DWG NO

REVISIONS

SYM	DESCRIPTION	APPROVED	DATE
A	Production Release		

J. P. Spencer
Director
Engineering Operations

T. H. Sweere
Director
Systems Engineering

DR	J. Demuzio	
CHK	S. T. Cohen	3/30/71
DSGN	S. T. Cohen	3/30/71
ENGR	E. Ohrenstein	3/30/71
APPD	E. Ohrenstein	3/30/71
APPD	G. Watson	4/7/71

varian data machines /a varian subsidiary
2722 michelson drive / irvine / california / 92664

TITLE

SOFTWARE PERFORMANCE SPECIFICATION
EBCDIC TO HOLLERITH
HOLLERITH TO EBCDIC
EBCDIC TO ASCII CONVERSION ROUTINES for the Varian Data Machines 620 Computer System

CODE IDENT NO.	SIZE	DWG NO.	REV
21101	A	89A0200	A

SCALE

SHEET 1 OF 24

96A0153-000B

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

This conversion subroutine package consists of three subroutines which may be run on any Varian Data Machines 620 series computer. These subroutines enable the user to convert from one character code, usually associated with a particular peripheral device, to the character code of a different device. The three subroutines (described in detail in section 2) are:

EBCDIC TO HOLLERITH CONVERSION –
HOLLERITH TO EBCDIC CONVERSION–
EBCDIC TO ASCII CONVERSION —

The EBCDIC TO HOLLERITH conversion subroutine (SA01) converts an 8-bit EBCDIC character in the A-register to its equivalent 12-bit HOLLERITH code in the A-register.

The HOLLERITH TO EBCDIC conversion subroutine (SB01) converts a 12-bit 029 HOLLERITH character in the A-register to its equivalent 8-bit EBCDIC character in the A-register.

The EBCDIC to ASCII conversion subroutine (SC01) converts an 8-bit EBCDIC character in the A-register to its equivalent 8-bit ASCII code in the A-register. If other than 8-bit ASCII code is desired, this routine may be easily modified (see section 2.3.17, comments for SC01).

The user should note the following characteristics of these subroutines:

1) Require VDM 620 series computer with a 16-bit word
2) Source statements must be assembled with DAS 8A assembler
3) The extended addressing option is not necessary
4) The multiply/divide option is not necessary

This subroutine package is referenced by the following VDM Software part numbers:

Source Material - 92H0206-001
Object Material - 92U0206-001
Assembly Listing - 92L0206-001

SECTION 2

SUBROUTINE DESCRIPTIONS

2.1

1.	Title:	Convert EBCDIC to Hollerith
2.	Symbolic Name:	SA01
3.	Purpose:	To convert an EBCDIC character in bits 0 through 7 of the A register to IBM 029 Hollerith code in bits 0 through 11 of the A register.
4.	Description:	Index register B and the input value in the A register are saved. An index pointer into the SAT2 table is built by shifting out the least significant bit of the EBCDIC input, saving only the low order 7 bits, and adding the beginning address of SAT2 to the result. The two packed Hollerith characters from SAT2 are then loaded into the B register. The original EBCDIC value is checked and, if even, the left half of the packed Hollerith word is saved or, if odd, the right half is saved. An index pointer into the SAT1 table is then built by saving the least significant 3 bits of the saved Hollerith word and adding the beginning address of SAT1. Rows 12, 11, 0 are shifted to their output position and merged with rows 1-7. The B register is restored and the word with the output positions of rows 12, 11, 0, 1-7 is saved. Rows 8 and 9 are now merged with the other rows into the final output value of the Hollerith code and a return is made to the calling program.

