ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

ECMA STANDARD

for

RULES FOR THE DEFINITION OF 4 BIT SETS DERIVED FROM THE ECMA 7 BIT CODED CHARACTER SET FOR INFORMATION INTERCHANGE

November 1967

Free copies of this standard **ECMA-14** are available from
ECMA European Computer Manufacturers Association
114, Rue du Rhône - 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

ECMA STANDARD

for

RULES FOR THE DEFINITION OF 4 BIT SETS DERIVED FROM THE ECMA 7 BIT CODED CHARACTER SET FOR INFORMATION INTERCHANGE

November 1967

BRIEF HISTORY

At a request of the Union Internationale des Chemins de Fer (UIC) in June 1965 ECMA TC1 started to consider the problem of defining 4 bit sets derived from the 7 bit coded character set for information processing (Standard ECMA-6). ISO/TC97/SC2 also requested ECMA in May 1966 to submit a contribution on this subject. The document presented (97/2/N221) was unanimously accepted at the March 1967 meeting of SC2 and will be processed as an ISO Draft Recommendation.

Following companies participated in the work of the Task Group which prepared the final document :

Compagnie des Machines Bull
N.V. Electrologica
IBM-WTEC
ICT, International Computers and Tabulators Ltd
ITT Europe Inc.
NCR, The National Cash Register Company Ltd

In addition the Committee has collaborated with following organizations :

International Organization for Standardization (ISO)

Union Internationale des Chemins de Fer (UIC)

Adopted by the General Assembly of ECMA on Nov. 30, 1967, as Standard ECMA-14.

TABLE of CONTENT

1. INTRODUCTION

The need has been expressed for 4 bit character sets of the 7 bit coded character set for information interchange (Standard ECMA-6). Packed numerics as currently used do not form a set in the sense of these Rules. There are economic reasons which require a 4 bit set for private or regional organizations with multiple terminals where the use of the full 7 bit code is not justified.

2. PURPOSE

The multiplicity and the diversity of the possible applications make it impossible to define a unique set or even a family of a limited number of sets. The purpose of the present Standard is therefore to set up Rules for the definition of such sets so that

(a) a given selection of characters from the 7 bit code table shall be uniquely arranged in a 4 bit set and

(b) the sets are as similar to and as compatible with each other as possible.

The Rules given should make conversion possible at minimum cost to and from 7 bit coded information.

3. SCOPE

The present Standard applies to the definition of 4 bit character sets derived from the ECMA 7 bit coded character set for information interchange (Standard ECMA-6).

ECMA 7 BIT CODED CHARACTER SET FOR INFORMATION INTERCHANGE

Bits b_7	b_6	b_5	b_4	b_3	b_2	b_1	Row \ Column	0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
								0	1	2	3	4	5	6	7
0	0	0	0	0			0	NUL	(TC_7)DLE	SP	0	(@) ③	P	` ④	p
0	0	0	0	1			1	(TC_1) SOH	DC_1	!	1	A	Q	a	q
0	0	1	0	0			2	(TC_2) STX	DC_2	" ⑥	2	B	R	b	r
0	0	1	1	1			3	(TC_3) ETX	DC_3	£ ② ⑦	3	C	S	c	s
0	1	0	0	0			4	(TC_4) EOT	DC_4	$ ②	4	D	T	d	t
0	1	0	1	0			5	(TC_5) ENQ	(TC)^{8}NAK	%	5	E	U	e	u
0	1	1	0	0			6	(TC_6) ACK	(TC_9)SYN	&	6	F	V	f	v
0	1	1	1	1			7	BEL	(TC_{10})ETB	' ⑥	7	G	W	g	w
1	0	0	0	0			8	FE_0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	0			9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	0			10	FE_2 (LF) ①	SUB	*	: ⑧	J	Z	j	z
1	0	1	1	0			11	FE_3 (VT)	ESC	+	; ⑧	K	([) ③	k	③
1	1	0	0	0			12	FE_4 (FF)	IS_4 (FS)	,	<	L	③	l	③
1	1	0	1	0			13	FE_5 (CR) ①	IS_3 (GS)	–	=	M	(]) ③	m	③
1	1	1	0	0			14	SO	IS_2 (RS)	.	>	N	^ ④ ⑥	n	¯ ④ ⑤
1	1	1	1	0			15	SI	IS_1 (US)	/	?	O	_	o	DEL

