Burroughs Corporation

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

2212 5355

1400 INTERPRETER

PRODUCT SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	10/21/76		Original Issue (MARK 6.0 Release Level)
B	4/5/77		Changes to Revision B (Mark 6.1 Release Level):

Page	Change
1-1 to 13-3	B1700 replaced by B1800/1700, where appropriate. Mark VI.0 replaced by Mark VI.1 throughout.
2-1	CRT SPO added to item B.
5-1 to 5-2	CRT application added to SPO GRAPHICS; ** Note added on page 5-1.
6-10	New SAMPLE MEMORY DUMP.
6-13	Note on EOF CRD added.
9-18	1400 Unit Number added to Abbreviation Table.
10-6	"No backspace . . ." added to DROPOUT ENCOUNTERED and TIMEOUT ENCOUNTERED messages.
13-1	Error Flags and Branch Indicators (Item J) added.
13-2	Paragraph on SPO interaction added.
13-3	New SAMPLE TRACE.

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
C	2/6/78		MARK VII.0 RELEASE

Page	Change
	Mix-number replaced by job-number throughout
7-1	Added to processing options: SYSIN note added.
8-1	PROCESSING OPTIONS added
8-3	INVERTED.EDIT explained
8-4	SYSIN explained
10-4	Message #14 (*** DISK # NOT SPECIFIED...) modified
12-1	CARD.READER.2 replaced by SYSIN

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
D	4/30/79		MARK VIII.0 RELEASE

Page	Change
2-1	Deleted B1820 from requirement *A.
2-2	Changed 17.5K to 18.0KB.
9-6	Added "Blank Don't Skip" character.
9-12	Changed Y to U for WT.
	Changed R to W for WCP.
	Added "Only ")"" sentence for WCP.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

TABLE OF CONTENTS

INTRODUCTION

The Burroughs B1800/B1700 interpreter for creating the environment of an IBM 1401/1440/1460 machine is called the 1400 Interpreter. It is a means of executing 1401/1440/1460 object programs on a Burroughs 1800/1700 under the control of MCPII. Through the use of the 1400 Interpreter and the 1400 IEP (See Related Documentation below), the object programs which were originally intended to be executed on an IBM machine can be emulated on the B1800/B1700's virtual machine.

The intent of this product is to provide IBM 1401/1440/1460 users and IBM 360 users simulating 1400 systems with the ability to take the object programs, data, and operating instructions of the original machine and execute them on a B1800/B1700 as if it were the original machine.

RELATED DOCUMENTATION

Name	Number
1400 Interpreter Environment Program	P. S. 2212 5504
B1800/B1700 MCP II	P. S. Not Released
B1800/B1700 Software Operational Guide	1068731
CREATE/VRTL.1311	P. S. 2212 5520

OPERATING REQUIREMENTS

The operating requirements of the interpreter are as follows:

* A. 96 KB main memory (B1710 and B1720)
2 KB control memory (B1720 only)

B. Console printer/CRT SPO--for communication purposes

C. Peripheral devices -- those which are similar to those of the 1401/1440/1460 and/or those which are necessary for operation of the B1800/B1700 MCPII.

D. Minimum version of MCPII -- MARK VI.1

E. Minimum version of 1400/IEP - Mark VI.1

This interpreter is available to those IBM users who desire an efficient means of transition to a Burroughs B1800/B1700.

* S-memory requirements are minimum requirements. Those users who will multiprogram other jobs with the 1400 Interpreter may want to exceed 96K S-memory.

The interpreter will operate on a B1720 series computer with 2K M-memory, however, since a minimum of two interpreters compete for M-memory space (SDL Interpreter and 1400 Interpreter), 4K M-memory is beneficial to the user.

Memory requirements for the interpreter itself and the calculations necessary to compute the size of a given target memory are shown under "Memory Calculations" in the next section.

MEMORY CALCULATIONS

Interpreter	18.0KB
With 4K environment, add	6.0KB
With 8K environment, add	10.0KB
With 12K environment, add	14.0KB
With 16K environment, add	18.0KB
With any disk, add disk buffer	136 Bytes
For each pseudo 1311 disk declared, add (Maximum of 5)	2.6KB
For any tape, add tape buffer	0 to 2 X target memory size (Default=4000 Bytes)
For each tape declared, add (Maximum of 5)	120 Bytes
Plus the MCPII requirements	

Note: All figures are approximate and include requirements for line printer, 80 column card reader, and 80 column punch.

INTERPRETER FEATURES

The 1400 Interpreter provides the following 1401/1440/1460 operational features:

A. The interpreter has the same operating environment as defined for the original IBM 1401/1440/1460 object program.

B. All halts allow the operator to continue, reset, load a new program, start at another 1400 address, go to end-of-1400-job, go to end-of-B1800/B1700-job, or execute any other operator command (which is a superset of those defined by the 1401/1440/1460 operator panel).

C. The 1401/1440/1460 arithmetic logic is maintained during execution of the object program.

D. The 1401/1440/1460 comparison-indicator logic is maintained during execution.

E. One level of address indexing is implemented.

F. The A- and B-address registers are maintained during execution.

G. The sense switches are implemented (A-G).

EMULATED INSTRUCTIONS

The following types of 1401/1440/1460 instructions are emulated by the 1400 Interpreter:

A. Input/Output
B. Arithmetic
C. Logical
D. Move and load
E. Magnetic tape
F. Disk
G. Console Printer
H. Miscellaneous

INPUT/OUTPUT INSTRUCTIONS

A. Read card (and branch)
B. Read binary card (and branch)
C. Write line (and branch)
D. Write wordmarks (and branch)
E. Write line and suppress (and branch)
F. Write and read card (and branch)
G. Write wordmarks and read card (and branch)
H. Write line, suppress and read card (and branch)
I. Punch card (and branch)
J. Read and punch cards (and branch)
K. Write line and punch card (and branch)
L. Write wordmarks and punch card (and branch)
M. Write line, suppress and punch card (and branch)
N. Write line, read and punch cards (and branch)
O. Write wordmarks, read and punch cards (and branch)
P. Write line, suppress, read and punch cards (and branch)
Q. Start read feed (treated as effective NOP)
R. Start punch feed (treated as effective NOP)

ARITHMETIC INSTRUCTIONS

A. Add
B. Subtract
C. Zero and add
D. Zero and subtract
E. Multiply
F. Divide

LOGICAL INSTRUCTIONS

A. Branch unconditional
B. Branch if equal compare
C. Branch if high compare
D. Branch if low compare
E. Branch if unequal compare
F. Branch if arithmetic overflow
G. Branch if character equal
H. Branch if bit equal
I. Branch if end of reel
J. Branch if tape transmission error
K. Branch if I/O check stops or busy (treated as effective NOP)
L. Branch if inquiry indicator on
M. Branch if clear indicator
N. Branch if last card indicator on
O. Branch if sense switch on (B - G)
P. Branch if wordmark and/or zone condition
Q. Branch if validity error
R. Branch if wrong length disk record
S. Branch if unequal disk address compare)
T. Branch if any disk error condition
U. Branch if disk access busy (treated as effective NOP)

MOVE AND LOAD INSTRUCTIONS

A. Load characters to a wordmark
B. Move numeric
C. Move zone
D. Move characters and edit
E. Move characters to recordmark or groupmark w/wordmark
F. Move characters and suppress zeros
G. Move characters to A- or B-field wordmark

MAGNETIC TAPE INSTRUCTIONS

The 1400 Interpreter emulates the following magnetic tape instructions for 7- and 9-track NRZ and phase encoded tapes in both parities and most densities. Note: Due to B1800/B1700 system hardware restrictions, 7-track tape records of less than eight characters cannot be written or read. Attempts to write those records are trapped by the interpreter, but attempts to read them are lost because the hardware tape control treats them as noise.

A. Backspace one record
B. Read tape
C. Read tape with wordmarks
D. Read binary tape
E. Read binary tape with wordmarks
F. Rewind tape
G. Rewind tape and unload
H. Skip and blank tape (erase)
I. Write tape
J. Write tape with wordmarks
K. Write binary tape
L. Write binary tape with wordmarks
M. Write tapemark
N. Diagnostic read tape

DISK INSTRUCTIONS

A. Read disk sectors
B. Read disk sectors with wordmarks
C. Read disk with sector count overlay
D. Read disk with sector count overlay with wordmarks
E. Read disk track with addresses
F. Read disk track with addresses and with wordmarks
G. Seek disk
H. Write disk sectors
I. Write disk sectors with wordmarks
J. Write disk check
K. Write disk check with wordmarks
L. Write disk with sector count overlay
M. Write disk with sector count overlay with wordmarks
N. Write disk track with addresses
O. Write disk track with addresses and with wordmarks
P. Scan disk
Q. Scan disk with wordmarks

CONSOLE PRINTER INSTRUCTIONS

(For 1407/1447 only)

A. Read console
B. Read console with wordmarks
C. Write console
D. Write console with wordmarks

MISCELLANEOUS INSTRUCTIONS

A. Compare
B. Control carriage on printer
C. Clear storage (and branch)
D. Clear wordmark
E. Halt (and branch)
F. Modify address
G. No operation
H. Store A-address register
I. Store B-address register
J. Set wordmark
K. Select stacker (read select and punch select)

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INTERNAL CODE

	1401/1440/1460 GRAPHIC	1401/1440/1460 INTERNAL CODE MCBA8421	INTERPRETER INTERNAL CODE* XMXX XXXX	HEX	B1800/B1700 PRINTER GRAPHICS	B1800/B1700 SPO/CRT GRAPHICS
0	BLANK		0100 0000	40	BLANK	BLANK
1	.	BA8 21	0100 1011	4B	.	.
2	⌑ (LOZENGE)	BA84	0100 1100	4C	<	<
3	(	BA84 1	0100 1101	4D	(	(
4	<	BA842	0100 1110	4E	+	+
5	⌗ (GROUPMARK)	BA8421	0100 1111	4F	!	!
6	&	BA	0101 0000	50	&	&
7	$	B 8 21	0101 1011	5B	$	$
8	*	B 84	0101 1100	5C	*	*
9	)	B 84 1	0101 1101	5D	)	)
10	;	B 842	0101 1110	5E	;	;
11	Δ (DELTA)	B 8421	0101 1111	5F	¬	↑ **
12	-	B	0110 0000	60	-	-
13	/	A 1	0110 0001	61	/	/
14	,	A8 21	0110 1011	6B	,	,
15	%	A84	0110 1100	6C	%	%
16	⌄ (WORD SEP.)	A84 1	0110 1101	6D	_	←
17	'	A842	0110 1110	6E	:	/
18	## (TAPE SEG.)	A8421	0110 1111	6F	?	?
19	b (SUBST.BLANK)	A	0111 1010	7A	:	
20	#	8 21	0111 1011	7B	#	#
21	@	84	0111 1100	7C	@	@
22	:	84 1	0111 1101	7D	:	:
23	>	842	0111 1110	7E	=	=
24	⌄ (TAPEMARK)	8421	0111 1111	7F	"	
25	◇ (+0)	BA8 2	1100 0000	C0	&	&
26	A	BA 1	1100 0001	C1	A	A
27	B	BA 2	1100 0010	C2	B	B
28	C	BA 21	1100 0011	C3	C	C
29	D	BA 4	1100 0100	C4	D	D
30	E	BA 4 1	1100 0101	C5	E	E
31	F	BA 42	1100 0110	C6	F	F

* In the 1400 Interpreter, the wordmark bit being off represents a wordmark being present.