2.14.1 EBCDIC - HOLLERITH TABLE

EBCDIC Character (Hexadecimal)	Punched Card Code	EBCDIC Character (Hexadecimal)	Punched Card Code
00	12-0-1-8-9	20	11-0-1-8-9
01	12-1-9	21	0-1-9
02	12-2-9	22	0-2-9
03	12-3-9	23	0-3-9
04	12-4-9	24	0-4-9
05	12-5-9	25	0-5-9
06	12-6-9	26	0-6-9
07	12-7-9	27	0-7-9
08	12-8-9	28	0-8-9
09	12-1-8-9	29	0-1-8-9
0A	12-2-8-9	2A	0-2-8-9
0B	12-3-8-9	2B	0-3-8-9
0C	12-4-8-9	2C	0-4-8-9
0D	12-5-8-9	2D	0-5-8-9
0E	12-6-8-9	2E	0-6-8-9
0F	12-7-8-9	2F	0-7-8-9
10	12-11-1-8-9	30	12-11-0-1-8-9
11	11-1-9	31	1-9
12	11-2-9	32	2-9
13	11-3-9	33	3-9
14	11-4-9	34	4-9
15	11-5-9	35	5-9
16	11-6-9	36	6-9
17	11-7-9	37	7-9
18	11-8-9	38	8-9
19	11-1-8-9	39	1-8-9
1A	11-2-8-9	3A	2-8-9
1B	11-3-8-9	3B	3-8-9
1C	11-4-8-9	3C	4-8-9
1D	11-5-8-9	3D	5-8-9
1E	11-6-8-9	3E	6-8-9
1F	11-7-8-9	3F	7-8-9

2.1. 4.1 (Continued)

EBCDIC Character (Hexadecimal)	Punched Card Code	EBCDIC Character (Hexadecimal)	Punched Card Code
40	No punches	60	11
41	12-0-1-9	61	0-1
42	12-0-2-9	62	11-0-2-9
43	12-0-3-9	63	11-0-3-9
44	12-0-4-9	64	11-0-4-9
45	12-0-5-9	65	11-0-5-9
46	12-0-6-9	66	11-0-6-9
47	12-0-7-9	67	11-0-7-9
48	12-0-8-9	68	11-0-8-9
49	12-1-8	69	0-1-8
4A	12-2-8	6A	12-11
4B	12-3-8	6B	0-3-8
4C	12-4-8	6C	0-4-8
4D	12-5-8	6D	0-5-8
4E	12-6-8	6E	0-6-8
4F	12-7-8	6F	0-7-8
50	12	70	12-11-0
51	12-11-1-9	71	12-11-0-1-9
52	12-11-2-9	72	12-11-0-2-9
53	12-11-3-9	73	12-11-0-3-9
54	12-11-4-9	74	12-11-0-4-9
55	12-11-5-9	75	12-11-0-5-9
56	12-11-6-9	76	12-11-0-6-9
57	12-11-7-9	77	12-11-0-7-9
58	12-11-8-9	78	12-11-0-8-9
59	11-1-8	79	1-8
5A	11-2-8	7A	2-8
5B	11-3-8	7B	3-8
5C	11-4-8	7C	4-8
5D	11-5-8	7D	5-8
5E	11-6-8	7E	6-8
5F	11-7-8	7F	7-8

2.1. 4.1 (Continued)

EBCDIC Character (Hexadecimal)	Punched Card Code	EBCDIC Character (Hexadecimal)	Punched Card Code
80	12-0-1-8	A0	11-0-1-8
81	12-0-1	A1	11-0-1
82	12-0-2	A2	11-0-2
83	12-0-3	A3	11-0-3
84	12-0-4	A4	11-0-4
85	12-0-5	A5	11-0-5
86	12-0-6	A6	11-0-6
87	12-0-7	A7	11-0-7
88	12-0-8	A8	11-0-8
89	12-0-9	A9	11-0-9
8A	12-0-2-8	AA	11-0-2-8
8B	12-0-3-8	AB	11-0-3-8
8C	12-0-4-8	AC	11-0-4-8
8D	12-0-5-8	AD	11-0-5-8
8E	12-0-6-8	AE	11-0-6-8
8F	12-0-7-8	AF	11-0-7-8
90	12-11-1-8	B0	12-11-0-1-8
91	12-11-1	B1	12-11-0-1
92	12-11-2	B2	12-11-0-2
93	12-11-3	B3	12-11-0-3
94	12-11-4	B4	12-11-0-4
95	12-11-5	B5	12-11-0-5
96	12-11-6	B6	12-11-0-6
97	12-11-7	B7	12-11-0-7
98	12-11-8	B8	12-11-0-8
99	12-11-9	B9	12-11-0-9
9A	12-11-2-8	BA	12-11-0-2-8
9B	12-11-3-8	BB	12-11-0-3-8
9C	12-11-4-8	BC	12-11-0-4-8
9D	12-11-5-8	BD	12-11-0-5-8
9E	12-11-6-8	BE	12-11-0-6-8
9F	12-11-7-8	BF	12-11-0-7-8

2.1. 4.1 (Continued)