4. CHARACTER SET

The characters available to a 4 bit set are :

(a) all the characters of the 7 bit code table

(b) other characters whose meaning and graphical representation must be agreed between the sender and recipient of the data.

5. DEFINITION OF A 4 BIT CHARACTER SET

5.1 A 4 bit set comprises the ten numerals 0 to 9 and six further characters allocated as described in section 6.

5.2 These characters are coded in increasing binary order and referenced Position 0 to Position 15, the numerals being allocated to positions 0 to 9.

Position	Character	Coded Representation
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10		1010
11		1011
12		1100
13		1101
14		1110
15		1111

6. RULES FOR THE ALLOCATION OF CHARACTERS TO POSITION 10 TO 15

6.1 Decide which characters from the 7 bit code table are required for the 4 bit set.

NOTE : It is important that no attempt be made at this stage to foresee the graphical representation of characters from columns 0 and 1 of the 7 bit code table.

6.2 Consider any characters required from row 10 of the 7 bit code table.

6.3 Allocate to position 10 the first one encountered when taken in the order of columns: 3 2 1 0 7 6 5 4.

6.4 Leave any remaining characters in row 10 to be dealt with under 6.6.

6.5 Repeat the procedure of 6.2 to 6.4 for each of the rows 11 to 15.

6.6 Sort any character left from rows 10 to 15 together with any other characters required from row 0 to 9 (other than numerals) into ascending order of their 7 bit binary coded representation and insert them into unfilled positions of the set in order of position number.

6.7 If a control (i.e. a character from column 0 to 1 of the 7 bit code table) has been used, decide whether a graphical representation of it is required. If so, allocate either the graphic from column 3 of the 7 bit code corresponding to the position the control now occupies in the 4 bit set, or a graphic not included in the 7 bit code.

6.8 Allocate any required characters not included in the 7 bit code, to the remaining vacant positions. If a graphical representation is required of such a character, allocate either the graphic from column 3 of the 7 bit code corresponding to the position which the character occupies in the 4 bit set, or a graphic not included in the 7 bit code.

Note : Recommendations on the Selection of Characters

(a) It is recommended that the non-numeric characters of the 4 bit set be chosen from the 7 bit code, even though other characters are not prohibited.

(b) When the graphical representation is unimportant to the application it is particularly recommended to choose the characters in positions 10 - 15 of column 3 of the 7 bit code table to represent the controls allocated to these positions. However, these graphics must not be used to represent another graphic of the 7 bit code.

(c) Every effort should be made to use one but no more than one character from each of rows 10 to 15 of the 7 bit code table.

APPENDIX I

EXAMPLES OF 4 BIT SETS

1. To demonstrate the method of allocating characters to positions 10 to 15 of a 4 bit set the following two examples are given :

1.1 Example 1

All these characters are comprised in the 7 bit code table.

Rule 6.1 : Characters required are: Asterisk, Equals, Line Feed, Minus, Plus and Space.

Rules 6.2 to 6.5: These allocate
* ⟶ Position 10
+ ⟶ Position 11
= ⟶ Position 13

Rule 6.6 : The binary values of the remaining characters are :

Line Feed	0001010
Minus	0101101
Space	0100000

They therefore are allotted as follows:

New Line ⟶ Position 12
Space ⟶ Position 14
Minus ⟶ Position 15

Rule 6.7 : We may now choose to represent New Line by < but by no other graphic from the 7 bit code.

1.2 Example 2

Rule 6.1 : The characters required from the 7 bit code table are: Acknowledge, Cancel, Start of Data Block, End of Data Block, Line Feed and Space.

