Published: 03/30/67
(Supersedes: BC.2.06; 11/10/66)

Identification

Multics standard card punch codes
and
Relation between ASCII and EBCDIC
J. H. Saltzer

Purpose

This section defines standard card punch codes to be used in representing ASCII characters for use with Multics. Since the card punch codes are based on the punch codes defined for the IBM EBCDIC standard, automatically a correspondence between the EBCDIC and ASCII character sets is also defined.

Note

The Multics standard card punch codes described in this section are not identical to the currently proposed ASCII punched card code. The ASCII standard code is not supported by any currently available punched card equipment; until such support exists it is not a practical standard for Multics work. The Multics standard card punch code described here is based on widely available card handling equipment used with IBM System/360 computers. The six characters for which the Multics standard card code differs with the ASCII card code are noted in the table below.

Background

The fundamental character set used for symbolic source programs and input/output on Multics is the proposed revised ASA standard, known here as pr(1966)ASCII. This set is described completely and accurately in MSPM section BC.2.01 dated November 10, 1966. Earlier versions of BC.2.01 described earlier versions of this standard, or else pr(1965) ASCII. Pr(1966)ASCII, which we will now refer to simply as ASCII, differs with earlier ASA standards on the graphic representation and code assignments of a very few relatively obscure special characters.* The changes have been made

---

*By way of terminology, the characters are divided into two groups, named graphics and controls (including space); the graphics are further divided into alphabetic (upper and lower), numeric, and specials.

to remain compatible with International Standard Organization recommendations, and to allow IBM more freedom in defining a correspondence between ASCII and EBCDIC.

Similarly, the fundamental character set used for input and output with some devices from a system 360 is the IBM standard, known here as EBCDIC-3(rev). This set is completely and accurately described (except for the slash character erroneously left out of position 01100001) on page 149 of "System 360, Principles of Operation", A22-6821-3, as revised by TNL N22-0232-0.

The EBCDIC character set has similarly gone through several minor transformations as needs for compatibility with ASCII have been recognized. EBCDIC has now become frozen for all practical purposes by the delivery of a large number of system 360 computers and associated input/output devices.

Relation between EBCDIC and ASCII

EBCDIC is an eight-bit code for which graphics have been assigned to 88 code values, controls to 27 code values, and to which card punch codes have been assigned to all 256 possible values. ASCII is a seven bit code with graphics assigned to 94 code values, controls to 34.

Although there are 85 graphics in common between EBCDIC and ASCII, there is no practical algorithm by which one can deduce an EBCDIC code value from the ASCII code value (or vice versa), short of a complete table look up. That is to say, the numerical values of the two codes are more or less completely unrelated.

Graphic Correspondence

On the other hand, since there are so many common graphics, one can define a correspondence between at least the graphic part of the two codes and thereby establish conventions for communication between computers using the codes. Simultaneously, a card punch code for ASCII is defined, as mentioned above, which has the immense practical advantage of equipment available in quantity using these card codes.

In interpreting Table I, it may be helpful to observe that the correspondence between "ASCII code value" in column 1 and "ASCII character meaning" in column 2 is firmly defined by the ASA standard. Similarly, correspondence between "EBCDIC meaning" in column 3, "EBCDIC code value" in column 4, and "EBCDIC card punch code" in column 5 is firmly defined by the IBM standard. This table suggests a correspondence between the first two columns on the one hand, and the last three on the other, based on graphic similarities and other suggestions, as noted.

The suggested graphic correspondence in Table I is derived as follows. 85 ASCII graphic characters correspond directly with identical EBCDIC graphics. Three ASCII graphics are made to correspond with three non-identical EBCDIC graphics as follows:

ASCII	EBCDIC
´ acute accent	' apostrophe
\ left slant	¢ cent sign
^ circumflex	¬ negation

Thus all 88 EBCDIC graphics have an equivalent ASCII graphic.

