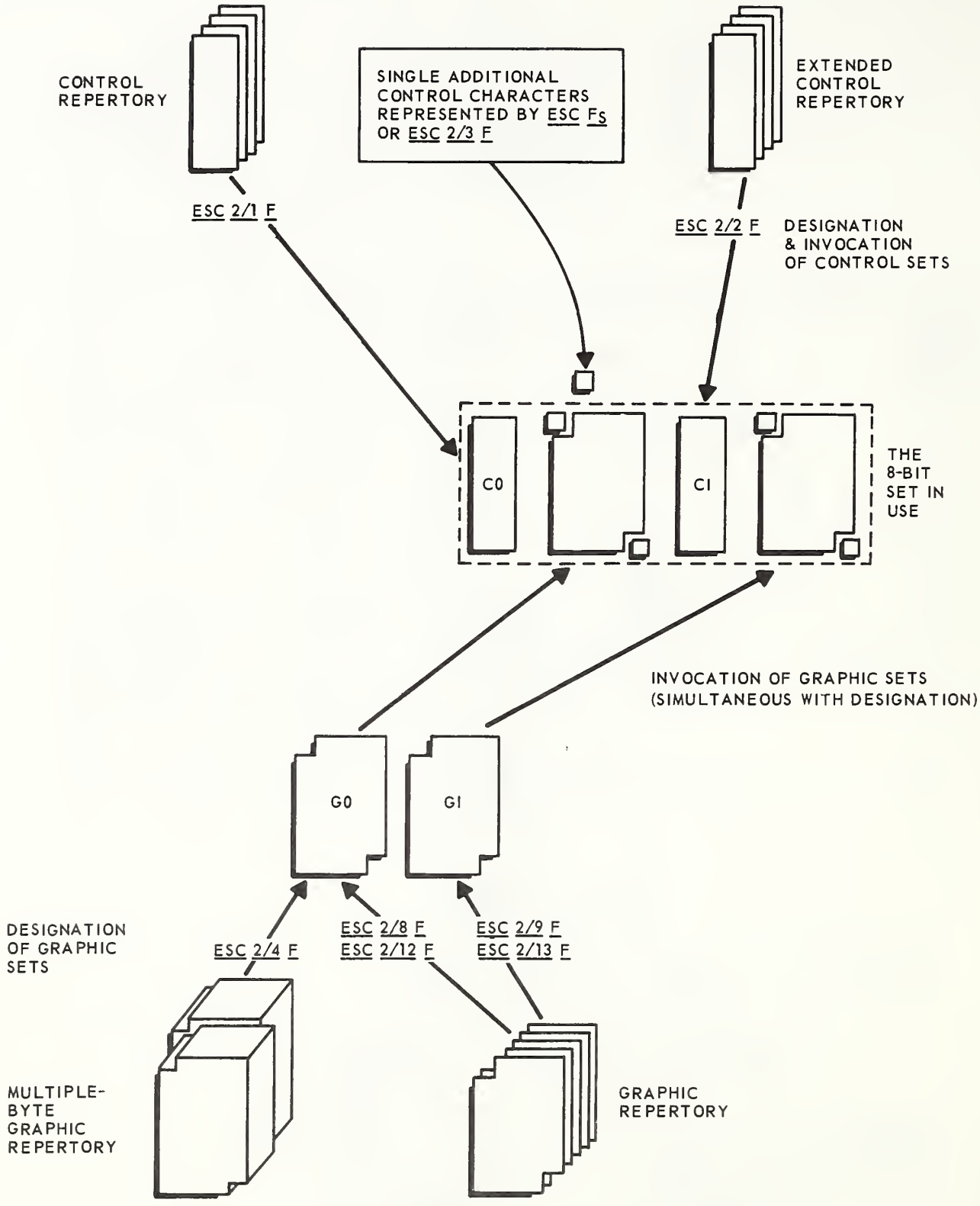


Fig. B2
8-Bit Code Extension Structure Using C0, C1, G0, and G1 Sets