** For CRT SPO operation, the graphic for (DELTA) is the same as the printer graphic (¬).

1401/1440/1460 GRAPHIC		1401/1440/1460 INTERNAL CODE MCBA8421	INTERPRETER INTERNAL CODE* XMXX XXXX	HEX	B1800/B1700 PRINTER GRAPHICS	B1800/B1700 SPO/CRT GRAPHICS
32	G	BA 421	1100 0111	C7	G	G
33	H	BA8	1100 1000	C8	H	H
34	I	BA8 1	1100 1001	C9	I	I
35	! (-0)	B 8 2	1101 0000	D0		
36	J	B 1	1101 0001	D1	J	J
37	K	B 2	1101 0010	D2	K	K
38	L	B 21	1101 0011	D3	L	L
39	M	B 4	1101 0100	D4	M	M
40	N	B 4 1	1101 0101	D5	N	N
41	O	B 42	1101 0110	D6	O	O
42	P	B 421	1101 0111	D7	P	P
43	Q	B 8	1101 1000	D8	Q	Q
44	R	B 8 1	1101 1001	D9	R	R
45	≠ (RECORDMARK)	A8 2	1110 0000	E0	:	:
46	S	A 2	1110 0010	E2	S	S
47	T	A 21	1110 0011	E3	T	T
48	U	A 4	1110 0100	E4	U	U
49	V	A 4 1	1110 0101	E5	V	V
50	W	A 42	1110 0110	E6	W	W
51	X	A 421	1110 0111	E7	X	X
52	Y	A8	1110 1000	E8	Y	Y
53	Z	A8 1	1110 1001	E9	Z	Z
54	0	8 2	1111 0000	F0	0	0
55	1	1	1111 0001	F1	1	1
56	2	2	1111 0010	F2	2	2
57	3	21	1111 0011	F3	3	3
58	4	4	1111 0100	F4	4	4
59	5	4 1	1111 0101	F5	5	5
60	6	42	1111 0110	F6	6	6
61	7	421	1111 0111	F7	7	7
62	8	8	1111 1000	F8	8	8
63	9	8 1	1111 1001	F9	9	9

* In the 1400 Interpreter, the wordmark bit being off represents a wordmark being present.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

1400 SPO COMMANDS

INTRODUCTION

The 1400 SPO commands were designed to provide the user with all the controls necessary to emulate the functions of the 1401/1440/1460 control panel. Since, in some cases emulation can provide some extra features, or provisions for ease of use; an effort has been made to allow the user to maintain complete control of his emulated program with the help of the following SPO commands.

GENERAL SYNTAX

Following these few rules will provide the necessary knowledge to operate the 1800/1700 SPO.

<job-number>AX<blanks.allowed><command><any.text><blanks><parameters>ETX

<job-number>	::=	The job-number of the B1800/B1700 job.
<blanks.allowed>	::=	Zero to any number of blanks allowed here.
<command>	::=	The three character command described on the following pages.
<any.text>	::=	Any text that the user would like to include to document the command in a better fashion (it may not contain any embedded blanks). <any.text> is optional.
<blanks>	::=	Any number of blanks may be inserted here, but at least one must be present as a delimiter.
<parameters>	::=	Depending on the particular command, parameters may be required and must be separated by at least one blank.

In the case of emulating the target SPO (1407/1447), however, the format is a little different:

<job-number>AX<blanks.allowed>?<input text to the 1400 program>ETX

<input text to the 1400 program> ::= This is exactly what it says, input text to the 1407 program in response to a 1407/1447 READ op code (MZTO###R or LZTO###R).

Note: All invalid specifications put the target machine into an accept state and processing is temporarily suspended waiting for operator intervention.

ALTER TARGET MEMORY

****
* ALTer *
****

Format: <job-number> ALTer <3-CHAR ADDR> [<NEW CONTENTS>]

Function:

The ALTer command is used to modify the contents of 1400 target memory beginning at the address given in the command by <3-CHAR ADDR>. These contents are replaced by the <NEW CONTENTS> given in the command.

The contents of memory locations succeeding the given location are modified until the <NEW CONTENTS> is exhausted.

All leading blanks occurring in the <NEW CONTENTS> field are ignored.

Upon completion of the ALTer, the interpreter displays the new contents of 1400 target memory beginning at the given 3-character address and continuing for 25 characters. If the <NEW CONTENTS> field is omitted, memory is not altered, but the display only is given.

To enter wordmarks into memory, an up-arrow (↑) character should precede the character to contain the wordmark. A wordmark should not be entered without a succeeding character.

The ALTer command can be specified during the wait state or run state of the virtual machine with no apparent lag in execution of the target program.

Examples:

```
1AXALT 123
 % G/1400/ENV =1 123=↑2↑2↑2↑2↑2↑2↑2↑2↑2↑2↑2↑2↑2
 G/1400/ENV =1 ACCEPT.
1AXALT 123 ↑A005008↑2↑B123↑N
 % G/1400/ENV = 1 123=↑A005008↑2↑B123↑N↑2↑2↑2↑2
 G/1400/ENV =1 ACCEPT.


1AXALTER 400 ↑2↑B400↑N
 % G/1400/ENV =1 400=↑2↑B400↑N
 G/1400/ENV =1 ACCEPT.
1AXALTER.THIS.TO.THAT    600 ↑N12345678901234↑.000999↑N
 % G/1400/ENV =1 600=↑N12345678901234↑.000999↑
 G/1400/ENV =1 ACCEPT.
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

CLEAR CONSOLE INQUIRY

```
*********
* CLEar *
*********
```

Format: <job-number> AX CLEar

Function:

The CLEar command clears the last unused message to the target SPO buffer if used while the target program is not running. As a response to a 1407/1447 read, the CLEar command causes the read to be ignored and processing continues with the next instruction. CLEars are not stored up as in the target SPO input ("?"<TARGET SPO INPUT>). CLEar resets the Inquiry Indicator and sets the CLEar Indicator.

Examples:

```
1AXCLE
 % G/1400/ENV =1 ACCEPT.

1AXCLEAR
 %G/1400/ENV =1 ACCEPT.
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

DISPLAY TARGET MEMORY ON SPO

```
***********
* DISplay *
***********
```

Format: <job-number> AX DISplay <3-CHAR ADDRESS>

Function:

The DISplay command is used to display on the SPO the contents of 1400 target memory beginning at the location given by the <3-CHAR ADDRESS> field and continuing for 25 succeeding locations.

A wordmark in a location is displayed as an up arrow ( ) character preceding the character contained in that location.

All trailing blanks are removed from the display.

The DISplay command can be specified during the wait state or the run state of the virtual machine and will return to that state automatically.

Examples:

```
1AXDIS 120
 Z G/1400/ENV =1 120 =↑2↑2↑2↑A005008↑2↑B123↑N↑2
 G/1400/ENV =1 ACCEPT.


1AXDISPLAY 599
 Z G/1400/ENV =1 599= ↑N12345678901234↑.000999
 G/1400/ENV =1 ACCEPT.

1AXDISPLAY.THE.NEXT.LOCATION 615
 Z G/1400/ENV =1 615=↑.000999↑N
 G/1400/ENV =1 ACCEPT.
```

DUMP TARGET MEMORY TO PRINTER

```
********
* DUMp *
********
```

Format: <job-number> AX DUMp [<DUMP TITLE>]

Function:

The DUMp command is used to dump the contents of 1400 target memory to the line printer.

The dump is preceded on the listing by the lines:

```
**1400 MEMORY DUMP
                    [<DUMP TITLE>]
```

followed by a line which contains the following.

1. The Instruction Address Register (in decimal) and (3-char format).
2. The next instruction to be executed.
3. The contents of the A and B address registers.
4. The Sense Switch settings.
5. The Arithmetic Condition Indicator (<, =, >)
6. The Overflow Indicator ("OV" if set, blank otherwise).
7. The error flags and branch indicators (L,K,Y,X,N,W,V,\).
8. The contents of the Index Registers X1, X2, X3.
9. The contents of memory locations pointed to by the A and B address registers (*'s if the corresponding register is not a valid 1400 address).

If 100 or more consecutive 1400 target memory locations each contain <BLANK>, the printing of the corresponding line is suppressed. The end of the dump is signified by the line:

*** END DUMP

The <DUMP TITLE> can be null.

This dump can now be taken while a target program is executing, and the virtual machine will return to execution of the target program after the dump.

Example:

```
1AXDUMP
 G/1400/ENV =1 ACCEPT.
1AXDUMP THIS IS A SAMPLE PROGRAM DUMP WITH THE L INDICATOR ON.
 G/1400/ENV =1 ACCEPT.
```

SAMPLE MEMORY DUMP

```
*** 1400 MEMORY DUMP ***

                     THIS IS A SAMPLE PROGRAM DUMP WITH THE L INDICATOR ON.

00639 639 OP=B865L    A=091 B=226 SW=A       >  L        X1=251 X2=XER X3=R   0XERROR     169          X
  000C0 (C00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                &,019027,031,038042803IT981B400L046352BW04BS88   VERT. REDUNDANCY CHECK  5080B153 0101125120XERROR
                 1      1   1      1      21   1      1   1      1                                1 1 1  1  11    1
  00100 (100) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                    169             1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ.<&$*-/,%#@+!: EVENODDREAD ERRORWRITTEN
                    1               1                   1                   1           1   1  1         1
  00200 (200) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                               X           X                                  61         61  1
                                                                              13          2  3
  00300 (300) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                                                 ,008012,00110011B361080AB421/340080          B389S681BS88N000M360392
                                                 1      1   1   11        1   1      1         1       1   1   1
  00400 (400) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                /332/8S00S521L083085L091088BT01 LS50262LS50276,261261M088463M125261M%U4261WB809LM%B4261WB823LM%B4261
                1   11       1      1      1    1      1      1      1      1       1       1    1       1    1
  00500 (500) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                WB837LM%U4261WB851LU%U4BU%U4BU%U4BU%U4B0567                        /250M%U4272RB612BM092204M092227B8
                 1    1       1    1    1    1    1    1                            1   1       1    1      1      1
  006C0 (600) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                81LB900C/250M%U4272RB668BM092215M092227B865LM097285B921E2M102285/250M%B4272RB709BM092215M092235B925L
                   1    1   1       1    1      1      1    1      1    11      1   1       1    1      1      1
  007C0 (700) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                8944C/250M%B4272RB427BM092204M092235B963LM097285B983E2M102285/250B427DC0881068786/B997A0830888427
                1    1   1       1    1      1      1    1      1    11       1   1    1      1    1   1      1
  00800 (800) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                U%U4RB427U%U4BU%U4EB467U%U4BU%U4EB480U%U4BU%U4EB493U%U4BU%U4EB506M092247B656CB664M092247M097285B915E
                1    1   1    1    1   1    1    1   1    1    1   1    1    1   1      1    1   1      1      1
  00900 (900) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                2M102285/612250.901  .657M092247M097285B959E2M102285/709250.945M092247B753CB761    .754L091088B:2708
                11      1      1   1 1   1      1      1    11      1      1   1      1    1       1   1      1
  01000 (:00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                55A0830858427              B:49GU%U4M/348299    L083085B427M174205M177216M174228M177235M187252M1942
                  1      1                 1    1    1          1      1   1      1      1      1      1      1
  01100 (/00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                64M1512742/299F413J WRITTEN     WRITTEN       READ     READ       INDICATED      CHARACTER    CHARACTER
                  1      11   1     1
  01200 (S00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                2,049L0772772/2772</40/60</80L/992802/280B:60     I 1
                11   1      11   11      1   1      11   1   1    3 1
  01300 (T00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                 BU01FB800KB432 SW =F LOAD MODE
                 1    1    1
  01400 (U00) - ........09.........19.........29.........39.........49.........59.........69.........79.........89.........99
                 M413467M413480M413493M413506M413571M413612M413668M413709MU992998T06  SW=F LOAD      SW=F LOAD MODE
                 1      1      1      1      1      1      1      1      1      1               1
*** END DUMP ***
```