EBCDIC Character (Hexadecimal)	Punched Card Code	EBCDIC Character (Hexadecimal)	Punched Card Code
C0	12-0	E0	0-2-8
C1	12-1	E1	11-0-1-9
C2	12-2	E2	0-2
C3	12-3	E3	0-3
C4	12-4	E4	0-4
C5	12-5	E5	0-5
C6	12-6	E6	0-6
C7	12-7	E7	0-7
C8	12-8	E8	0-8
C9	12-9	E9	0-9
CA	12-0-2-8-9	EA	11-0-2-8-9
CB	12-0-3-8-9	EB	11-0-3-8-9
CC	12-0-4-8-9	EC	11-0-4-8-9
CD	12-0-5-8-9	ED	11-0-5-8-9
CE	12-0-6-8-9	EE	11-0-6-8-9
CF	12-0-7-8-9	EF	11-0-7-8-9
D0	11-0	F0	0
D1	11-1	F1	1
D2	11-2	F2	2
D3	11-3	F3	3
D4	11-4	F4	4
D5	11-5	F5	5
D6	11-6	F6	6
D7	11-7	F7	7
D8	11-8	F8	8
D9	11-9	F9	9
DA	12-11-2-8-9	FA	12-11-0-2-8-9
DB	12-11-3-8-9	FB	12-11-0-3-8-9
DC	12-11-4-8-9	FC	12-11-0-4-8-9
DD	12-11-5-8-9	FD	12-11-0-5-8-9
DE	12-11-6-8-9	FE	12-11-0-6-8-9
DF	12-11-7-8-9	FF	12-11-0-7-8-9

5. Entry Point: SA01

6. Calling Sequence:

```
P-1   LDA value to be converted
P     JMPM  SA01
P+1
P+2   Any instruction
```

7. Entrance Parameters: EBCDIC character in bits 0 through 7 of A register.

8. Exit Point: Only one exit exists for this subroutine. Return is to P+2 of the calling program.

9. Exit Parameters:

X Register unchanged
B Register unchanged
A Register converted value in bits 0 through 11, as follows:

CPU bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Card column	-	-	-	-	12	11	0	1	2	3	4	5	6	7	8	9

10. Tables or Files Modified or Read: SAT1 - Hollerith 0-7 punch table
SAT2 - Hollerith character table

11. Tables or Files Created: None

12. Called By: Any system program

13. Called From: None

14. Exception Conditions: None

15. Timing: 55 cycles

16. Size:

30 words - Instructions
2 words - Temporary storage
136 words- Tables

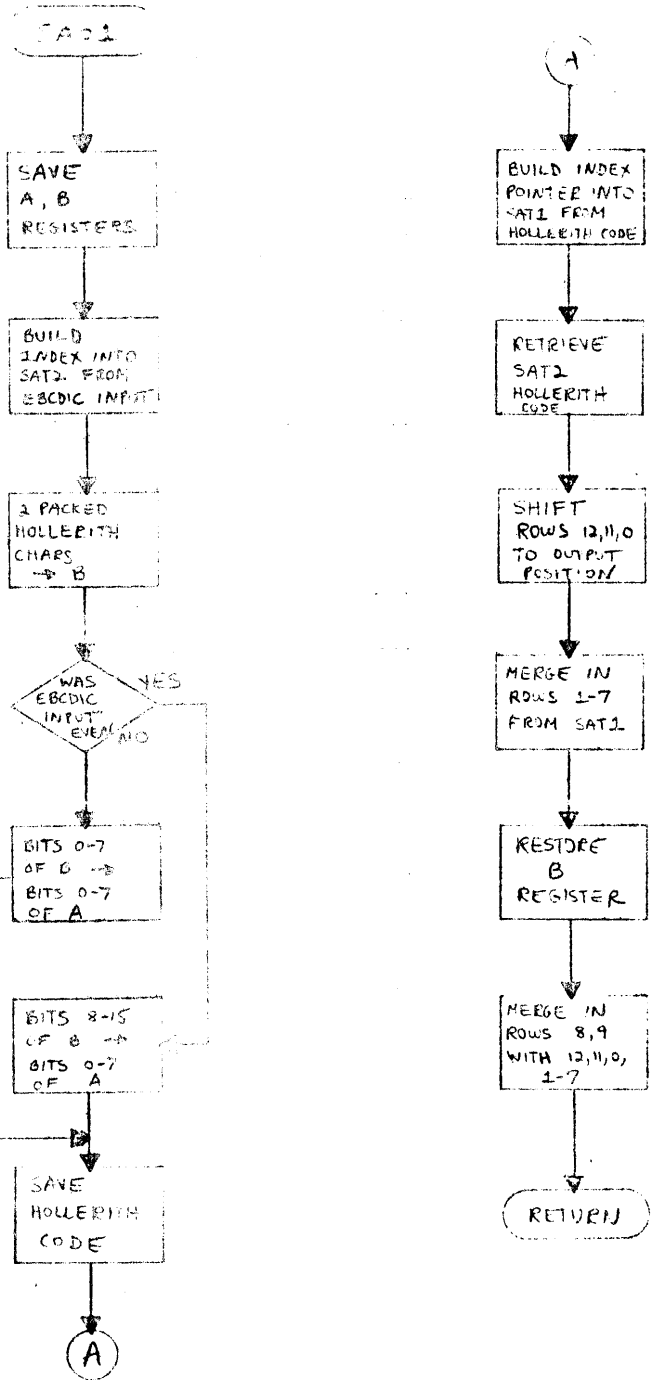
17. Comments: This subroutine is not re-entrant. Every EBCDIC character is convertable. That is, there is no error condition associated with this subroutine.