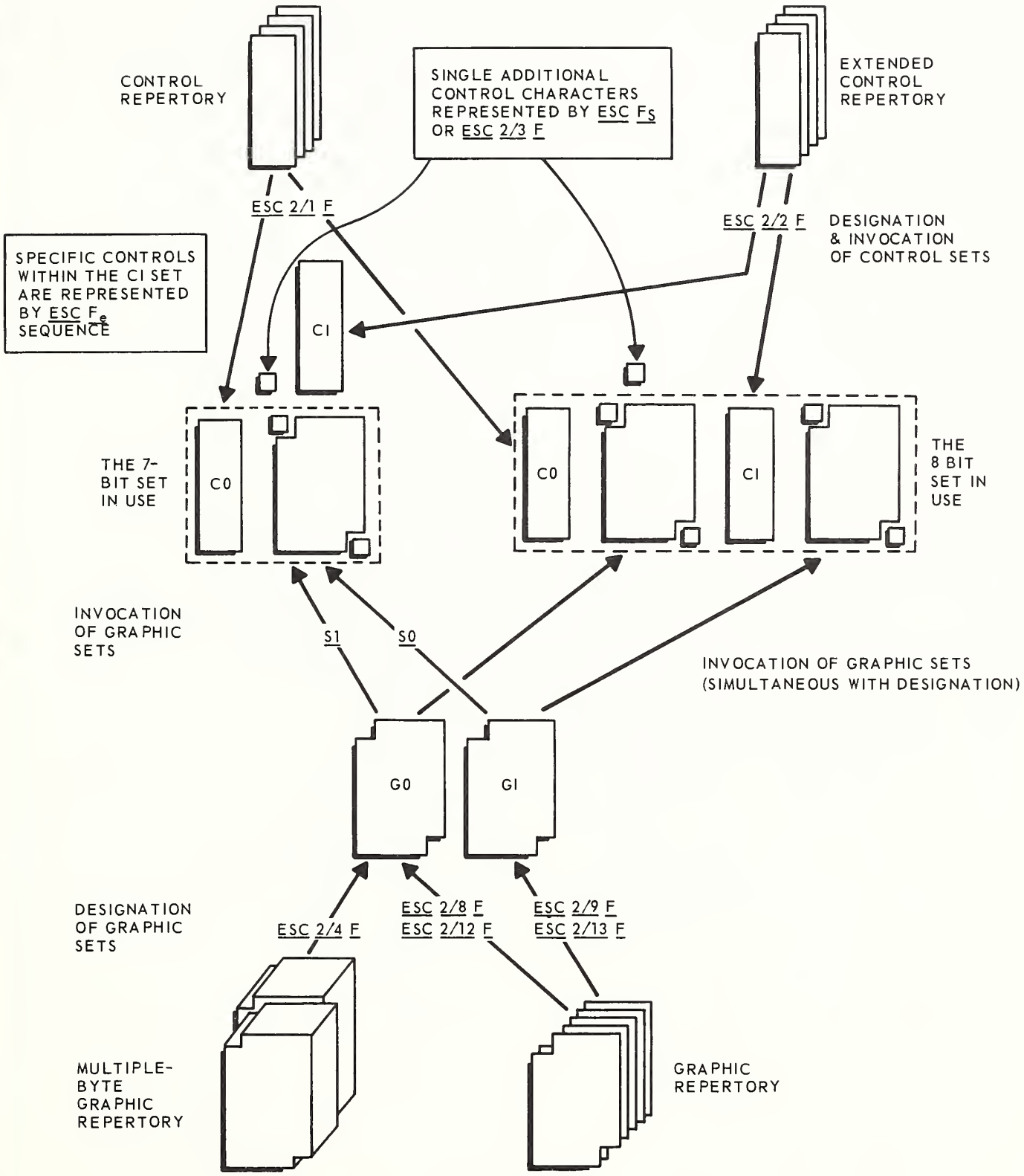


Fig. B3
Composite 7-Bit and 8-Bit Code Extension Structure
Using C0, C1, G0, and G1 Sets