END OF 1400 JOB

```
*******
* END *
*******
```

Format: <job-number> AX END [<6-CHAR ADDRESS>DUMP]

Function:

The END command is used to terminate a 1400 job. When only END is specified, processing stops temporarily while the next 1400 job is loaded (same as <job-number> AX LOAd). When a 6-character address is given, the virtual machine will issue an END automatically when the following is true:

1. A-register = first 3 char address (leftmost out of 6-chars).
2. B-register = last 3 char address (rightmost out of 6-chars).
3. A HALT instruction is executed.

If DUMp is also specified, then a target memory dump will be produced before the END is issued. The 6-char address must be specified to get the dump.

The END command can be specified during the wait state or run state of the virtual machine and will return to that state automatically.

Examples:

```
1AXEND

1AXEND 999999
 G/1400/ENV =1 ACCEPT.
1AXEND 999999 DUMP
 G/1400/ENV =1 ACCEPT.
```

END OF FILE ON DEVICE

****
* EOFile *
****

Format: <job-number> EOFile [<ETX> or <LOGICAL DEVICE MNEMONIC>]

Function:

The EOFile command is used to close a file on a particular device and divert one's output to another device. For instance, the command may be used to stop output from being printed and to continue backup only. Specification of <ETX> or incorrect specification of the logical device mnemonics will produce a list on the SPO of the correct mnemonics. The legal mnemonics are:

CRD ... Card reader (which includes the reader stacker select files)

PRT ... Line printer

PCH ... Card punch

DK0, DK2, DK4, DK6 and DK8 ... Respective 1311 disks

TP1, TP2, TP3, TP4, TP5, and TP6 ... Respective tape units

The EOFile command does not stop the virtual machine's target program, unless specified incorrectly.

Note: = "EOF CRD" does not set the last card indicator.

Examples:

1AXEOF TP1

1AXEOFILE PRT

END OF B1800/B1700 JOB

```
*******
* EOJ *
*******
```

Format: <job-number> AX EOJ [<6-CHAR ADDRESS>DUMP]

Function:

The EOJ command is used to terminate the interpreter (1800/1700 job). When only EOJ is specified, then processing stops entirely and the interpreter goes to end of job. When a 6-character address is specified, the interpreter will go to end of job if the following is true:

1. A-register = first 3 char address (leftmost of 6-chars)

   and

2. B-register = last 3 char address (rightmost of 6-chars)

   and

3. A HALT instruction is executed

If DUMp is also specified, a target memory dump is issued just before end of job. The EOJ command can be specified during the wait state or the run state of the virtual machine and will return to its previous state automatically.

Examples:

```
1AXEOJ

 G/1400/ENV =1 EOJ. #=14 TIME = 22:18:16.0


1AXALTER 123 ↑N145678↑.↑N
 % G/1400/ENV = 1 ACCEPT.
1AXEOJ 145678
 G/1400/ENV = 1 ACCEPT.
1AXTRANSFER.TO 123
 G/1400/ENV =1 EOJ. #=17 TIME = 22:22:29.9
```

```
EX G/1400/ENV
 G/1400/ENV =1 BOJ. #=20 PR=4 TIME = 22:22:52.4
 % G/1400/ENV =1 READY.
 G/1400/ENV =1 ACCEPT.


1AXALTER 145 ↑N999999↑.↑N
 % G/1400/ENV =1 145=↑N999999↑.↑N
 G/1400/ENV =1 ACCEPT.
1AXEOJ 999999 DUMP
 G/1400/ENV =1 ACCEPT.
1AXTRA 145
 G/1400/ENV =1 EOJ. #=20 TIME = 22:23:54:5

          <DUMP WAS PRODUCED HERE>
```

CONVERT

3 CHARACTER ADDRESS TO 5 DIGIT
5 DIGIT ADDRESS TO 3 CHARACTER

****
* EQUals *
****

Format: <job-number> AX EQUals [<5-DIGIT DECIMAL ADDRESS> or
<3-CHAR 1400 ADDRESS>]

Function:

The command EQUals accepts either a 5-digit decimal address or a 3-character 1400 address and then displays the opposite equivalent.

This command may be specified during the wait state as well as run state of the virtual machine with no appreciable performance loss.

Examples:

```
1AXEQU I9R
 % G/1400/ENV =1  I9R = 11999
 G/1400/ENV =1 ACCEPT.
1AXEQUALS G6I
 % G/1400/ENV =1  G6I = 15769
 G/1400/ENV = 1 ACCEPT.
1AXEQU 11999
 % G/1400/ENV = 1 11999 = I9R
 G/1400/ENV =1 ACCEPT.
1AXEQUALS 15769
 % G/1400/ENV =1 15769 = G6I
 G/1400/ENV =1 ACCEPT.
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

LOAD CARD

```
********
* LOAd *
********
```

Format: <job-number> AX LOAd

Function:

When specified, this command reads the first card (record) from a unit labelled by the IEP (i.e., the 80-column card reader). (Default is "CARDS.1400", unless changed by the environment specifications.) The Instruction Address Register is set to 1400 location 001. A wordmark is placed in that location and then control is transferred to the instructions at that location.

The 1400 program should be self-bootstrapping (self-loading) or the interpreter will completely scan the deck of cards (file of records) until self-bootstrapping becomes evident or until the ?END card.

Examples:

```
1AXLOA
```

```
1AXLOAD
```

```
1AX LOAD CARD
```

LOAD TAPE

****
* LOAd TAPe *
****

Format: <job-number> AX LOAd TAPe

Function:

This sequence of events takes place:

1. Clears target memory
2. Sets a wordmark in Location 001.
3. Reads the first record on TAPE1 in Load mode and even parity. (TAPE1 must be specified within the environment for this command to execute.)
4. Moves the record to target memory beginning at location 001.
5. Control is transferred to the instruction at location 001.

Errors occurring in the specification of the command other than LOAd will cause the command to be considered as "LOAD CARDS".

Examples:

1AX LOA TAP

1AX LOAD TAPE

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

TURN ON/OFF DISK KEYS/SWITCHES

```
*******
* ON  *
* OFF *
*******
```

Format: <job-number> AX ON or OFF [[DISable or WRIte or ADDress] ...]

Function:

DISable refers to the Disable Disk Address Compare Key.

WRIte refers to the Write Disk Switch.

ADDress refers to the Write Address Key.

Whether ON or OFF is specified, those indicators which are on, whether or not they have been changed, are displayed as a response. Just specifying ON or OFF will respond with those indicators that are on.

More than one indicator can be operated on by one command specification, by following the basic rule of <BLANK> delimiters.

Also, the disk error indicators (Y, X, W, V, N, and \), are displayed if they are on.

Examples:

```
1AXON
 Z G/1400/ENV =1 ON=DIS WRI ADD Y X W V N
1AXOFF DIS
 Z G/1400/ENV =1 ON=WRI ADD Y X W V N

1AX OFF ADDRESS
 Z G/1400/ENV =1 ON=WRI Y W
```

ENTER NORMAL MODE

```
**********
* REGular *
**********
```

Format: <job-number> AX REGular

Function:

The REGular command is used to return to Regular processing mode (as opposed to Single-Step processing mode). This command should be followed by one of the following commands to continue processing:

```
<job-number> AX STArt
<job-number> AX OK
<job-number> AX TRAnsfer *
<job-number> AX TRAnsfer <3-char address>
```

Examples:

```
1AXREG
 % G/1400/ENV =1 *** REGULAR PROCESSING MODE RESUMED
 % G/1400/ENV = 1 IAR=00500 (500) OP=B400    A=400 B=504
 G/1400/ENV =1 ACCEPT.


1AXREGULAR
 % G/1400/ENV =1 *** REGULAR PROCESSING MODE RESUMED
 % G/1400/ENV =1  IAR=00500 (500) OP=B400    A=400 B=504
 G/1400/ENV =1 ACCEPT.
```

RESET INDICATORS

```
*********
* RESet *
*********
```

Format: <job-number> AX RESet

Function:

The RESet command performs the same functions as the RESET key on the 1400 Processing Unit. Specification of this command causes all indicator flags to be cleared, except the Comparison Indicators and the Sense Switches. The Op Code and Instruction Length Registers are cleared also. The contents of virtual memory, and the target machine's address registers are not affected.

(Exception: the branch address on a "halt and branch" type of halt, in which the IAR is set to NSI of the halt instruction.)

Examples:

```
1AXRES
 % G/1400/ENV =1 *** MACHINE RESET
 % G/1400/ENV =1  IAR=00500 (500) OP=B400     A=400 B=504
 G/1400/ENV =1 ACCEPT.
1AXRESET
 % G/1400/ENV =1 *** MACHINE RESET
 % G/1400/ENV =1  IAR=00500 (500) OP=B400     A=400 B=504
 G/1400/ENV =1 ACCEPT.
```

RUN

```
*******
* RUN *
*******
```

Format: <job-number> AX RUN

Function:

The RUN command performs the following functions:

1. Returns the state of the virtual machine to Regular Processing mode (non Single-Step mode).

2. Resets the Stop Address

3. Begins execution of the target program at the location currently in the Instruction Address Register (IAR).

Examples:

```
1AXRUN
```

SET TARGET MEMORY

```
*******
* SET *
*******
```

Format: <job-number> AX SET <CHARACTER>

Function:

The SET command is used to set all of 1400 target memory to the character contained in the <CHARACTER> field.

If more than one character is entered in the <CHARACTER> field, the first character is used.

The character <BLANK> cannot be entered directly, but memory will be set to <BLANK> if the <CHARACTER> field is omitted.

Memory can be set to a character with a wordmark by entering up arrow ( ) followed by the character.

Examples:

```
1AXSET
 G/1400/ENV =1 ACCEPT.
1AXSET↑2
 G/1400/ENV =1 ACCEPT.
1AXSET↑N
 G/1400/ENV =1 ACCEPT.
```

ENTER SINGLE STEP PROCESS MODE

****
* SINgle *
****

Format: <job-number> AX SINgle

Function:

The SINgle command is used to enter the Single-Step Process mode of the 1400. It may be entered at any time during or before the execution of a 1400 program. The fetch of the next instruction in sequence is performed and then control is returned to the user. To execute the next sequential instruction, one of the following should be specified:

```
<job-number> AX STArt
<job-number> AX OK
<job-number> AX TRAnsfer *
```

To return to Regular processing mode, the command <job-number> AX REGular should be invoked. Messages as to the processing modes currently invoked are displayed on the SPO whenever SINgle or REGular is specified.

Examples:

```
1AXSIN
 % G/1400/ENV =1   *** SINGLE STEP MODE
 % G/1400/ENV =1   IAR=00500 (500) OP=B400      A=400 B=504
 G/1400/ENV =1 ACCEPT.
1AXSINGLE
 % G/1400/ENV =1   *** SINGLE STEP MODE
 % G/1400/ENV =1   IAR=00500 (500) OP=B400      A=400 B=504
 G/1400/ENV =1 ACCEPT.
```

START PROCESSING

```
*********
* STArt *
* OK    *
*********
```

Format: <job-number> AX STArt
<job-number> AX OK

Function:

The STArt/OK command causes the execution of the 1400 program to begin at the current value of the Instruction Address Register. If the virtual machine is in Single-Step mode, the STArt/OK command will cause only the current instruction to execute. The Instruction Address Register is incremented by the length of the current instruction.

Examples:

```
1AXSTA
 % G/1400/ENV =1  IAR=00500 (500) OP=B400     A=400 B=504
 G/1400/ENV =1 ACCEPT.