The remaining six ASCII graphics, namely

left and right square brackets
left and right braces
grave accent
overline (tilde)

have no EBCDIC graphic equivalent. In Table I they are made to correspond to "illegal" EBCDIC codes which, nevertheless, have well-defined card punch code equivalents. The programmer faced with the problem of representing ASCII data in the EBCDIC environment must make some arbitrary decisions if he needs to obtain graphic representation of these

6 characters. (If he is fortunate enough to be working with ASCII data which does not happen to contain these six characters, his problems are minimized.) One appropriate technique is that the suggested "illegal" code be used wherever EBCDIC code representation is required, (e.g., in cards or in core memory) but when printing readable output the illegal codes be printed as escapes or overstrikes.

For example, choosing the cent sign as an escape character, one has the following graphic representation borrowed from Multics conventions.

ASCII graphic	EBCDIC escape representation
{	¢(
}	¢)
~	¢t
`	¢'
]	¢>
[	¢<
\	¢136

The last escape is required in order to insure unambiguous meaning of the cent sign as an escape character.

Alternatively, one can propose a series of overstruck graphics which are more suggestive of the ASCII graphics being represented, e.g.,

ASCII graphic	EBCDIC overstrike representation
{	⊄
}	⊅
	parenthesis over minus
[	∉
]	∌
	parenthesis over equals
`	˙-
	apostrophe over minus
~	¨¬
	double quote over negation.

These two alternatives suggested for printing readable output in an EBCDIC environment are mirrored in the Multics card input conventions (based on card punching with EBCDIC equipment) in which either the multi-column escape sequences described above or single-column multiple punch codes,(with meaningless graphics printed on the card, of course), may be used to represent these characters.

Control Character Correspondence

The 34 ASCII control characters and 27 EBCDIC control characters match in only five cases: space, EOT, null, backspace, and horizontal tab. The remainder have no correspondence which can be expected to work in most cases.

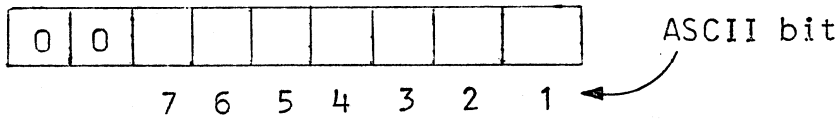
As a result, the programmer transforming character data from one environment to another must study the precise meaning of the control codes in the new environment. For example, some ASCII control codes may logically transform into EBCDIC hardware escape sequences ("Prefix" code followed by some graphic) for some EBCDIC hardware devices. Other controls may not be imitable in the new environment and might instead be printed with graphic escape sequences, or possibly ignored.

Since in general it is difficult to punch the card codes which might correspond to the controls, it should be noted that for Multics input, control codes may be punched as graphic octal escape sequences. Also, the end of a card is interpreted as a "new line" character (changeable under control of other escape sequences--see BC.2.04.)

8-bit Environment

In the System 360 Manual, "Principles of Operation", there is published a code table known as ASCII-8. This table purports to show how the 7-bit ASCII code is represented in an 8-bit environment. It is obtained by taking pr(1966) ASCII, interchanging bits 6 and 7 and duplicating bit 7 as bit 8. This method of representing ASCII in an 8-bit environment is not an ASA standard, but rather an IBM suggestion (resulting from a 9-track tape design problem) which has no official sanction. It is too early yet to determine whether or not the currently proposed ASCII standard for 8-bit environment (bits 1-7 as in pr(1966)ASCII, for representation of ASCII codes in the 9-bit environment

of the GE 645 remains 7-bit codes right-adjusted in 9-bit fields, with leading zeros, as shown below:



Bibliography

1. IBM SRL, "IBM System/360 Principles of Operation," form A22-6821-3, as amended by Technical Newsletter N22-0232-0. Aug. 16, 1966, pp. 149-150

2. "Proposed American Standard: Twelve-Row Punched-Card Code for Information Interchange," Comm. ACM, June 1966, pp. 450-459. Obsolete.