18. Special Notation: None

19. Hardware Details: None

20. Flowcharts:

PROGRAMMER EBKO / S. Cohen	PROGRAM NO.	DATE	PAGE 1 + 1
CHART I.D.	CHART NAME EBCDIC TO HOLLERITH	PROGRAM NAME SAT 1	



FOLD UNDER AT DOTTED LINE

varian data machines, a varian subsidiary	CODE IDENT. NO.	89A0200	REV A
	PREPARED BY	APPROVED BY	SHEET 12 OF 24

2.2

1. Title: Convert Hollerith to EBCDIC

2. Symbolic Name: SB01

3. Purpose: To convert an 029 Hollerith character in bits 0 through 11 of the A register to its corresponding EBCDIC code in bits 0 through 7 of the A register.

4. Description: Index registers B and X are saved. The B register is initialized to zero and will be used as the field 1 (1-7 punches) pointer. The original Hollerith input is checked to see if row 9 is punched. If punched, bit 11 is set and if not punched, bit 11 is reset. The Hollerith input rows 1-7 are now searched against table SBT1 to determine which of the rows 1-7 (or none) has been punched. More than 1 punch in rows 1-7 results in a return to the calling program with the accumulator set negative. An index into table SBT2 is built by isolating card rows 9, 12, 11, 0 into bit positions 6, 5, 4, 3 and adding the displacement into SBT1 which was determined when a match on rows 1-7 was found. This 7 bit number is added to the beginning address of table SBT2 to obtain the address of the corresponding two EBCDIC characters - one with an 8 punch and one without an 8 punch. The original Hollerith input is now checked to see if row 8 was punched. The appropriate EBCDIC is then right justified in the accumulator, the B and X registers are restored, and a return is made to the calling program.