1AXREG
 % G/1400/ENV =1 *** REGULAR PROCESSING MODE RESUMED
 % G/1400/ENV =1  IAR=00500 (500) OP=B400     A=500 B=404
 G/1400/ENV =1 ACCEPT.

1AXSTART

1AXOK
```

STOP VIRTUAL MACHINE

```
********
* STOp *
* STP  *
********
```

Format: <job-number> STOp [<3-CHAR HALT ADDR>]
<job-number> STP [<3-CHAR HALT ADDR>]

Function:

If the halt-address is left blank, the STOp/STP command causes the halting of the target program in the fetch cycle following the execution of the current 1400 instruction. The current state of the target machine is displayed on the SPO.

If the 3-character halt address is specified, then the target machine will halt when the Instruction Address Register is equal to that halt address. To void the halt address, a STOp/STP command without the halt address given can be specified, or the RUN command. A STOp/STP command with the halt address given, may be specified before or during the target program's execution.

To restart the target machine, one of the following can be specified:

<job-number> AX TRAnsfer *
<job-number> AX STArt
<job-number> AX OK

To restart the target program at a new location, specify the following:

<job-number> AX TRAnsfer <3-CHAR ADDR>

If the virtual machine is in Single-Step mode, the STOp/STP command has no effect.

Examples:

```
1AXSTOP
 % G/1400/ENV =1 *** PROGRAM STOPPED
 % G/1400/ENV =1  IAR=00400 (400) OP=B500      A=500 B=404
 G/1400/ENV =1 ACCEPT.


1AXSTOP 500
 G/1400/ENV =1 ACCEPT.
1AXSTART
 % G/1400/ENV =1 *** PROGRAM STOPPED
 % G/1400/ENV =1  IAR=00500 (500) OP=B400      A=500 B=404
```

SET/DISPLAY SENSE SWITCHES

```
**********
* SW=      *
* SWItches *
**********
```

Format: <job-number> AX SW= <SWITCHES>
SWItches ?
OFF

Function:

The SW= command is used to set or display the current sense switch settings.

When the <SWITCHES> field is omitted or when the <SWITCHES> field is a question mark (i.e., <job-number> AX SW= ?), the sense switches are displayed on the SPO in alphabetical order.

When OFF is specified, all switches are reset.

When the <SWITCHES> field is a <VALID SWITCH GROUP>, the switches are set and current status of the sense switches is displayed on the SPO.

A <VALID SWITCH GROUP> is defined by:

```
<VALID SWITCH GROUP> ::= <VALID SWITCH>/
                         <DELIMITER><VALID SWITCH GROUP>/
                         <VALID SWITCH><VALID SWITCH GROUP>

<VALID SWITCH>       ::= A/B/C/D/E/F/G

<DELIMITER>          ::= BLANK
```

Note: As the above BNF implies, switch names may be entered in any order (i.e., SW= A G B <ETX>).

The SW= command can be specified during the wait state or run state of the virtual machine, with no apparent lag in the target program's execution.

Examples:

```
1AXSW=
 % G/1400/ENV = 1 SW= AC
 G/1400/ENV =1 ACCEPT.
1AXSW=?
 % G/1400/ENV =1 SW= AC
 G/1400/ENV =1 ACCEPT.
1AXSW= AB D
 % G/1400/ENV =1 SW= ABD
 G/1400/ENV =1 ACCEPT.
1AXSW= E
 % G/1400/ENV =1 SW= E
 G/1400/ENV =1 ACCEPT.
1AXSW= OFF
 % G/1400/ENV =1 *** ALL SWITCHES ARE OFF
 G/1400/ENV =1 ACCEPT.
1AXSW=
 % G/1400/ENV =1 *** ALL SWITCHES ARE OFF
 G/1400/ENV =1 ACCEPT.
1AX SWI FG
 % G/1400/ENV =1 SW= FG
G/1400/ENV =1 ACCEPT.
1AX SWITCHES.ARE ABC
 % G/1400/ENV =1 SW= ABC
G/1400/ENV =1 ACCEPT.
```

TRANSFER TO TARGET MEMORY ADDRESS

```
************
* TRAnsfer *
************
```

Format: <job-number> AX TRAnsfer <3-CHAR ADDRESS>

Function:

The TRAnsfer command is used to alter the Instruction Address Register to <3-CHAR ADDRESS> and begin execution. The <3-CHAR ADDRESS> must be present. If the <3-CHAR ADDRESS> given is a single asterisk ("*"), execution begins at the current value of the Instruction Address Register.

Examples:

```
1AXTRA 400

1AXTRANSFER 400

1AXTRA *
```

DISPLAY STATUS

```
******
* WY *
******
```

Format: <job-number> AX WY

Function:

The WY command displays on the SPO the current state of the virtual machine, as well as the current values of the Instruction Address Register, the current instruction, and the A and B registers.

Since the interpreter was designed to multiprogram, it is sometimes difficult to know what the virtual machine is doing at any particular time. Without visibly interrupting the target program's execution, the current status of the machine is displayed on the SPO.

Examples:

```
1AXWY
 % G/1400/ENV =1  *** AX DS WAITING FOR KEYBOARD INPUT
 % G/1400/ENV =1  IAR=00500 (500) OP=8400     A=400 B=504
 G/1400/ENV =1 ACCEPT.
```

```
1AXWY
 % G/1400/ENV =1  *** PROGRAM EXECUTING
 % G/1400/ENV =1  IAR=00400 (400) OP=2        A=400 B=333
```

MESSAGE IO SPO

```
**********
* *COmment *
**********
```

Format: <job-number> AX *COmment [<COMMENT>]

Function:

The *COmment command is used to enter a comment line onto the SPO listing.

The <COMMENT> is listed on the SPO only and performs no function in the interpreter.

The *COmment may be specified at any time with no effect on performance.

The <COMMENT> field may be null.

Examples:

```
1AX *CO  THIS IS A COMMENT ON THE SPO
 G/1400/ENV =1 ACCEPT.
```

LIST COMMENT ON PRINTER

```
*******
* *** *
*******
LIST COMMENT ON PRINTER
```

Format: <job-number> AX *** [<COMMENT>]

Function:

This command is used to enter a comment line on the 1400 printer output.

The three asterisks (***) are listed with the <COMMENT> on the listing so that the comment can be discerned from 1400 program output.

The <COMMENT> field may be null.

Examples:

```
1AX***THIS IS A COMMENT TO THE LINE PRINTER
 G/1400/ENV =1 ACCEPT.
```

ENTER TARGET SPO INPUT

```
**************************
* ?<TARGET SPO INPUT> *
**************************
```

Format: <job-number> AX ?<TARGET SPO INPUT>

Function:

This command sets the Inquiry Indicator and fills the target SPO buffer with <TARGET SPO INPUT>. It can be specified before it is asked for, and will be remembered. Otherwise, when it is asked for:

```
*** ENTER 1400 DATA -
```

the same response: "?<TARGET SPO INPUT>" is required. The interpreter will save only one target SPO message, so repeated "?" commands will replace the existing contents of the target SPO buffer.

Examples:

```
1AX?THIS IS A RESPONSE TO AN INQUIRY

1AX?YES
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INTERPRETER OPTIONS (INTOPT)

****
* INTOPT *
****

The INTOPT implementation is another effort toward elimination of all 1400 Interpreter user/operator intervention. This stratagem allows one to specify all necessary SPO commands on cards within one's program deck. Also, certain options may only be specified at run time.

SPO commands inaccessible from INTOPT cards:

LOAD ... to prevent recursive loading
MY ..... because fields normally displayed have not yet been given values

SPO commands usable from INTOPT cards:

ALT	CLE	CON	DIS	DUM	END	EOF	EOJ	EQU	LOAD TAPE
ON	OFF	REG	RES	RUN	SET	SIN	STA	STO	STP
SW=	TRA	***	*CO	?					

Note: *CO displays a comment from an INTOPT card to the SPO.

Processing options include:

RCB.ALLOWED ..... switch modes to Read Column Binary
INVERTED.EDIT ... switch EDIT to non-U.S. style (i.e. periods/commas interchanged)
SYSIN ..... do LOAD from SYSIN file

Note: Processing options are <u>only</u> usable through INTOPT and <u>not</u> from the SPO.

Character changes:

¬ replaces ↑ (up arrow)
: replaces ETX and is used as a command entry separator

Examples:

```
          /---------------------------------------------------
          !                INTOPT    ALTER    123    ¬A123687¬N
          !
      /---------------------------------------------------
      ! INTOPT      DIS 123 :  RCB.ALLOWED
      !
  /---------------------------------------------------
  !                        INTOPT        SW= B : EQU I9R
  !
```

Note: The keyword INTOPT can be located in any card columns; the format is free, but relational. All specifications must occur within columns 1 through 72 inclusive.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

PROCESSING OPTIONS

The options listed below modify the behavior of the interpreter during the processing of an entire job. (Therefore, these options cannot be changed during processing of the job, or essentially after the LOAD process has occurred.)

OPTION	MINIMUM ABBREVIATION
RCB.ALLOWED	RCB
INVERTED.EDIT	INV
SYSIN	SYS

Note that RCB and SYS are mutually exclusive because they are both required to be the last INTOPT specified.

READ COLUMN BINARY

****
RCB.allowed
****

Read Column Binary is implemented in the 1400 Interpreter. When utilizing column binary, it is imperative that the following be true:

(a) RCB.allowed is the final INTOPT option
and (b) is followed by a ?END record
and (c) is followed by the binary deck prefixed by ?DATA<file-name> and postfixed by BIN.END.