Rules 6.2 to 6.5: These allocate Line Feed ⟶ Position 10

Rule 6.6 : Binary values of the remaining characters of the 7 bit code are :

Acknowledge	0000110
Cancel	0011000
Space	0100000

They are therefore allotted as follows:

Acknowledge ⟶ Position 11
Cancel ⟶ Position 12
Space ⟶ Position 13

Rule 6.7 : This permits to allocate the following graphics to 7 bit code controls:

Line Feed :
Acknowledge ;
Cancel <

Rule 6.8 : Start of Data Block and End of Data Block are allocated positions 14 and 15. They may be represented by > and ? respectively.

2. A few examples of 4 bit sets

Position Number	Example 1	Example 2	Example 3
0 - 9	0 - 9	0 - 9	0 - 9
10	:	*	NL
11	;	+	+
12	<	,	SP
13	=	-	-
14	>	.	.
15	?	/	DEL

Example 4	Example 5	Example 5	Example 6
0 - 9	0 - 9	0 - 9	0 - 9
*	NL	NL	NL(:)
+	;	ENQ(;)	SP
NL	SP	SP	Progr. Start (<)
=	-	-	Tape End (=)
SP	EOB (>)	EOB (>)	End of Data Block
-	DESTROY	ERROR	DEL

These examples illustrate two things :

(a) A character from the 7 bit code can occupy different positions in different 4 bit sets. Thus the minus sign in examples 3 and 4 and 'space' in examples 3,4,5 and 6 are differently placed. This is regrettable but unavoidable.

(b) A graphic can be used to represent different things quite apart from its intrinsic meaning (e.g. < , and > in examples 5 and 6).

Both examples 5 are suitable for UIC.

Example 6 represents a possible machine tool application.

APPENDIX II

BACKGROUND AND DEVELOPMENT OF THE RULES

II.1 Need for 4 Bit Character Sets

The need has been expressed for 4 bit sets derived from the 7 bit code, comprising 0 to 9 and some additional characters. There are economic reasons which require a 4 bit set (or 4 information bits plus parity) for private and regional organizations with multiple terminals where the full 7 bit code is not justified. Nevertheless, such systems require a logical relationship to the 7 bit code since they must exchange information with general purpose systems using the 7 bit code. Examples of such areas of application which have become prominent are numeric control and data collection with particular field of application such as those described by UIC (Union Internationale des Chemins de Fer). Many of the potential fields of application require data to pass across national boundaries which makes it desirable to reach an international understanding.

The purpose of this 4 bit standardization is not to set up another code for interchange between computers; that is recognized to be the 7 bit code according to Standard ECMA-6. It is to permit simple closed systems to communicate with each other at the four bit level in such a way that when conversion to 7 bit is necessary, it can be achieved easily and with the minimum of cost.

II.2 Media Considerations

Examples of the media involved are punched tape, communication channels and certain classes of printed documents. The devices likely to be involved in such applications will probably require a minimum keyboard (to reduce cost). Where punched tape and/or communication channels are involved, a parity bit may be added to provide a check against malfunctioning of the terminal device and for simple protection of the transmission links. In order to allow the use of the all zeros combination in the four data bits, it would be essential to use odd parity to distinguish it from unpunched tape.

II.3 Character Sets Required

Consideration of the different 4 bit sets desired shows that other characters besides the numerals are required

by a number of different applications. For example, Space and New Line are commonly required controls and - (minus) and . (full stop) common graphics. They are however not universally needed. Sometimes controls or graphics not in the 7 bit code table are required, such as programmed start in numeric control or the abstract symbols specifically wanted for character recognition. There are indeed so many different requirements that no fixed standardization of a unique set is possible. Consequently it is considered necessary to make provision for a user to select any combination of the non-numeric characters from the 7 bit code for inclusion in positions 10 - 15 of the 4 bit set. Furthermore, the standard should define a unique way of allotting the selected characters to these positions. Naturally the resulting 4 bit sets should be as similar and compatible as the other requirements will permit.