5. Entry Point: SB01

6. Calling Sequence:

P-1	LDA value to be converted
P	JMPM
P+1	SB01
P+2	Any instruction

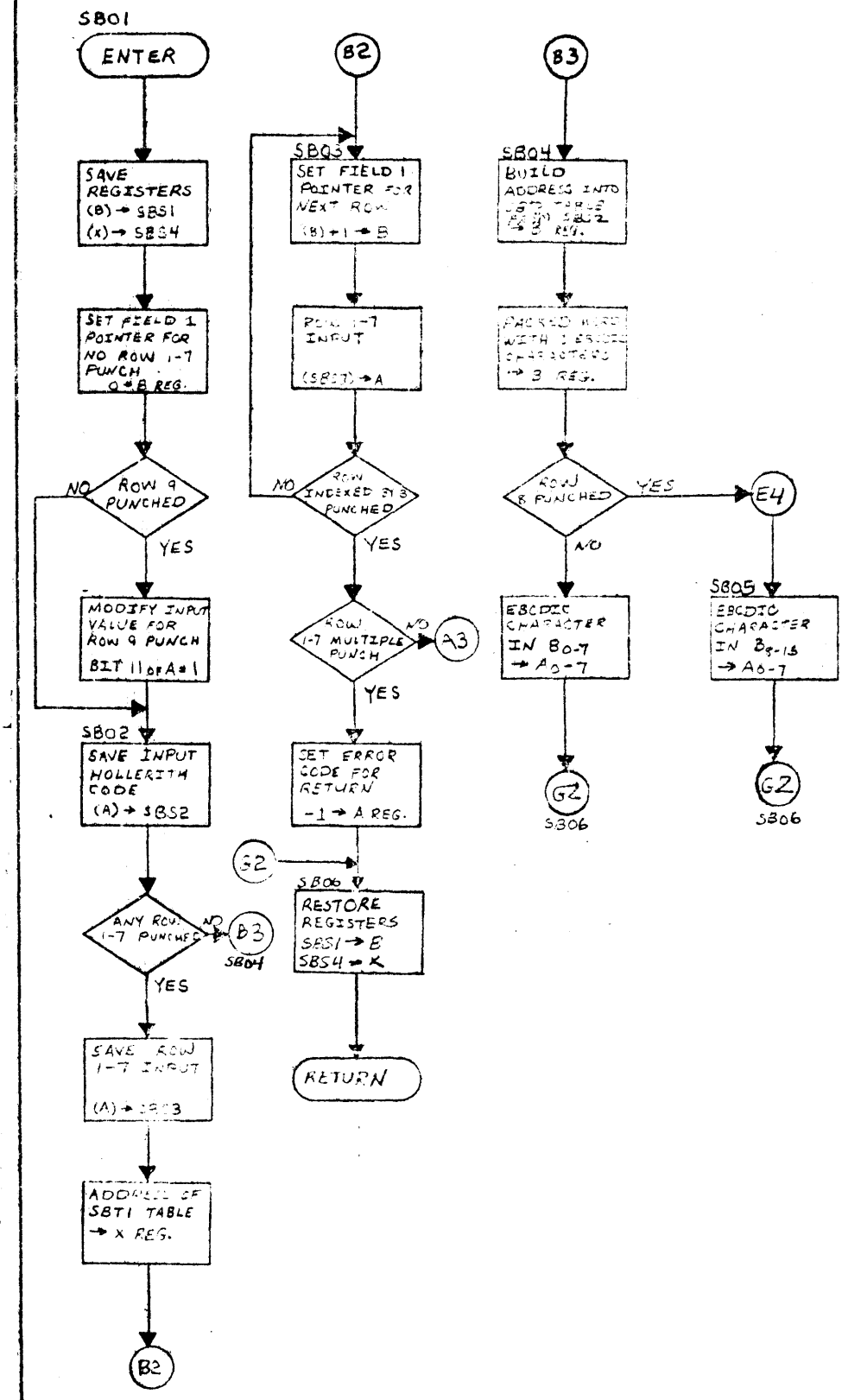
7. Entrance Parameters: 029 Hollerith character in bits 0 through 11 of A register.

8.	Exit Point:	Only one exit exists for this subroutine. Return is to P+2 of the calling program.
9.	Exit Parameters:	X Register unchanged B Register unchanged A Register converted value in bits 0 through 7.
10.	Tables or Files Modified or Read:	SBT1 1-7 row punch table SBT2 EBCDIC character table
11.	Tables or Files Created:	None
12.	Called By:	Any system program
13.	Called From:	None
14.	Exception Conditions:	An input containing more than one punch in rows 1-7 is an error and results in a return to the calling program with the accumulator set negative.
15.	Timing:	44 cycles minimum 114 cycles maximum
16	Size:	46 words - Instructions 4 words - Temporary storage 136 words - Tables
17.	Comments:	This subroutine is not re-entrant.
18.	Special Notation:	None
19.	Hardware Details:	None
20.	Flowcharts:	