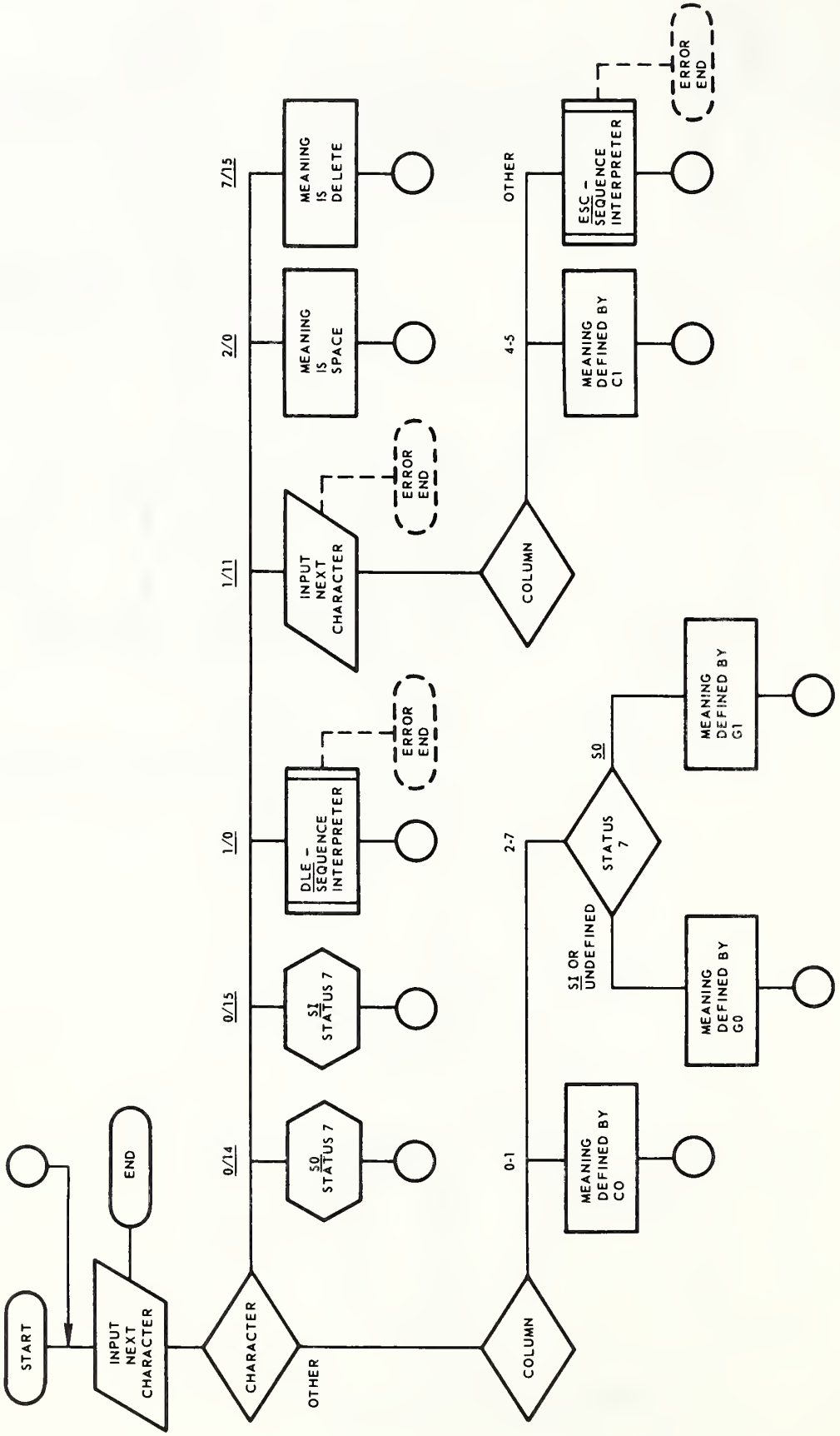


Fig. B4
Interpretation of 7-Bit Data with C0, C1, G0, and G1 Sets

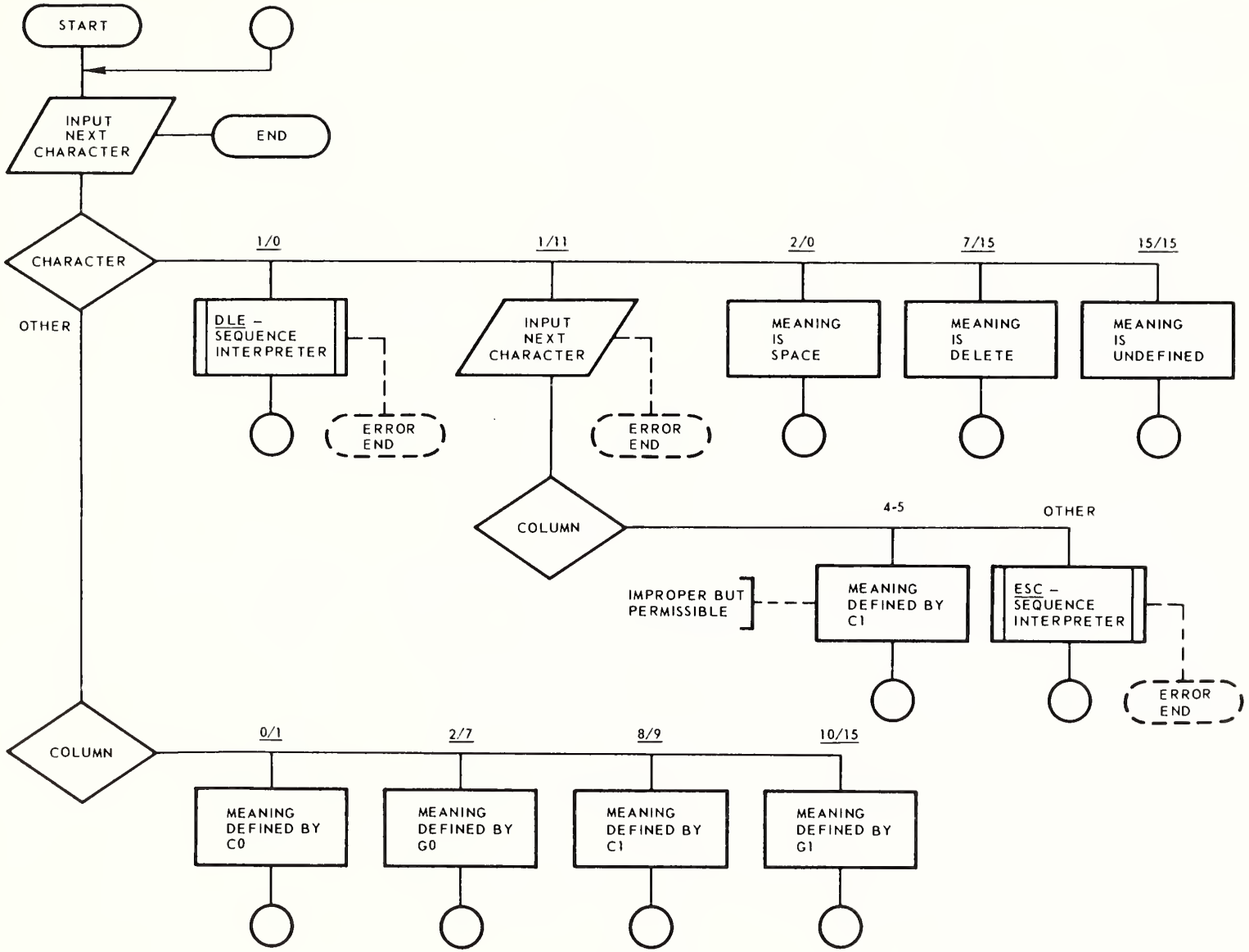


Fig. B5
Interpretation of 8-Bit Data with C0, C1, G0, and G1 Sets

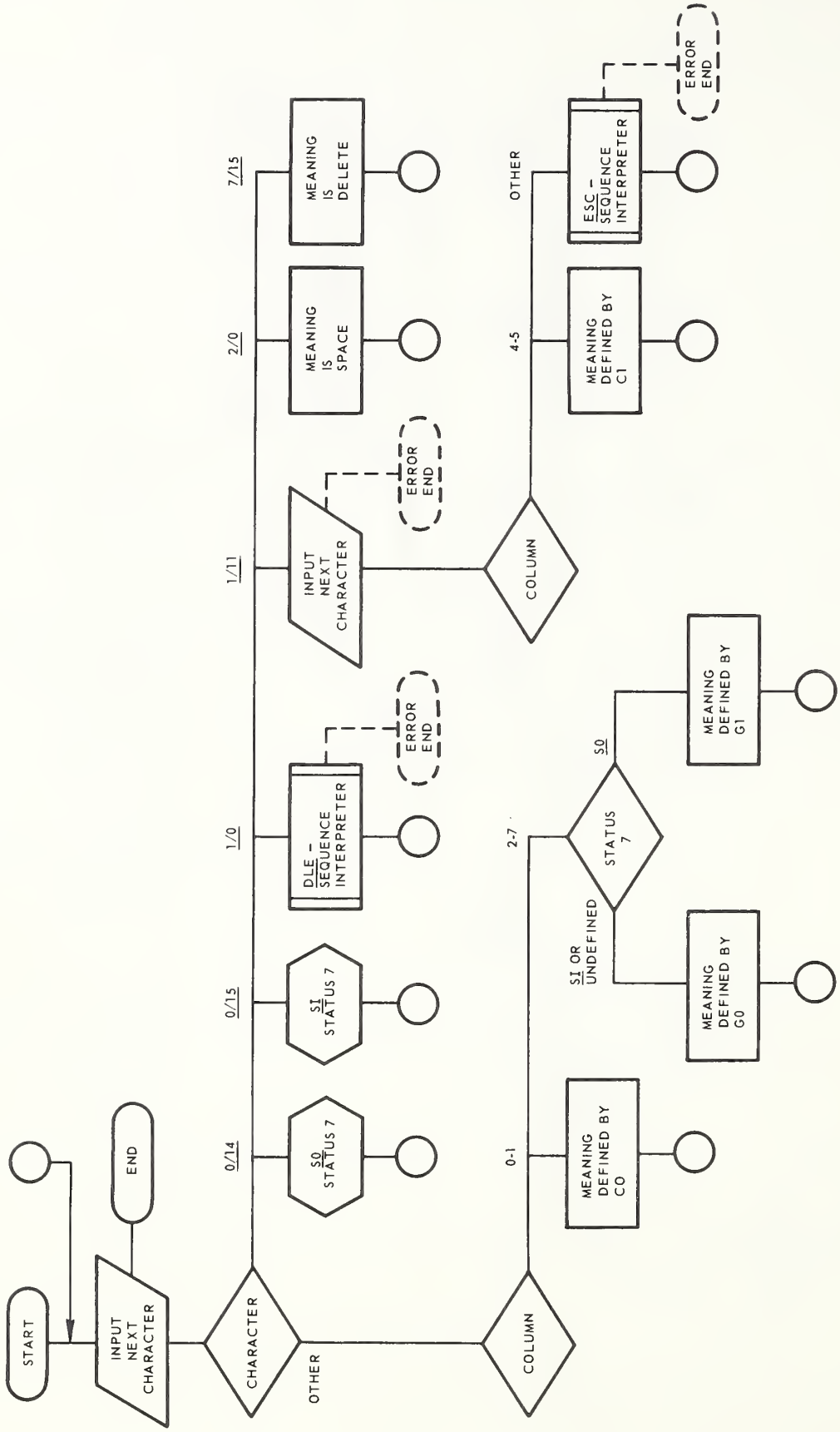


Fig. B6