II.4 Conversion Philosophy

Since the essence of a 4 bit system is low cost, and it is likely to be used with a multiplicity of terminals, it is clearly undesirable to require each terminal to provide elaborate conversion into 7 bit form. It would be convenient if such converters were to merely add three bits, in a simple uniform manner solely to convert the 4 bit form into one acceptable to the 7 bit medium.

This would only yield a true 7 bit code representation if the 4 bit set consisted of column 3 of the 7 bit code table. In view of the diverse character requirements this would not normally be so but it would not matter if the recipient of the information were aware of its source. He would be aware of the special meaning of the sixteen 7 bit combinations (011 XXXX). Alternatively, if interchange of information of the standard 7 bit code were required, conversion from the pseudo 7 bit form would be needed and would be carried out in the 7 bit medium.

II.5 Recommended 4 Bit Sets

To be considered as a set of the 7 bit code, a 4 bit set should have 10 numerals (0 to 9) coded in binary. Since the numerals 0 - 9 are allocated to column 3 of the 7 bit code table, the conversion between 4 bit and 7 bit numerals requires the systematic addition or deletion of bits 7, 6 and 5 (011). The remaining six characters cannot necessarily be dealt with in such a simple way since they need to be selected from various parts of the 7 bit code table.

The requirements stated in Sections 2 and 3 may be summarized as follows :

(a) Code conversion within the 7 bit environment should be made as simple as possible.

(b) Any combination of non-numeric characters must be available for positions 10 - 15 and a unique way of allocating the selected characters must be defined.

(c) The various 4 bit sets should be as similar and compatible as possible.

As a result of (a) and the requirement to include the numerals, it is most desirable to define the 4 bit representation of a character as the 4 least significant bits of its representation in the 7 bit code (in the same order).

Another way of regarding this is that the position a character occupies within the 4 bit set should ideally be the same as it occupies within its column in the 7 bit set. If this rule were followed faithfully, two characters having the same position within different columns of the 7 bit table (i.e. having the same least significant four bits) could not be included in the same 4 bit set.

For many applications this restriction is not overwhelming, though for others - for example if " Space" is required - it can be. Accordingly the best method of standardization is to set down some rules which are considered binding, together with recommendations which users are asked to obey if at all possible.

Requirements (a) and (c) above lead to the following recommendation. Because of their location in column 3 of the 7 bit table, if any or all of the characters : ; < = > ? are required they must occupy the same position in the 4 bit code as they do in the said column 3.

In some applications characters not included in the 7 bit code are required but the precise graphical representation is unimportant. To prevent unnecessarily arbitrary choices being made, users in this situation are urged to choose graphics from within the said column 3 of the 7 bit code. Thus "Part Number" could be represented by "Colon". Since this meaning is different from the intrinsic meaning of "Colon" it must be agreed between sender and recipient.

It should be emphasized, however, that a graphic in the said column 3 cannot be used to represent another graphic in the 7 bit code. Thus if addition is required, the plus sign must be used, not semicolon.

Requirement (b) is of course in conflict with (c). If the latter were of no importance one might frame the rule for allocating characters to positions 10 to 15 to be determined by their binary order within the 7 bit code. This

would mean, however, that most of the non-numeric characters could occupy any of the six positions, in the various 4 bit sets which would arise.

In order to try to provide some standardization for those applications which can submit to the concomitant restrictions, the rules are in fact a blend of those described in the previous paragraphs. Namely to fill positions 10 - 15 users are encouraged to use the characters in column 3 of the table if possible. If this is not acceptable, they are recommended to use only one character from each row of the 7 bit table. Moreover, provisions are made for the foregoing recommendations to be followed in part, if not universally.

It is realized that there are many applications for which the resulting rules yield sets which for one reason or another could be improved upon. Nevertheless, it was felt that simple although rather arbitrary rules which were easily applied and which went part of the way to standardization were most desirable for the majority.