Flowchart Worksheet

PROGRAMMER | PROGRAM NO. | DATE | PAGE

CHART I.D. | CHART NAME | PROGRAM NAME



FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines
a varian subsidiary

CODE IDENT. NO.		89A0200	REV. A
PREPARED BY	APPROVED BY	SHEET 15 OF 24	

2.3

1. Title: EBCDIC to ASCII Conversion

2. Symbolic Name: SC01

3. Purpose: To convert an 8-bit EBCDIC character in the A register to its equivalent 8-bit ASCII code in the A register.

4. Description: Index register B and the input value in the A register are saved. The input value is checked to see if it is in the range of either 40-7F or C0-FF. If not, a return to the calling program is made with the accumulator negative. Otherwise, a shift command to test the validity of the value to be converted is built. The displacement into the SCT1 table is calculated by right justifying bits 4 and 5 of the EBCDIC input. Adding the beginning address of SCT1 to this value yields the address of the word in SCT1 containing the legality bit for this EBCDIC input. A check is now made to determine exactly which range the input value is in, 40-7F or C0-FF. If 40-7F, an offset of zero into the SCT1 table is used. If C0-FF, an offset of four is used. The legality bit for this input value is now checked. If not legal, a return to the calling routine is made with the accumulator negative. If legal, bits 1-5 of the EBCDIC input value are added to the beginning address of the SCT2 table to obtain the address of the word containing the two ASCII characters for this value. If the EBCDIC input value was even, the left ASCII character is set in the accumulator, right justified. If odd, the right ASCII character is set in the accumulator, also right justified. The B register is then restored and a return is made to the calling program.

5. Entry Points: SC01

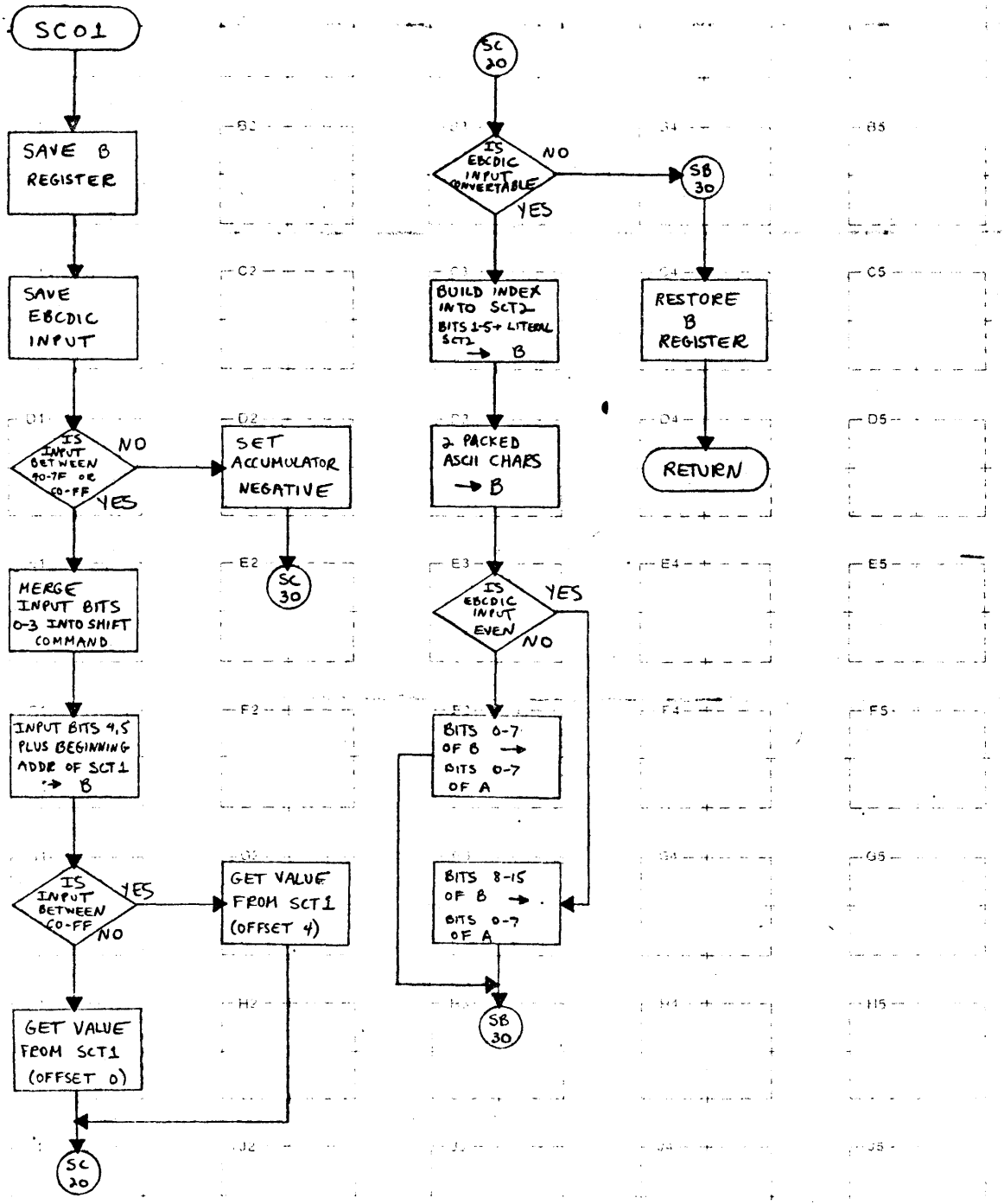
6. Calling Sequence:

```
P-1   LDA value
P     JMPM SC01
P+2   Any Instruction
```