Interpretation of 7-Bit and 8-Bit Data with C0, C1, G0, and G1 Sets

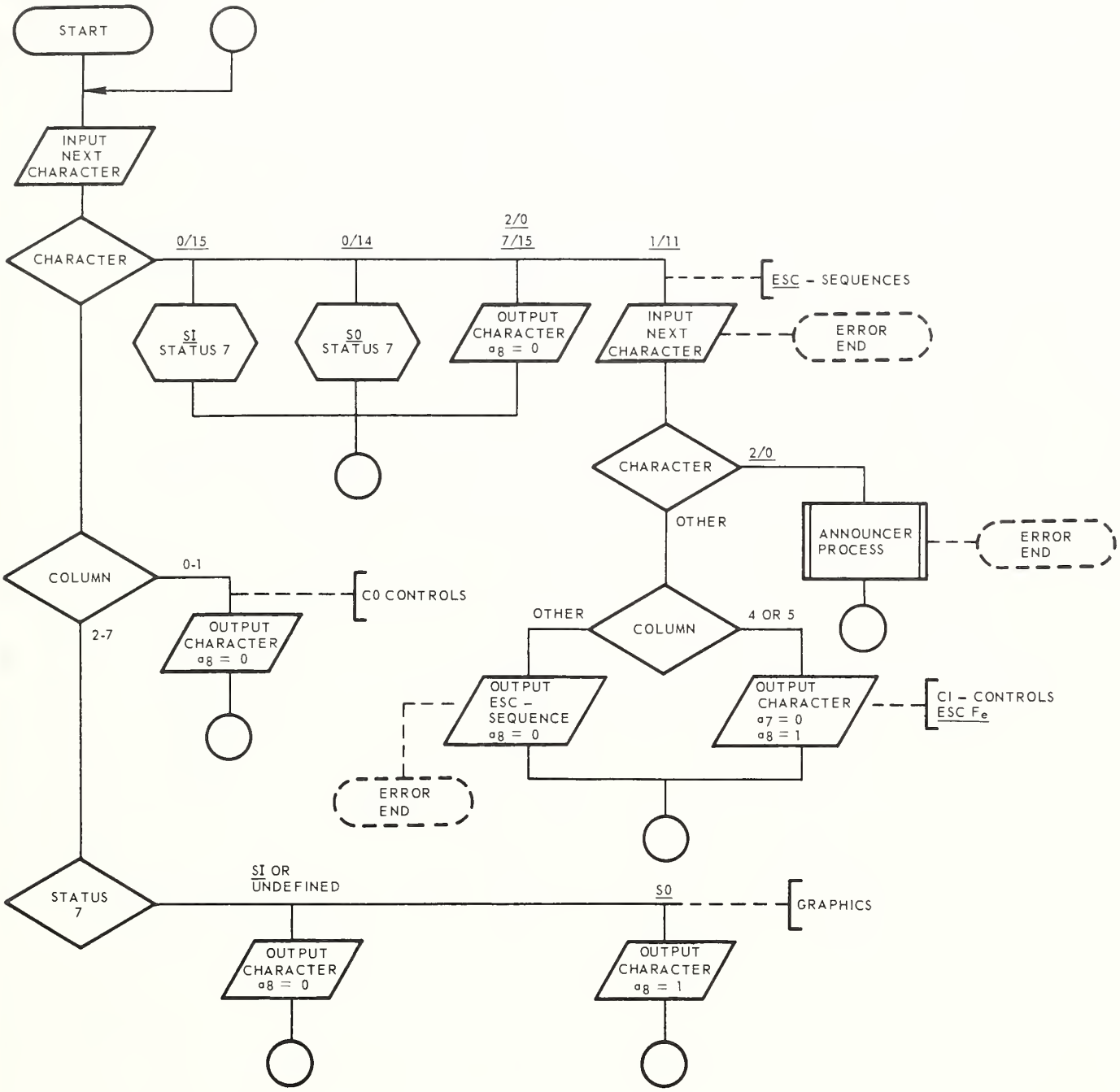


Fig. B7
Transformation from 7 Bits to 8 Bits Including C0, C1, G0, and G1 Sets

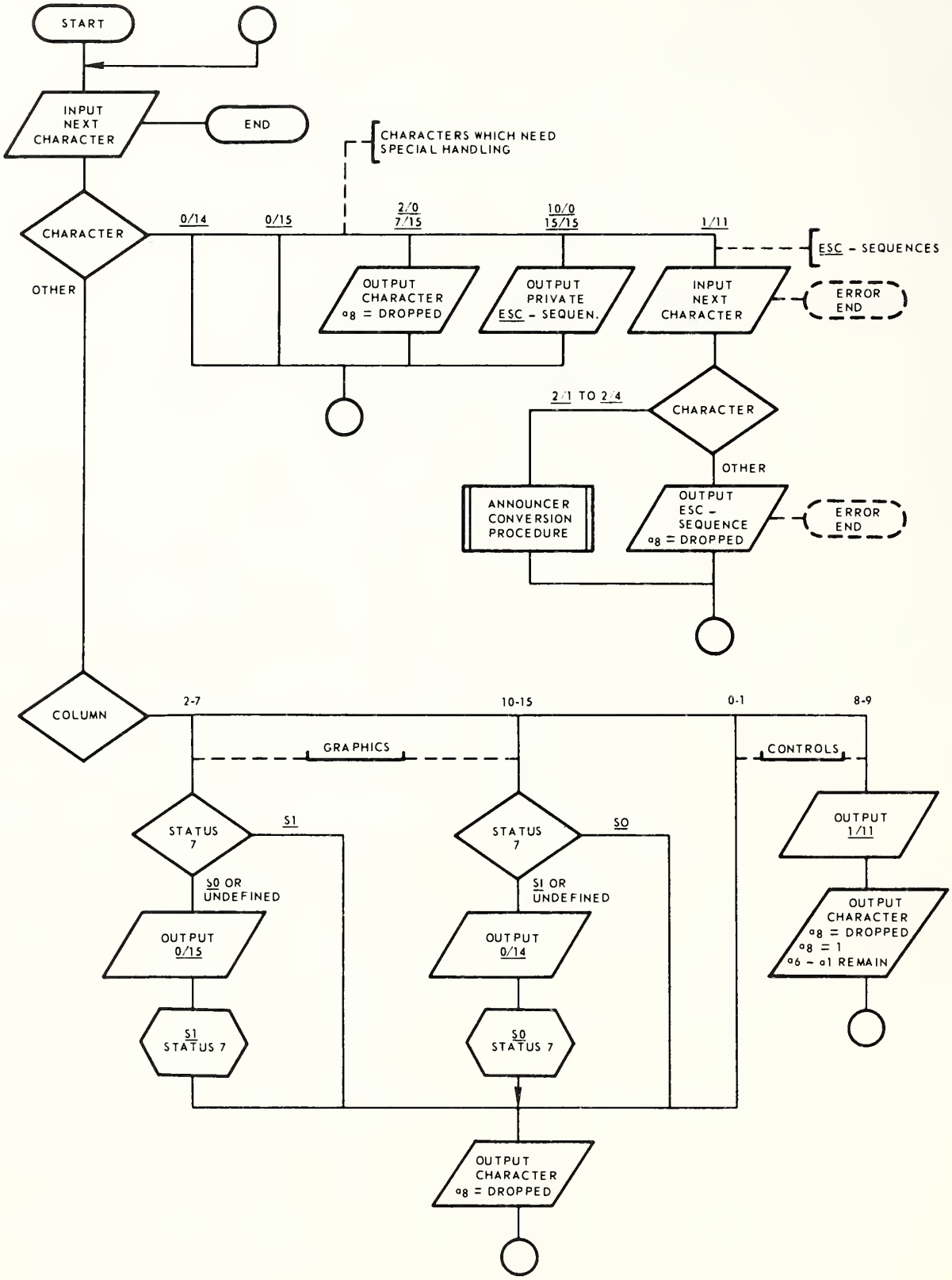


Fig. B8
Transformation from 8 Bits to 7 Bits Including C0, C1, G0, and G1 Sets

X3.115-1984 Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)

X3.117-1984 Printable/Image Areas for Text and Facsimile Communication Equipment

X3.118-1984 Financial Services — Personal Identification Number — PIN Pad

X3.119-1984 Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter

X3.120-1984 Contact Start/Stop Storage Disk

X3.121-1985 Two-Sided, Double-Density, Unformatted 5.25-inch (130-mm), 48-tpi (1,9-tpmm), Flexible Disk Cartridge for 7958 bpr Use

X11.1-1977 Programming Language MUMPS

IEEE 416-1978 Abbreviated Test Language for All Systems (ATLAS)

IEEE 716-1982 Standard C/ATLAS Language

IEEE 717-1982 Standard C/ATLAS Syntax

IEEE 770X3.97-1983 Programming Language PASCAL

IEEE 771-1980 Guide to the Use of ATLAS

MIL-STD-1815A-1983 Reference Manual for the Ada Programming Language

---

X3/TRI-82 Dictionary for Information Processing Systems (Technical Report)