The following example illustrates this sequence.

```
?EX 1400
?DATA CARDS.1400
INTOPT   RCB.ALLOWED
?END
?DATA BIN.1400
.
.
.  BINARY OR EBCDIC PROGRAM
.
BIN.END
?END
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INVERTED EDIT

****
INVerted.edit
****

This processing option causes the EDIT opcode to generate non-U.S. style of formats. This means that commas and decimal points are interchanged.

Example:

Field	1234567
U.S. Result	$12,345.67
Non U.S. Result	12.345,67

SYSTEM INPUT

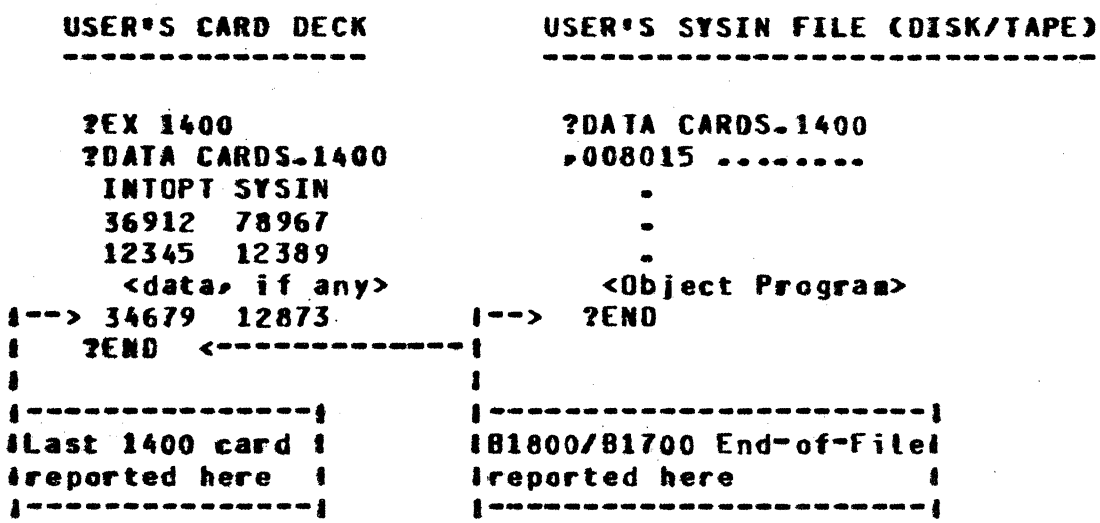
****
SYSin
****

This processing option allows the user to keep his object deck on other storage media. Through file equations of the SYSIN file, or by declarations in the environment compiled by the IEP (see P.S. 2212 5504, 1400 Interpreter Environment Program, "SYSIN") the user defines where the SYSIN file resides. It can reside on 7- or 9-track magnetic tape (NRZ or PE) or on any type of disk. Note that the SYSIN file is a B1800/B1700 file which is usually created by using DMPALL, or file equating the Autocoder's object code punch file to disk/tape.

It is not a 1311 file; it does not reside on a virtual 1311 disk and cannot be accessed by 1400 disk/tape instructions.

The default name for the label card is "CARDS.1400".

Sample SYSIN deck setup:



Currently the implementation is not a dynamic one. In other words, SYSIN should only appear once in any particular job.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

EMULATED INSTRUCTIONS

INSTRUCTION FORMAT						ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	A	B	d		IAR	AAR	BAR	COMMENTS
H	-					NSI	Ap	Bp	Halt.
H	.	AAA				BI	BI	BI	(With indexing).
CW	☒	AAA				NSI	A-1	A-1	Clear wordmark. Wordmarks
CW	☒	AAA	BBB			NSI	A-1	B-1	are cleared at the locations which are specified by the A- and B-address registers without disturbing the data at these locations.
CS	/	AAA				NSI	A	x00-1	Clear Storage. Storage is cleared to blanks starting at the A-Address and continuing downward in storage to the nearest hundreds position.
CS	/	AAA	BBB			BI	BI	NSI	Storage is cleared starting at the B-Address and an unconditional branch is taken to the address specified by the A-Address.
SW	▸	AAA				NSI	A-1	A-1	Set Wordmark. A wordmark
SW	▸	AAA	BBB			NSI	A-1	B-1	is set at the location specified by the A-Address, the data at that address is not disturbed.
D	Z	AAA	BBB			NSI	A-La	A-Lq	Divide. The B-Field data is divided by the A-Field data and the quotient is developed in the high-order positions of the B-Field; the remainder is in the low-order positions of the B-Field. A wordmark must be associated with the high-order position of the divisor.

MNE-OP	ACT-OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
MA	#	AAA			NSI	A-3	A-1	Modify Address. This instruction causes the 3-position field specified by the A-Address to be added to itself.
MA	#	AAA	BBB		NSI	A-3	B-1	The 3-position field specified by the A-Address is added to the B-Address.
M	3	AAA	BBB		NSI	A-La	B-Lp	Multiply. Both fields must have a wordmark associated with them. The units position of the product is located at the B-Address.
ZA	+0	AAA			NSI	A-La	A-La	Zero and Add. The B-Field
ZA	+0	AAA	BBB		NSI	A-Lw	B-Lb	data is set to zero before the A-Field data is added to the B-Field data. The zone bits are removed from all positions except the units.
A	A	AAA			NSI	A-La	A-La	Add. The A-Field data is added to itself. The A-Field must have a wordmark.
A	A	AAA	BBB		NSI	A-Lw	B-Lb	The contents of the A-Address added to the contents of the B-Address. The B-Field must have a wordmark. If the A-Field is shorter than the B-Field, a wordmark must be associated with the A-Field.
B	B	AAA			BI	BI	NSI	With Indexing. A Branch Unconditional is taken to the instruction specified by the A-Address.
BU	B	AAA		/	BI	BI	NSI	Successful Branch.
					NSI	BI	/NS	No Branch. If the B-Field is longer than the A-Field, an unequal compare results.
BLC	B	AAA		A	BI	BI	NSI	Branch if Last Card Indicator ON.

MNE. OP	INSTRUCTION FORMAT ACT. OP	OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
					NSI	BI	A##	No Branch. This instruction tests the last card condition and branches to the instruction specified by the A-Address if this condition exists.
BEF	B	AAA		K	BI	BI	NSI	Branch if End-of-Reel.
					NSI	BI	K##	No Branch. If a tapemark or a reflective strip is sensed during a Write Tape operation, an automatic branch is taken to the instruction specified by the A-Address.
BER	B	AAA		L	BI	BI	NSI	Branch if Tape Transmission Error.
					NSI	BI	L##	No Branch. If a transmission error occurs between the tape unit and the processor, this indicator is turned ON; and a branch occurs.
BIN	B	AAA		Q	BI	BI	NSI	Branch if Inquiry Request.
					NSI	BI	Q##	No Branch. The branch is taken if the "?" command has been executed.
BE	B	AAA		S	BI	BI	NSI	Branch if Equal Compare.
					NSI	BI	S##	No Branch. The branch is taken to the instruction specified by the A-Address if the compare indicator has been set to equal by the previous compare operation.
BL	B	AAA		T	BI	BI	NSI	Branch if Low Compare.
					NSI	BI	T##	No Branch. The branch is taken to the instruction specified by the A-Address if the compare indicator has been set to low by the previous compare operation.

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE-OP	ACT-OP	OPERANDS A	B	d	IAR	AAR	BAR	COMMENTS
BH	B	AAA		U	BI	BI	NSI	Branch if High Compare.
					NSI	BI	U###	No Branch. The branch is taken to the instruction specified by the A-Address if the compare indicator has been set to high by the previous compare operation.
BAV	B	AAA		Z	BI	BI	NSI	Branch if Arithmetic Overflow
					NSI	BI	Z###	No Branch. If an overflow condition is detected in the result field, a branch is taken to the instruction specified by the A-Address.
BC9	B	AAA		9	BI	BI	NSI	Branch if Carriage Channel No. 9.
					NSI	BI	9###	No Branch. The branch is taken to the instruction specified by the A-Address if Channel No. 9 is sensed.
BCV	B	AAA		a	BI	BI	NSI	Branch if Carriage Channel No. 12.
					NSI	BI	a###	No Branch. The branch is taken to the instruction specified by the A-Address if Channel No. 12 is sensed.
BSS	B	AAA		d	BI	BI	NSI	Branch if Sense Switch ON.
					NSI	BI	d###	No Branch.
								d-Character Sense Switch B B C C D D E E F F G G
								The branch is taken to the instruction specified by the A-Address if the Sense Switch specified by the d-Character is ON.

MNE. OP	ACT. OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
BIN	B	AAA		d	BI	BI	NSI	Branch if I/O Check Stop.
					NSI	BI	d##	No Branch.
								d-Character I/O Device
								Plus 0 Reader
								+ Printer
								Minus 0 Punch
								The branch is taken to the
								instruction specified by the
								A-Address if the I/O indica-
								tor specified by the
								d-Character is OFF.
BIN	B	AAA		d	BI	BI	NSI	Branch if Disk Indicator ON.
					NSI	BI	d##	No Branch.
								d-Character Indicator
								N Access Inoperable
								V Validity Error
								W Wrong-length record
								X Unequal-Address compare
								Y Any Disk Condition
								Access Busy
								The branch is taken to the
								instruction specified by the
								A-Address if the disk
								indicator tested is ON.
	B	AAA		d	BI	BI	NSI	Branch if I/O Busy.
					NSI	BI	d##	No Branch.
								d-Character I/O
BIN								J Channel - busy
BPB								P Printer
BPCB								R Printer Carriage
								The branch is taken to the
								instruction specified by the
								A-Address if the I/O indica-
								tor specified by the
								d-Character is busy.
BCE	B	AAA	BBB	d	BI	BI	NSI	Branch if Character Equal.

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	OPERANDS A	B	d	IAR	AAR	BAR	COMMENTS
					NSI	BI	B-1	No Branch. The branch is taken to the instruction specified by the A-Address if the character specified by the B-Address has the same bit configuration as the d-Character. The d-Character may be any configuration of the six BCD bits.
C	C	AAAA			NSI	A-La	A-La	Compare. The B-Field data is
C	C	AAAA	BBBB		NSI	A-Lw	B-Lw	compared to the A-Field data. The B-Field wordmark terminates the operation. If the B-Field is longer than the A-Field, the compare indicator is set to unequal.
MN	D				NSI	A-1	B-1	Move Numeric. A single digit
MN	D	AAAA			NSI	A-1	A-1	(8-4-2-1 bits) is moved to
MN	D	AAAA	BBBB		NSI	A-1	B-1	the B-Address. The zone bits are undisturbed.
MCE	E	AAAA	BBBB		NSI	A-La	B-Lb	Move Character and Edit. The A-Field data is modified by the B-Field edit control word. With zero suppression the A-Address register contains the location of the control zero + 1.
CC	F			d	NSI	dbb	dbb	Control Carriage.
CC	F	AAAA		d	BI	BI	NSI	Control Carriage and Branch. d-Character Immediate Skip to

Blank	Don't Skip
1	Channel 1
2	Channel 2
3	Channel 3
4	Channel 4
5	Channel 5
6	Channel 6
7	Channel 7
8	Channel 8
9	Channel 9

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	OPERANDS A	B	d	IAR	AAR	BAR	COMMENTS
								0 Channel 10
								# Channel 11
								@ Channel 12
								d-Character Skip After Print to
								A Channel 1
								B Channel 2
								C Channel 3
								D Channel 4
								E Channel 5
								F Channel 6
								G Channel 7
								H Channel 8
								I Channel 9
								¢ Channel 10
								. Channel 11
								☆ Channel 12
								d-Character Immediate Space
								J 1 space
								K 2 spaces
								L 3 spaces
								d-Character After Print Space
								/ 1 space
								S 2 spaces
								T 3 spaces
SBR	H	AAAA			NSI	A-3	Bp	Store B-Address register. The contents of the B-Address register are stored in a 3-position field specified by the A-Address.
SBR	H	AAAA	BBBB		NSI	A-3	Bp	Store B-Address register. The contents of the B-Address register are stored in a 3-position field specified by the A-Address. The B-Address register contains the address of this present instruction.

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	A	B	d	IAR	AAR	BAR	COMMENTS
ZS	-0	AAAA			NSI	A-La	A-La	Zero and Subtract. The sign of the A-Field is changed. A word mark must be associated with the A-Field data.
ZS	-0	AAAA	BBBB		NSI	A-Lw	B-1b	The A-Field data is subtracted from the zeroed B-Field and the sign is changed.
SS	K			-	NSI	-ØØ	-ØØ	Overlap OFF. This instruction returns the processing unit to normal operation.
SS	K	AAAA		-	NSI	BI	-ØØ	Overlap OFF and Branch. A branch is taken to the instruction specified by the A-Address.
SS	K			¤	NSI	¤ØØ	¤ØØ	Reset Overlap. This instruction resets the Overlap mode and returns the processing unit to normal.
SS	K	AAAA		¤	NSI	BI	¤ØØ	Reset Overlap and Branch. A branch is taken to the instruction specified by the A-Address.
SS	K			s	NSI	sØØ	sØØ	Overlap ON. The processing unit is set in Overlap mode.
SS	K	AAAA		s	NSI	BI	sØØ	Overlap ON and Branch. A branch is taken to the instruction specified by the A-Address.
SS	K			d	NSI	dØØ	dØØ	Select Stacker. The card that has been read or punched goes to the pocket specified by the d-Character.

d-Character	Feed	Pocket
1	Read	1
2	Read	8/2
4	Punch	4
8	Punch	8/2

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	OPERANDS A	OPERANDS B	d	IAR	AAR	BAR	COMMENTS
SS	K	AAA		d	NSI	BI	d##	Select Stacker and Branch. A branch is taken to the instruction specified by the A-Address.
LCA	L	AAA			NSI	A-La	BP-La	Load Characters to a wordmark. The A-Field data and wordmark are moved to the B-Field. The move is stopped by the A-Field wordmark. The B-Field wordmarks are cleared, if present.
LCA	L	AAA	BBB		NSI	A-La	B-La	
LU	L	%F0	BBB	R	NSI	B+6	B+7	Seek disk in Load Mode.
RDW	L	%F1	BBB	R	NSI	B+6	B+Lb+11	Read disk sectors with wordmarks.
WDW	L	%F3	BBB	W	NSI	B+6	B+Lb+11	Write disk sectors with wordmarks.
WDCW	L	%F3	BBB	W	NSI	Ap	Bp	Write disk check with wordmarks.
RDCOW	L	%F5	BBB	R	NSI	B+6	B+Lb+3	Read disk with sector count overlays with wordmarks.
WDCOW	L	%F5	BBB	W	NSI	B+6	B+Lb+8	Write disk with sector count overlays with wordmarks.
RDTW	L	%F6	BBB	R	NSI	B+6	B+11+2120	Read disk track with addresses with wordmarks.
WDTW	L	%F6	BBB	W	NSI	B+6	B+11+2120	Write disk track with addresses with wordmarks.
SDLW	L	%F7	BBB	W	NSI	B+6	B+11+Lf	Scan disk with wordmarks.
SDEW	L	%F8	BBB	W	NSI	B+6	B+11+Lf	%F7...Low or equal
SDHW	L	%F9	BBB	W	NSI	B+6	B+11+Lf	%F8...Equal
								%F9...High or equal
MCW	M	AAA			NSI	A-Lw	BP-La	Move Characters to a wordmark. The B-Address is taken from the B-Address register. The first wordmark sensed terminates the operation.
MCW	M	AAA	BBB		NSI	A-Lw	B-Lw	Move Characters to A or B wordmark. The A-Field data is unchanged by this instruction.

MNE. OP	ACT. OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
RDT	M	XF6	BBB	R	NSI	B+9	B+11+2120	Read disk track with addresses.
WDT	M	XF6	BBB	W	NSI	B+9	B+11+2120	Write disk track with addresses.
SDL	M	XF7	BBB	W	NSI	B+6	B+11+Lf	Scan disk.
SDLE	M	XF8	NNN	W	NSI	B+6	B+11+Lf	XF7...Low or equal
SDH	M	XF9	BBB	W	NSI	B+6	B+11+Lf	XF8...Equal
								XF9...High or equal
R*	M	XG1	BBB	R	NSI	BBB	B+Lb	Read a Card. The B-Address specifies the beginning address of the input card area. The data is transferred to storage until a groupmark wordmark is sensed which terminates the instruction.
P*	M	XG1	BBB	G	NSI	BBB	B+Lb	Punch a Card. The B-Address specifies the beginning address of the output card area. The data is transferred from storage to the card punch until a groupmark wordmark is sensed which then terminates the instruction.
W*	M	XY1	BBB	W	NSI	BBB	B+Lb	Write a Line. The B-Address specifies the beginning address of the print area. The data is transferred from storage to the printer until a groupmark wordmark is sensed which terminates the instruction.
SD	M	XF0	BBB	R	NSI	B+6	B+7	Seek Disk. The A-Address specifies that a seek operation is to be performed. The B-Address specifies the high-order position of the disk-control field in storage.

*1440 only

MNE-OP	INSTRUCTION FORMAT ACT-OP	OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
RD	M	XF1	BBB	R	NSI	B+6	B+Lb+11	Read Disk Sectors. The B-Address specifies the high-order position of the disk-control field. The data is read into storage beginning at the B-Address plus 10 positions and continues until a groupmark wordmark is sensed in memory or the end of sector is reached. If a groupmark is detected before reading of the record is completed, the wrong-length-record indicator is turned ON.
WD	M	XF1	BBB	W	NSI	B+6	B+Lb+11	Write Disk Sectors. The B-Address specifies the high-order position of the disk-control field. The data to be written begins at the B-Address plus 10 positions and continues until a groupmark wordmark is sensed in memory or the end of the sector is reached.
WDC	M	XF3	BBB	W	NSI	Ap	Bp	Write Disk Check. The B-Address specifies the high-order position of the disk-control field. The operation causes the data on disk to be compared against the data in memory.