7.	Entrance Parameters:	EBCDIC character to be translated in bits 0 through 7 of A register
8.	Exit Point:	Only one exit exists for this subroutine. Return is to P+2 of the calling program.
9.	Exit Parameters:	B register unchanged. ASCII code in bits 0 through 7 of the A register.
10.	Table or Files Modified or Read:	SCT1 Legality check table SCT2 ASCII character table
11.	Tables or Files Created:	None
12.	Called By:	Any system program
13.	Called From:	None
14.	Exception Conditions:	If the input value is not in the ranges 40-7F or C0-FF, or if the legality status bit in Table SCT1 is set, a return to the calling program is made with the accumulator set negative.
15.	Timing:	17 cycles minimum 72 cycles maximum
16.	Size:	44 words - Instructions 2 words - Temporary storage 40 words - Tables
17.	Comments:	This subroutine is not re-entrant. Some output devices allow only 7-bit ASCII. If other than 8-bit ASCII is desired, this subroutine should be modified as follows: Either - 1) Modify table SCT2 to include desired codes or 2) Insert an appropriate mask instruction at location SC30+1.
18.	Special Notation:	None
19.	Hardware Details:	None
20.	Flowcharts:	

PROGRAMMER EBKD/S Cohen | PROGRAM NO | DATE 2/24/71 | PAGE 1 of 1

CHART I.D. | CHART NAME EBCDIC TO ASCII | PROGRAM NAME SC01



FOLD UNDER AT DOTTED LINE

FOLD UNDER AT DOTTED LINE

varian data machines
a varian subsidiary

CODE IDENT. NO.

89A0200 RE A

PREPARED BY | APPROVED BY | SHEET 18 OF 24

89A0349-000A

SECTION 3

TEST PROCEDURE AND RESULTS

3.1 DESCRIPTION OF PROCEDURE

The three conversion routines were tested as follows:

Two tables, TBL1 and TBL2 were generated. For the EBCDIC TO HOLLERITH conversion, sequential EBCDIC values from 0 to 255 were placed into the A-register and a call to SA01 was made for each value. The output HOLLERITH characters were then placed sequentially into TBL1.

For the HOLLERITH TO EBCDIC conversion, the HOLLERITH values from TBL1 were placed into the A-register and a call to SB01 was made for each value. The output EBCDIC characters were placed sequentially into TBL2. An illegal HOLLERITH character was then given to SB01 with the associated error return stored in TBL2+256.

For the EBCDIC TO ASCII conversion, the EBCDIC values from TBL2 were placed into the A-register and a call to SC01 was made for each value. The output ASCII characters (or error codes) were sequentially stored back into TBL1.

PAGE 0 C00001

```
001500                          .CRG    .01500
                        ***
                        *       TEST ROUTINE FOR SA01
                        ***
001500  031562          TS01    .LDX    .ADT1           ADDRESS OF HOLLERITH TABLE TO X
001501  011561                  .LDA    .TSC1           LOAD EBCDIC VALUE
001502  005111                  .IAR    .               ADD ONE
001503  141000                  .SLB    .=256           ARE WE DONE
001504  001010                  .JAZ    .TS05           YES
001505  001516 R
001506  121000                  .ADD    .=256           NO - GET EBCDIC VALUE BACK
001507  031561                  .STA    .TSC1
001510  002000                  .CALL   .SA01           CONVERT
001511  002565 R
001512  056000                  .STA    .0,X            STORE HOLLERITH VALUE
001513  005144                  .IXR    .               INCREMENT HOLLERITH TABLE ADDRESS
001514  001000                  .JMP    .TS01           DO NEXT VALUE
001515  001501 R
001516  000001          TS05    .HLT    .1
                        ***
                        *       TEST ROUTINE FOR SB01
                        ***
001517  031562                  .LDX    .ADT1           ADDRESS OF HOLLERITH TABLE TO X
001520  022163                  .LDB    .ADT2           ADDRESS OF EBCDIC TABLE TO B
001521  015000          TS10    .LDA    .0,X            GET HOLLERITH VALUE
001522  002000                  .CALL   .SB01           CONVERT
001523  003037 R
001524  056000                  .STA    .0,B            STORE EBCDIC VALUE
001525  005144                  .IXR    .               INCREMENT HOLLERITH TABLE ADDRESS
001526  005122                  .IBR    .               INCREMENT EBCDIC TABLE ADDRESS
001527  011560                  .LDA    .CNTR           ARE WE DONE
001530  141001                  .SUB    .=255
001531  001010                  .JAZ    .TS15           YES
001532  001536 R
001533  041560                  .INR    .CNTR           NO - INCREMENT COUNTER
001534  001010                  .JMP    .TS10
001535  001521
```

```
001536  011002      TS15  .LDA    .=014          ERROR VALUE
001537  002000            .CALL   .SBOJ
001540  003037 R
001541  055000            .STA    .0,6           STORE ERROR RETURN
```