American National Standards for Information Processing

X3.1-1976 Synchronous Signaling Rates for Data Transmission
X3.2-1970 Print Specifications for Magnetic Ink Character Recognition
X3.4-1977 Code for Information Interchange
X3.5-1970 Flowchart Symbols and Their Usage
X3.6-1965 Perforated Tape Code
X3.9-1978 Programming Language FORTRAN
X3.11-1969 General Purpose Paper Cards
X3.14-1983 Recorded Magnetic Tape (200 CPI, NRZI)
X3.15-1976 Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission
X3.16-1976 Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.17-1981 Character Set for Optical Character Recognition (OCR-A)
X3.18-1974 One-Inch Perforated Paper Tape
X3.19-1974 Eleven-Sixteenths-Inch Perforated Paper Tape
X3.20-1967 Take-Up Reels for One-Inch Perforated Tape
X3.21-1967 Rectangular Holes in Twelve-Row Punched Cards
X3.22-1983 Recorded Magnetic Tape (800 CPI, NRZI)
X3.23-1974 Programming Language COBOL
X3.25-1976 Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.26-1980 Hollerith Punched Card Code
X3.27-1978 Magnetic Tape Labels and File Structure
X3.28-1976 Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links
X3.29-1971 Specifications for Properties of Unpunched Oiled Paper Perforator Tape
X3.30-1971 Representation for Calendar Date and Ordinal Date
X3.31-1973 Structure for the Identification of the Counties of the United States
X3.32-1973 Graphic Representation of the Control Characters of American National Standard Code for Information Interchange
X3.34-1972 Interchange Rolls of Perforated Tape
X3.36-1975 Synchronous High-Speed Data Signaling Rates between Data Terminal Equipment and Data Communication Equipment
X3.37-1980 Programming Language APT
X3.38-1972 Identification of States of the United States (Including the District of Columbia)
X3.39-1973 Recorded Magnetic Tape (1600 CPI, PE)
X3.40-1983 Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)
X3.41-1974 Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange
X3.42-1975 Representation of Numeric Values in Character Strings
X3.43-1977 Representations of Local Time of the Day
X3.44-1974 Determination of the Performance of Data Communication Systems
X3.45-1982 Character Set for Handprinting
X3.46-1974 Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)
X3.47-1977 Structure for the Identification of Named Populated Places and Related Entities of the States of the United States for Information Interchange
X3.48-1977 Magnetic Tape Cassettes (3.810-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)
X3.49-1975 Character Set for Optical Character Recognition (OCR-B)
X3.50-1976 Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets
X3.51-1975 Representations of Universal Time, Local Time Differentials, and United States Time Zone References
X3.52-1976 Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)
X3.53-1976 Programming Language PL/I
X3.54-1976 Recorded Magnetic Tape (6250 CPI, Group Coded Recording)
X3.55-1982 Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase encoded
X3.56-1977 Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.57-1977 Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control
X3.58-1977 Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)
X3.59-1981 Magnetic Tape Cassettes, Dual Track Complementary Return-to-Bias (CRB) Four-States Recording on 3.81-mm (0.150-Inch) Tape
X3.60-1978 Programming Language Minimal BASIC
X3.61-1978 Representation of Geographic Point Locations
X3.62-1979 Paper Used in Optical Character Recognition (OCR) Systems
X3.63-1981 Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)
X3.64-1979 Additional Controls for Use with American National Standard Code for Information Interchange
X3.66-1979 Advanced Data Communication Control Procedures (ADCCP)
X3.72-1981 Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.73-1980 Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)
X3.74-1981 Programming Language PL/I, General-Purpose Subset
X3.76-1981 Unformatted Single-Disk Cartridge (Top Loading, 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)
X3.77-1980 Representation of Pocket Select Characters
X3.78-1981 Representation of Vertical Carriage Positioning Characters in Information Interchange
X3.79-1981 Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures
X3.80-1981 Interfaces between Flexible Disk Cartridge Drives and Their Host Controllers
X3.82-1980 One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)
X3.83-1980 ANSI Sponsorship Procedures for ISO Registration According to ISO 2375
X3.84-1981 Unformatted Twelve-Disk Pack (200 Megabytes) (General, Physical, and Magnetic Requirements)
X3.85-1981 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge
X3.86-1980 Optical Character Recognition (OCR) Inks
X3.88-1981 Computer Program Abstracts
X3.89-1981 Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)
X3.91M-1982 Storage Module Interfaces
X3.92-1981 Data Encryption Algorithm
X3.93M-1981 OCR Character Positioning
X3.95-1982 Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters
X3.96-1983 Continuous Business Forms (Single-Part)
X3.98-1983 Text Information Interchange in Page Image Format (PIF)
X3.99-1983 Print Quality Guideline for Optical Character Recognition (OCR)
X3.100-1983 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network
X3.101-1984 Interfaces Between Rigid Disk Drive(s) and Host(s)
X3.102-1983 Data Communication Systems and Services — User-Oriented Performance Parameters
X3.103-1983 Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in)
X3.104-1983 Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded
X3.105-1983 Data Link Encryption
X3.106-1983 Modes of Operation for the Data Encryption Algorithm
X3.110-1983 Videotex/Teletext Presentation Level Protocol Syntax
X3.112-1984 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk
X3.114-1984 Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983

(continued on reverse)