3. "Proposed Revised American Standard Code for Information Interchange," Comm. A.C.M., Apr. 1965, pp. 207-214. Definition of pr(1965) ASCII.

4. "Proposed Revised (1966) ASCII--X3 Action," D. A. Kerr, July 25, 1966 (BTL internal document). Only currently available definition of pr(1966) ASCII.

5. "Proposed American Standard: Recorded Magnetic Tape for Information Interchange (800 CPI, NRZI), Comm. A.C.M., Apr. 1966 pp. 285-292. On p. 286 it is proposed that in an 8-bit environment, the 8th bit be zero.

6. "Proposed American Standard: Twelve-Row Punched-Card Code, "ASA document X3.2/476, not yet published. Latest proposed ASCII standard card code. (replaces published reference 2, above.)

Table I: Correspondence between ASCII Characters and EBCDIC Characters

ASCII Code Value	ASCII Meaning	Corresponding EBCDIC meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
000	(NUL)	NUL	00	9-12-0-8-1	
001	(SOH)	None	01	9-12-1	
002	(STX)	None	02	9-12-2	
003	(ETX)	None	03	9-12-3	
004	(EOT)	EOT	37	9-7	
005	(ENQ)	None	2D	9-0-8-5	
006	(ACK)	None	2E	9-0-8-6	
007	BEL	CU3	2F	9-0-8-7	(Note 3)
010	BS	BS	16	9-11-6	
011	HT	HT	05	9-12-5	
012	NL(LF)	NL	15	9-11-5	(Note 1)
013	VT	None	0B	9-12-8-3	
014	NP(FF)	None	0C	9-12-8-4	
015	(CR)	None	0D	9-12-8-5	
016	RRS(SO)	None	0E	9-12-8-6	
017	BRS(S1)	CU1	0F	9-12-8-7	(Note 3)
020	(DLE)	None	10	12-11-9-8-1	
021	(DC1)	None	11	9-11-1	
022	HLF(DC2)	None	12	9-11-2	

ASCII code values are in octal, EBCDIC code values are in hexadecimal.

ASCII Code Value	ASCII Meaning	Corres-ponding EBCDIC meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
023	(DC3)	TM	13	9-11-3	(Note 3)
024	HLR(DC4)	None	3C	9-8-4	
025	(NAK)	None	3D	9-8-5	
026	(SYN)	None	32	9-2	
027	(ETB)	ECB	26	9-0-6	(Note 3)
030	(CAN)	None	18	9-11-8	
031	(EM)	None	19	9-11-8-1	
032	(SUB)	None	3F	9-8-7	
033	(ESC)	PRE	27	9-0-7	(Note 3)
034	(FS)	None	1C	9-11-8-4	
035	(GS)	None	1D	9-11-8-5	
036	(RS)	None	1E	9-11-8-6	
037	(US)	CU2	1F	9-11-8-7	(Note 3)
040	Space	Space	40	No Punches	
041	!	!	5A	11-8-2	(Note 1)
042	"	"	7F	8-7	
043	#	#	7B	8-3	
044	$	$	5B	11-8-3	
045	%	%	6C	0-8-4	
046	&	&	50	12	

ASCII Code Value	ASCII Meaning	Corres-ponding EBCDIC meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
047	'	'	7D	8-5	Maps ASCII right quote into EBCDIC apostrophe
050	(	(	4D	12-8-5	
051	)	)	5D	11-8-5	
052	*	*	5C	11-8-4	
053	+	+	4E	12-8-6	
054	,	,	6B	0-8-3	
055	-	-	60	11	
056	.	.	4B	12-8-3	
057	/	/	61	0-1	
060	0	0	F0	0	
061	1	1	F1	1	
062	2	2	F2	2	
063	3	3	F3	3	
064	4	4	F4	4	
065	5	5	F5	5	
066	6	6	F6	6	
067	7	7	F7	7	
070	8	8	F8	8	
071	9	9	F9	9	
072	:	:	7A	8-2	