RDCO	M	XF5	BBB	R	NSI	B+6	B+Lb+3	Read Disk with Sector-Count Overlay. The B-Address specifies the high-order position of the disk-control field. The number of sectors to be read is taken from the first three positions of the disk record and placed in the sector-count field.

MNE. OP	ACT. OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
WDCO	M	%F5	BBB	W	NSI	B+6	B+Lb+8	Write Disk with Sector-Count Overlay. The B-Address specifies the high-order position of the disk-control field. The sector-count field in the disk-control field is decremented by one and then written in the first three positions of the first sector written.
RT	M	%U#	BBB	R	NSI	%4#	B+Lb+1	Read Tape. The # in the A-Address signifies the tape unit from which the record is to be read. The operation is terminated by an inter-record gap on tape or a groupmark wordmark in storage.
MB	M	%B#	BBB	R	NSI	%2#	B+Lb+1	Read Binary Tape.
WT	M	%U#	BBB	W	NSI	%4#	B+Lb+1	Write Tape. The B-Address specifies the high-order position of the record in storage.
MB	M	%B#	BBB	W	NSI	%2#	B+LB+1	Write Binary Tape.
RCP	M	%T0	BBB	R	NSI	%30	B+Lb+1	Read from Console Printer. The B-Address specifies the high-order position of the B-Field into which the data from the console is to be read.
WCP	M	%T0	BBB	W	NSI	%30	B+Lb+1	Write on Console Printer. A groupmark wordmark terminates the instruction. Only ")" (25D2)" is supported as console printer control (CRLF), carriage return and line feed character.
NOP	N				NSI	Ap	Bp	No Operation. If characters without wordmarks follow this instruction they enter the A and B-Address registers. Any length allowed.

MNE. OP	ACT. OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
MRCM	P	AAA			NSI	A+La	A+La	Move Characters to recordmark or groupmark. Characters are moved from the A-Field to the B-Field. The A and B-Addresses specify the high-order positions of the fields. The operation is terminated by a recordmark or groupmark wordmark.
MRCM	P	AAA	BBB		NSI	A+La	B+La	
SAR	Q	AAA			NSI	A-3	Ap	Store A-Address Register. The contents of the A-Address register of the previous operation are stored in a 3-position field specified by the A-Address.
S	S	AAA			NSI	A-La	A-La	Subtract. The A-Field data is not affected by the instruction.
S	S	AAA	BBB		NSI	A-Lw	B-Lb	Subtract. The A-Field data is subtracted from the B-Field data and the result is stored in the B-Field. Algebraic sign control is implemented.
BSP	U	ZU#		B	NSI	X4#	B##	Backspace Tape Record. The
CB	U	ZB#		B	NSI	X2#	B##	tape unit specified by the A-Address is backspaced untit an inter-record gap on the tape is sensed.
SKP	U	ZU#		E	NSI	X4#	E##	Skip and Blank Tape. The tape unit specified by the A-Address moves forward and erases approximately seven inches of tape.
WTM	U	ZU#		M	NSI	X4#	M##	Write Tapemark. A special character (8421) is recorded following the last tape record and specifies an End-of-Reel condition.

INSTRUCTION FORMAT						ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	OPERANDS A	B	d	R	IAR	AAR	BAR	COMMENTS
RWD	U	XU#		R		NSI	X4#	R##	Rewind Tape. The tape unit specified by the A-Address rewinds its tape.
RWU	U	XU#		U		NSI	X4#	U##	Rewind Tape and Unload. At the end of the rewind, the tape unit is locked to a Not Ready status.
BWZ	V	AAAA	BBB	d		BI	BI	NSI	Branch if Wordmark and/or Zone.
						NSI	BI	B-1	No Branch.
									d-Character Condition
									1 Wordmark
									2 No Zone
									B 12-Zone (A8-bits)
									K 11-Zone (B, No A bit.)
									S Zero Zone (A, No B bit.)
									3 Either a word mark or no zone.
									C Either a word mark or 12-zone.
									L Either a word mark or 11-zone.
									T Either a word mark or zero-zone.
R	1					BI	Ap	081	Read Card.
R	1	AAAA				BI	BI	NSI	Read Card and Branch. After the card has been read, an automatic branch is taken to the instruction specified by the A-Address.
RCB	1		C			NSI	C##	481	Read Column Binary.
RCB	1	AAAA	C			NSI	BI	481	Read Column Binary and Branch.

MNE. OP	ACT. OP	INSTRUCTION FORMAT OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
W	2				NSI	Ap	333	Write a Line. The data in the print area is transferred to the printer. After the printing is completed, the printer advances one line.
W	2	AAAA			BI	BI	NSI	Write a Line and Branch. An automatic branch is taken to the instruction specified by the A-Address after the print operation is completed.
WM	2			¤	NSI	¤¤¤	333	Write Wordmarks. All wordmarks in the print area are printed as a numeric 1.
WS	2			S	NSI	S¤¤	333	Write and Suppress Space.
WM	2	AAAA		¤	BI	BI	NSI	Write Wordmarks and Branch. After the wordmarks are printed, an automatic branch is taken to the instruction specified by the A-Address.
WS	2	AAAA		S	BI	BI	NSI	Write, Suppress and Branch.
WR	3				NSI	Ap	081	Write and Read.
WRM	3			¤	NSI	¤¤¤	333	Write Wordmarks and Read.
WRS	3			S	NSI	S¤¤	333	Write Suppress and Read.
WR	3	AAAA			BI	BI	NSI	Write, Read, and Branch. After the I/O operations are completed, an automatic branch is taken to the instruction specified by the A-Address.
WRM	3	AAAA		¤	BI	¤¤¤	NSI	Write Wordmarks, Read and Branch.
WRS	3	AAAA		S	BI	S¤¤	NSI	Write Suppress, Read and Branch.
P	4				NSI	Ap	181	Punch a Card. The data in storage positions 101 through 180 is punched into a card.

INSTRUCTION FORMAT					ADDRESS REGISTERS AFTER OPERATION			
MNE. OP	ACT. OP	A	B	d	IAR	AAR	BAR	COMMENTS
P	4	AAA			BI	BI	NSI	Punch and Branch. After the punch operation is completed, an automatic branch is taken to the instruction specified by the A-Address.
RP	5				NSI	Ap	181	Read and Punch. The two operations overlap and the B-Address register may stand at 081, depending on which operation is completed first.
RP	5	AAA			BI	BI	NSI	Read, Punch, and Branch. After the I/O operations are completed, an automatic branch is taken to the instruction specified by the A-Address.
WP	6				NSI	Ap	181	Write and Punch. The print operation is completed first and then the punch operation is performed.
WPM	6			¤				Write Wordmarks and Punch.
WRS	6			S				Write Suppress and Punch.
WP	6	AAA			BI	BI	NSI	Write, Punch, and Branch. After the I/O operations are completed, an automatic branch is taken to the instruction specified by the A-Address.
WPM	6	AAA		¤				Write Wordmarks, Punch and Branch.
WPS	6	AAA		S				Write Suppress, Punch and Branch.
WRP	7				NSI	Ap	181	Write, Read, and Punch. The print operation is completed first, then the read and punch operations are performed. The B-Address register may stand at 081, depending on which operation is completed first.

MNE. OP	INSTRUCTION FORMAT ACT. OP	OPERANDS A	B	d	ADDRESS REGISTERS AFTER OPERATION IAR	AAR	BAR	COMMENTS
WRPM	7			¤				Write Wordmarks, Read and Punch.
WRPS	7			S				Write Suppress, Read and Punch.
WRPM	7	AAA		¤				Write Wordmarks, Read, Punch and Branch.
WRPS	7	AAA		S				Write Suppress, Read, Punch and Branch.
WRP	7	AAA			BI	BI	NSI	Write, Read, Punch, and Branch. After the I/O operations are completed, an automatic branch is taken to the instruction specified by the A-Address.
SRF	8				NSI	Ap	Bp	Treated as NOP.
SPF	9				NSI	Ap	Bp	Treated as NOP.

Key to Abbreviations

Abbreviation	Definition
MNE.	Autocoder Mnemonic
ACT.	Actual Opcode
A	The A-Address of the instruction.
Ap	The previous contents of the A-Address register.
B	The B-Address of the instruction.
Bp	The previous contents of the A-Address register.
BI	The Address of the next instruction if the branch occurs
d##	The d-Character is at the hundreds position; the tens and units positions are blanks.
GM	A groupmark with a wordmark.
La	The number of positions in the A-Field.
Lb	The number of positions in the B-Field.
Lf	Length of compared field or search argument.
Lp	The number of positions in the product field.
Lq	The number of positions in the quotient field.
Lw	The number of positions in the A- or B-Field, whichever is shorter.
NSI	The address of the next sequential instruction.
x	A single alphanumeric character.
#	1400 Unit Number (as used in X-opcodes).

9.1 Table of Abbreviations

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

ERRORS AND ERROR RECOVERY

SPO OPERATIONS

1. *** INVALID SPO COMMAND

   This error is received when an unknown SPO command is issued or invalid syntax was used.

   Response: Respecify the command and/or refer to section 6.

2. *** EOF NEEDS CRD, PRT, PCH, TP# OR DK#

   This error is received only when specifying the EOF command and means that the device specification was incorrect.

   Response: Respecify the command and/or refer to EOF, section 6.

CARD READER

3. *** INVALID INTOPT CARD

   This error is basically the same as Error 1, but usually means one of the following:

   a. INTOPT was not punched in the card as the first token.

   b. Separation characters (":", 2-8 punch) were left out or incorrectly punched.

   c. A specification went over a card boundary or past column 72.

   d. A wordmark character ("¬", 11-7-8 punches) was incorrectly punched.

   e. If none of the above, the error falls into the same category as Error 1.

   Response: Refer to section 7 and/or repunch the card and try again.

4. *** EOF RECEIVED FROM CARD READER

This error occurs when a null card file has an attempted read. It usually occurs when the ?END card is in the wrong place or a card has an invalid punch in column one.

Response: Put deck in card reader (making sure ?DATA <name> is the first card and ?END is the last, without invalid punched cards between) and issue a LOAd command.

PROCESSOR

---

5. *** NOT IMPLEMENTED IN THIS VERSION

This error occurs when an op code is not implemented in the current version of the interpreter.

Response: Take a dump and examine the program, and/or trace the program from the beginning.

6. *** INVALID OPCODE

This error can be caused by:

a. Missing wordmark

b. Wordmark in wrong place

c. Unrecognized op code.

Response: Check section 4 for a list of implemented op codes and/or ALTer the contents of target memory to the desired op code with a wordmark, then specify STArt.

7. *** INVALID INSTRUCTION LENGTH

This error occurs when an instruction of unknown length is encountered. It is usually due to a missing wordmark.

Response: This problem can usually be repaired with the help of the ALTer command.

8. *** ATTEMPT TO READ OUTSIDE MEMORY LIMITS
*** ATTEMPT TO WRITE OUTSIDE MEMORY LIMITS

This problem usually occurs during a processor operation that should be terminated by a groupmark with a wordmark. This message might occur when attempting to run a program written for a larger machine than was selected for TARGET.MEMORY.SIZE of the environment specification. (Refer to 1400/IEP Specifications Manual.)

Response: Usually a dump will locate the problem.

9. *** INVALID D MODIFIER

This error occurs when an unrecognizable d-character is encountered in an instruction.

10. *** EDIT ERROR - FIELD LENGTH = 1

The 1401/1440/1460 EDIT command does not allow a mask field length of one.

Response: Check the program via a dump, and make appropriate changes.

11. *** ILLEGAL 1400 ADDRESS REFERENCED

Usually, this means that a reference was made to location 0 or to an address greater than the size of the target machine.

Response: Respecify the address within the bounds of the machine.

12. *** INCORRECT FIELD LENGTH

This error is usually generated only by Multiply when the product-field length is exceeded.

Response: Check the product-field length via a dump, and make the appropriate changes.

DISK

---

13. *** DISK # IS INCORRECTLY FORMATTED

This indicates that the wrong format of disk file has been found, (or a missing disk area has been referenced).

Response: Recompile environment using 1400/IEP (at least Mark VI.1 release level) and change the file type of the one in error.

14. *** DISK # NOT SPECIFIED IN ENVIRONMENT, ACCESS INOPERABLE

This error occurs when a declaration for a particular disk drive is left out of the environment. The message will be displayed only once for each attempted access of a non-declared virtual 1311 disk. At this point, the user program will receive the "ACCESS INOPERABLE", (DISK ERROR "N"), and continue processing.

Response: A Recompiled environment, or user's, program should be designed to "poll" the virtual 1311 drives.

MAGNETIC TAPE

---

15. *** TAPE #: ATTEMPTED WRITE W/O WRITE-RING

The interpreter cannot write on the tape specified without a write-ring.

Response: Put on write-ring and issue an OK command to the interpreter.

16. *** TAPE #: NOT SPECIFIED IN ENVIRONMENT

The interpreter cannot operate a tape drive that is not in the environment.

Response: Recompile environment to include that particular tape-drive.

17. *** TAPE #: NOT READY

For some reason, the hardware is not ready to operate.

Response: Check hardware and issue an OK command to the interpreter.

18. *** TAPE-BUFFER TOO SHORT, CANNOT CONTINUE

This error is issued when a record is too large for the specified or default tape buffer (default = 4000).

Response: Recompile environment with larger tape buffer specified. It should be as large as the largest record plus at least one.

19. *** TAPE #: 7-TRACK WRITE <= 7 CHARS ATTEMPTED

Because of the B1800/B1700 hardware restrictions, a tape record of less than or equal to seven characters cannot be written or read. When reading, the record is passed over as though it were noise.

Response: The user program may choose to pad and de-pad its records on 7-track tape or perform its functions on 9-track tape where padding and de-padding are done automatically by the interpreter.

20. *** GROUPMARK/WORDMARK AT LAST MEMORY LOCATION

Since the B-Address Register must point to the Groupmark/Wordmark at the end of a tape area plus one; it cannot do so if the end of the tape area and the end of target are the same.

Response: With the use of ALTer, place a groupmark/wordmark at one location less than its current location.

21. *** TAPE #: DROPOUT ENCOUNTERED

This means that two null characters were discovered during a read. One null is treated as a parity error; three or more are treated as an IRG, until timeout occurs.

Response: Error is unrecoverable, but the interpreter will allow the user to retry reading the tape from that point. No backspace is done by the interpreter.

22. *** TAPE #: TIMEOUT ENCOUNTERED

This error occurs when 9/30 feet (depending on hardware) of blank tape are found.

Response: Specifying OK will allow the user to retry the same read op. No backspace is done by the interpreter.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

DISK RECORD FORMATS

EBCDIC MOVE and LOAD mode are totally compatible with normal state jobs (COBOL, RPG, etc.). By following the correct format as shown below, these disk files are easily accessible.

MOVE MODE EXAMPLE:

```
-------------------------------------------------------------------
I I 0 0 2 4 0 0      100 EBCDIC CHARACTERS                    M I
-------------------------------------------------------------------
                          108 CHARACTERS
```

Where:

Position 1 = "I" to identify 1400 EBCDIC formatted record
Positions 2 - 7 = 1400 Series disk address
Positions 8 - 107 = 100 EBCDIC character records
Position 108 = "M" to signify Move mode

LOAD MODE EXAMPLE:

```
-------------------------------------------------------------------
II 0 0 2 4 3 0    90 EBCDIC CHARACTERS   LEVEL VERS LEVEL VERS LI
-------------------------------------------------------------------
                          108 CHARACTERS
```

I = Identifier
002430 = Address
LEVEL = 4 characters
VERS = 1 character
L = Signifies Load mode

All fields are in 8-bit EBCDIC format. There are five logical (108 bytes) records mapped onto three 1800/1700 segments (each 180 bytes).

Note: In LOAD mode, characters with wordmarks are represented with the same format as the interpreter's internal representation. If a wordmark is there, the second bit is turned off - thereby creating a lower case letter.

BCD MOVE and LOAD mode disk records are only compatible with the freestanding emulator, and must be "created" with the "CREATE.VRTL 1311" program (See Related Documentation).

MOVE MODE EXAMPLE:

```
-------------------------------------------------------------------
|E E E E 0 0 0 0 0 1    600 BITS/100 CHARS OF DATA     FILLER  1 1|
-------------------------------------------------------------------
```

EEEE = 16 bits of identifier
000001 = 24 bits of address
600 BITS/100 CHARS OF DATA = 6 bit format
FILLER = 30 bits
11 = 2 bits of mode identification - always 1's

LOAD MODE EXAMPLE:

```
-------------------------------------------------------------------
|E E E E 0 1 9 9 9 9      630 BITS/90 CHARS OF DATA          0 0|
-------------------------------------------------------------------
```

630 BITS/90 CHARS OF DATA = 7 bit format
00 = 2 bits of mode identification - always 0

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

ENVIRONMENT INTERFACE

The 1400/IEP places various flags in particular scratchpad registers to be passed on to the 1400 Interpreter. The interpreter reads those flags and interprets them as setting on/off conditions, peripheral specifications, memory size, etc.

Currently the scratchpad assignments are as follows:

S1B .. PERIPHERAL DESCRIPTIONS

BITS...	DEVICE PRESENCE
0	CARD.READER
1	SYSIN
2	PUNCH
3	PUNCH.2
4	PRINTER
5	PRINTER.2
6	DISK.0
7	DISK.2
8	DISK.4
9	DISK.6
10	DISK.8
11	CONSOLE.BACKUP
12	TAPE1
13	TAPE2
14	TAPE3
15	TAPE4
16	TAPE5
17	TAPE6
18	READ.STACKER.1
19	READ.STACKER.2
20	BINARY.CARD.READER
21	1447.CONSOLE (IF 1 THEN 1447, IF 0 THEN 1407)
22	SPARE
23	SPARE

S3A .. TAPE.BUFFER.ADDRESS

S3B .. TARGET.MEMORY.SIZE

S5B .. DISK.FILE.TYPE (EBCDIC = 1, BCD = 0)

BITS	DISK-REFERENCE
0-18	SPARE
19	DISK.FILE.0
20	DISK.FILE.2
21	DISK.FILE.4
22	DISK.FILE.6
23	DISK.FILE.8

S10B .. INDICATORS.AND.SWITCHES

BITS	RECEIVED FROM IEP
0-6	SPARE
7	INVERTED.EDIT
8	END.1401
9	EOJ.1700
10	EOJ.DUMP
11-16	SPARE
17-23	SENSE.SWITCHES

TRACE MODE

1. Invocation is by the following MCP SPO command:

   <job-number> GT <non-zero digit>

   or as a compile/execute card option:

   TC <non-zero-digit>
   TRACE <non-zero-digit>

   (Refer to the B1800/B1700 Software Operational Guide.)

2. To disengage the trace, use:

   <job-number> GTO

   (Refer to B1800/B1700 Software Operational Guide.)

3. The following information is given for each trace line:

   A. The instruction address register in 5-digit decimal
   B. The instruction address register contents, in parenthesis, as 3-character format
   C. The first 8-characters or less of the instruction
   D. A-address register of the last executed instruction
   E. B-address register of the last executed instruction
   F. Switches A, B, C, D, E, F and G, if on
   H. Compare indicators <, =, >
   I. Arithmetic overflow, OV if on
   J. Error flags and branch indicators, including L, K, Y, X, W, V, N, and \.
   K. Index registers X1, X2 and X3
   L. 25 characters from target memory starting at the value in the A-address register
   M. 25 characters from target memory starting at the value in the B-address register

Note: One trace line will be produced for each execution of an instruction except for tape instructions. During a rewind or backspace operation, one retry of each tape op may occur until the rewind or backspace is completed.

All SPO interaction is included on the printer, when tracing. Commands typed on the SPO are prefaced by "->" on the printout, while responses from the interpreter are prefaced by "<-". Note that INTOPT cards are considered SPO input in this case.

SAMPLE TRACE

```
<-READY.
->ALT 001 ¬MXT0201W¬N123456   ¬.001¬N
<-001=¬MXT0201W¬N123456   ¬.001
->ALT 201 THIS IS A MESSAGE THAT WILL BE TYPED ON THE SPO.¬]
<-201=THIS IS A MESSAGE THAT WI
->TRANSFER 001
  00001 001 OP=MXT0201W A=    B=    SW=A     =        X1=   X2=   X3=   ************************* ****************
<-THIS IS A MESSAGE THAT WILL BE TYPED ON THE SPO.
  00009 009 OP=N123456  A=X30 B=250 SW=A     =        X1=   X2=   X3=
  00019 019 OP=.001     A=123 B=456 SW=A     =        X1=   X2=   X3=
*** HALT ***
<-IAR=00019 (019) OP=.001     A=001 B=023
->UR
  00001 001 OP=MXT0201W A=001 B=023 SW=A     =        X1=   X2=   X3=   MXT0201WN123456   .001N   N
<-THIS IS A MESSAGE THAT WILL BE TYPED ON THE SPO.
  00009 009 OP=N123456  A=X30 B=250 SW=A     =        X1=   X2=   X3=
  00019 019 OP=.001     A=123 B=456 SW=A     =        X1=   X2=   X3=
*** HALT ***
<-IAR=00019 (019) OP=.001     A=001 B=023
->*** THIS IS A COMMENT THAT I TYPED IN ON THE SPO.
*** THIS IS A COMMENT THAT I TYPED IN ON THE SPO.
->*CG THIS IS A MESSAGE, ALSO.
->EOF PRT

       / / / / / / / / / / / / / / / / / / / / / / / /

 00674 674 OP=B500      A=712 B=692 SW=A   >       X1=E8A X2=A6* X3=001 00262557002WK133Y26255700 *000255702538432672000262
 00500 500 OP=LXF1011R  A=500 B=678 SW=A   >       X1=E8A X2=A6* X3=001 LXF1011RB500>B541507RM722 WRI9A00141EOF *0002557025
 00508 508 OP=B500>     A=014 B=011 SW=A   =  Y  X X1=E8A X2=A6* X3=001 2557002                   0262557002
 00513 513 OP=B541507R  A=500 B=>   SW=A   =  Y  X X1=E8A X2=A6* X3=001 LXF1011RB500>B541507RM722 *************************
 00541 541 OP=B911N     A=541 B=521 SW=A   =  Y  X X1=E8A X2=A6* X3=001 B911NB614YB591507R,720714 M722020LXF3011WB528>B911N
 00546 546 OP=B614Y     A=911 B=N   SW=A   =  Y  X X1=E8A X2=A6* X3=001 .500S692.629      X 384326 *************************
 00614 614 OP=B915692I  A=614 B=551 SW=A   =  Y  X X1=E8A X2=A6* X3=001 B915692IA622692M722020MXF B591507R,720714A722719<72
 00622 622 OP=A622692   A=915 B=691 SW=A   =  Y  X X1=E8A X2=A6* X3=001 S692.629      X 3843267200  *00025570253843267200026
 00629 629 OP=M722020   A=621 B=691 SW=A   =  Y  X X1=E9A X2=A6* X3=001 LA622692M722020MXF0693WB6  A00025570253843267200026
 00636 636 OP=MXF0693W  A=702 B=000 SW=A   =  Y  X X1=E8A X2=A6* X3=001 538432672000262557002WK13  38432672000262557002
 00644 644 OP=B636>     A=699 B=700 SW=A   <       X1=E8A X2=A6* X3=001 702538432672000262557002W 02538432672000262557002WK
 00649 649 OP=B500      A=636 B=>   SW=A   <       X1=E8A X2=A6* X3=001 MXF0693WB636>B500S654695Y *************************
 00500 500 OP=LXF1011R  A=500 B=653 SW=A   <       X1=E8A X2=A6* X3=001 LXF1011RB500>B541507RM722 S654695Y666695Y7136936500
 00508 508 OP=B500>     A=017 B=202 SW=A   <       X1=260 X2=M06 X3=02M B000M161242M849M167590M17
 00513 513 OP=B541507R  A=500 B=>   SW=A   <       X1=260 X2=M06 X3=02M LXF1011RB500>B541507RM722 *************************
 00541 541 OP=B911N     A=541 B=521 SW=A   <       X1=260 X2=M06 X3=02M B911NB614YB591507R,720714 M722020LXF3011WB528>B911N

       / / / / / / / / / / / / / / / / / / / / / / / /

 00700 700 OP=B944C     A=925 B=L   SW=A   =       X1=201 X2=XER X3=R   M092247M097285B959E2M1022 *************************
 00705 705 OP=/250      A=944 B=C   SW=A   =       X1=201 X2=XER X3=R   2M102285/709250.945M09224 *************************
 00709 709 OP=MX84272R  A=250 B=199 SW=A   =       X1=201 X2=XER X3=R         11        11
 00717 717 OP=B4278     A=X24 B=274 SW=A   = L     X1=201 X2=XER X3=R                                  I
 00722 722 OP=M092204   A=427 B=B   SW=A   = L     X1=201 X2=XER X3=R   BT01 LS50262LS50276,26126 *************************
 00729 729 OP=M092235   A=091 B=203 SW=A   = L     X1=201 X2=XER X3=R   OXERROR       169          X
 00736 736 OP=B963L     A=091 B=234 SW=A   = L     X1=201 X2=XER X3=R   OXERROR       169          X
 00963 963 OP=M092247   A=963 B=741 SW=A   =       X1=201 X2=XER X3=R   M092247B753C8761    .754L M097285B983E2M102285/250B
 00970 970 OP=B753C     A=091 B=246 SW=A   =       X1=201 X2=XER X3=R   OXERROR       169          X           11
 00975 975 OP=B761      A=753 B=C   SW=A   =       X1=201 X2=XER X3=R   2M102285/2508427DC0881068 *************************
 00761 761 OP=/250      A=761 B=979 SW=A   =       X1=201 X2=XER X3=R   /2508427DC0881068786/B987     .754L091088E:270855A0
 00765 765 OP=B427D     A=250 B=199 SW=A   =       X1=201 X2=XER X3=R         11        11
 00770 770 OP=C088106   A=427 B=D   SW=A   =       X1=201 X2=XER X3=R   BT01 LS50262LS50276,26126 *************************
```

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INDEX