PAGE 000002

```
001542  000003            .HLT    .3
                    ***
                    *       TEST ROUTINE FOR SC01
                    ***
001543  031562            .LDX    .ADT1          ADDRESS OF ASCII  TABLE TO X
001544  022163            .LDB    .ADT2          ADDRESS OF EBCDIC TABLE TO B
001545  016000      TS20  .LDA    .0,B           GET EBCDIC VALUE
001546  001004            .JAN    .TS30          ARE WE DONE - YES
001547  001557 R
001550  002000            .CALL   .SC01          NO - CONVERT
001551  003332 R
001552  055000            .STA    .0,X           STORE ASCII OR ERROR VALUE
001553  005144            .IXR    .              INCREMENT ASCII TABLE ADDRESS
001554  005122            .IBR    .              INCREMENT EBCDIC TABLE ADDRESS
001555  001000            .JMP    .TS20          DO NEXT EBCDIC VALUE
001556  001545 R
001557  000004      TS30  .HLT    .4

001560  000000      CNTR  .DATA   .0             EBCDIC COUNTER
001561  177777      TSC1  .DATA   .0177777       EBCDIC VALUE
001562  001563 R    ADT1  .DATA   .TBL1          HOLLERITH AND ASCII TABLE
001563            TBL1  .BSS    .256
002163  002164 R    ADT2  .DATA   .TBL2          EBCDIC TABLE
002164            TBL2  .BSS    .257
```

3.3.3 Test of EBCDIC TO ASCII (SC01)

Output of SC01 to TBL1 (TBL2 has been used as input).

```
01543.



D1563.
( 001563 )  177777 177777 177777 177777 177777
( 001570 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001600 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001610 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001620 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001630 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001640 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001650 )  177777 177777 177777 177777 177777 177777 177777 177777
( 001660 )  177777 177777 177777 000240 177600 177401 177003 176007
( 001670 )  174017 170037 160077 140177 100377 000333 000256 000214
( 001700 )  000250 000253 000336 000246 177600 177401 177003 176007
( 001710 )  174017 170037 160077 140177 100377 000241 000244 000252
( 001720 )  000251 000273 000337 000255 000257 177600 177401 177003
( 001730 )  176007 174017 170037 160077 140177 100377 000254 000245
( 001740 )  000335 000276 000277 177700 177601 177403 177007 176017
  001750 )  174037 170077 160177 140377 100777 000272 000243 000300
( 001760 )  000247 000275 000242 177777 177777 177777 177777 177777
( 001770 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002000 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002010 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002020 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002030 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002040 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002050 )  177777 177777 177777 177777 177777 177777 177777 177777
( 002060 )  177777 177777 177777 100077 000301 000302 000303 000304
( 002070 )  000305 000306 000307 000310 000311 177000 176001 174003
( 002100 )  170007 160017 140037 100077 000312 000313 000314 000315
( 002110 )  000316 000317 000320 000321 000322 177000 176001 174003
( 002120 )  170007 160017 140037 140077 100177 000323 000324 000325
( 002130 )  000326 000327 000330 000331 000332 177400 177001 176003
  002140 )  174007 170017 160037 000260 000261 000262 000263 000264
  002150 )  000265 000266 000267 000270 000271 176000 174001 170003
  002160 )  160007 140017 100037
```