ASCII Code Value	ASCII Meaning	Corres-ponding EBCDIC meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
073	;	;	5E	11-8-6	
074	<	<	4C	12-8-4	
075	=	=	7E	8-6	
076	>	>	6E	0-8-6	
077	?	?	6F	0-8-7	
100	@	@	7C	8-4	
101	A	A	C1	12-1	
102	B	B	C2	12-2	
103	C	C	C3	12-3	
104	D	D	C4	12-4	
105	E	E	C5	12-5	
106	F	F	C6	12-6	
107	G	G	C7	12-7	
110	H	H	C8	12-8	
111	I	I	C9	12-9	
112	J	J	D1	11-1	
113	K	K	D2	11-2	
114	L	L	D3	11-3	
115	M	M	D4	11-4	
116	N	N	D5	11-5	
117	O	O	D6	11-6	

ASCII Code Value	ASCII Meaning	Corresponding EBCDIC Meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
120	P	P	D7	11-7	
121	Q	Q	D8	11-8	
122	R	R	D9	11-9	
123	S	S	E2	0-3	
124	T	T	E3	0-3	
125	U	U	E4	0-4	
126	V	V	E5	0-5	
127	W	W	E6	0-6	
130	X	X	E7	0-7	
131	Y	Y	E8	0-8	
132	Z	Z	E9	0-9	
133	[	None	8D	12-0-8-5	May be punched as ¢< (note 1,note 2)
134	\	¢	4A	12-8-2	Maps ASCII left slant into EBCDIC cent sign. Used as escape (note 1)
135	]	None	9D	12-11-8-5	May be punched as ¢> (note 1, note 2)
136	^	¬	5F	11-8-7	Maps ASCII circumflex onto EBCDIC "negation"
137	-	-	6D	0-8-5	
140	`	None	C0	12-0	May be punched as ¢' (note 2)

ASCII Code Value	ASCII Meaning	Corres-ponding EBCDIC Meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
141	a	a	81	12-0-1	
142	b	b	82	12-0-2	
143	c	c	83	12-0-3	
144	d	d	84	12-0-4	
145	e	e	85	12-0-5	
146	f	f	86	12-0-6	
147	g	g	87	12-0-7	
150	h	h	88	12-0-8	
151	i	i	89	12-0-9	
152	j	j	91	12-11-1	
153	k	k	92	12-11-2	
154	l	l	93	12-11-3	
155	m	m	94	12-11-4	
156	n	n	95	12-11-5	
157	o	o	96	12-11-6	
160	p	p	97	12-11-7	
161	q	q	98	12-11-8	
162	r	r	99	12-11-9	
163	s	s	A2	11-0-2	
164	t	t	A3	11-0-3	

ASCII Code Value	ASCII Meaning	Correspondence EBCDIC Meaning	EBCDIC Code Value	EBCDIC/ Multics Punch Code	Comments
165	u	u	A4	11-0-4	
166	v	v	A5	11-0-5	
167	w	w	A6	11-0-6	
170	x	x	A7	11-0-7	
171	y	y	A8	11-0-8	
172	z	z	A9	11-0-9	
173	{	None	8A	12-0-2	May be punched as ¢((note 2)
174	|	|	4F	12-8-7	(Note 1)
175	}	None	9A	12-11-8-2	May be punched as ¢) (note 2)
176	~	None	D0	11-0	May be punched as ¢t (note 2)

NOTES

Note 1: In the punched card code proposed for ASCII in reference 6, a different card code is used for this character.

Note 2: This graphic does not appear in (or map into any graphic which appears in) the EBCDIC set; it is assigned to an otherwise illegal EBCDIC code value/ card code combination.

Note 3: In some applications, the ASCII meaning of this control character may not correspond to the EBCDIC meaning of the corresponding control character.

Where the Multics meaning of a control character differs with the ASCII meaning, the ASCII meaning is given in parenthesis.