DISTRIBUTION LIST

2212 5348

B1800/B1700 SOFTWARE PRODUCT SPECIFICATIONS

Detroit

Single Copy

T. Freeman - Proc. Mgmt.
F. Schoeman - International
H. F. Hayde - International
K. Stokes - Prod. Mgmt.
S. Johnson - BMG
W. Varns - BMG
L. Atkins - BMG
D. Hill - IC, BM & SS
V. Morton - GPS, BM & SS
J. Shifman - CSG
J. G. Cleary - SSG
P. E. Fleming - Int'l F. E.
B. Dent - CSG
D. Dahm - Corp. Eng.

U.S. and Europe

Single Copy

K. Conry (Plymouth)
D. R. Bookwalter (Plymouth)
J. H. Pedersen (Plymouth)
J. Berta (Downingtown)
W. Minarcik (Paoli)
G. Smolnik (Paoli)
J. Murtaugh (Tredyffrin)
A. Kosla (McLean)
A. Lacaneta - F&SSG (McLean)
B. Bell (Malvern)
Mgr. WADC (Irvine)
R. Solt (Pasadena)
H. M. Townsend (Pasadena)
N. Cass - Pat. Atty. (Pasadena)
E. Sweaney (Mission Viejo)
E. D. Earnest (Mission Viejo)
J. J. Dowling (Westlake)
J. C. Allan (Glenrothes)
W. McKee (Cumbernauld)
I. J. Carradine (Cumbernauld)
Mgr. NPSGrp (Ruislip)
P. R. Evans (Middlesex)
J. Gerain (Pantin)
A. Iscla (Gennevieliers)
P. Corril (Seneffe)
J. C. Wery (Liege)
P. Bouvier (Liege)
J. Cazanove (Villers)
B. Hammersley (Croydon)
S. Samman (Liege)

Santa Barbara Plant

Single/Multiple

R. S. Bunker
J. Hale
R. Shobe
K. Meyers
R. Bauerle
A. van der Linden
E. Yardi
J. Darga
B. Ross-Smith
L. Thomas
J. Henige
B. Dodson
E. Munsch - 2
G. Hammond - 2
J. Casey - 1
K. King - 6

Distribution current as of 12/22/77.

DOCUMENT TRANSMITTAL

PAGE ____ OF ____

CONTROL DOC. [illegible]

SYSTEM B1800/B1700

PL FILE NO. ____

PROJ. NO. ____ COORD. ____ CHKR. J Hale TO RECORDS 2-10-78 DISTRIBUTION 2-6

DOC PFX	DOCUMENT NUMBER	REV	UNIT NUMBER	DL NO.	PL FILE NO.	ENG. RECORDS ONLY: QTY REV PAGE	COPIES EACH PAGE	SEPIAS	MT REQ'D	CD OK REQ'D		REMARKS
P. S.	2212 5348	D				40	67					
												RECEIVED FEB 10 1978 GENERAL MANAGER SANTA BARBARA PLANT
DOCUMENT TRANSMITTAL												

SBP 1085 REV. 9-71

Burroughs Corporation

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

B500 INTERPRETER

PRODUCT SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	1/5/76	J Hale	Original Issue
B	12/2/76	J Hale	Updated to Mark VI.0 Release Level 4-1 to 4-20 Variables changed to lower case in format declarations. 4-1 to 4-3 CARD READER, TAPE, and DISK subsections added. 4-11 999 and OFF options added to EOJ accept command. 4-14 ⟨address⟩ added to HLT command. 4-15 LCC command added. 5-2 @DO@ added. 5-3 Note (corresponding to @DO@) added.
C	6/29/77	J Hale	Updated to Mark VI.1 Release Level Replaced all references to B1700 with B1800/B1700 1-1 Added "The B500 Interpreter has at least...." 2-1 to 2-4 Re-arranged order of B500 Control Panel, Hardware Configurations and B500 MCP Interp Requirements Now: 1) B1800/B1700 System & MCP Requirements (Heading was B500 MCP Interp Requirements). This section rewritten. 2) Hardware Configurations 3) B500 Control Panel. Added "Load then clear then continue" to additional controls available. 3-6 Data Compress and Data Expand commands deleted from Commands Not Emulated, added to Editing Instructions 4-2 Added Invalid Control Id to Tape Error & Message Added "Only seven-track, nine-track NRZ..." 4-8 CLL - Added "It releases the device assigned to the "CARDS"..." and changed Memory is loaded from cards to Memory is loaded from the LOAD file. 4-14 LOA - Added: Release of device during load 4-16 LCC - Added (mix-index) to Format and "It is the equivalent of..." to Function
D	2/3/78	J Hale	Updated to Software Release Level Mark 7.0. All references to <mix-index> and <MX> have been changed to <job-number>. 5-7 - 5-8: Added DMP (Dump B500 Target Memory) command and a sample memory dump. 5-11 - 5-12:Updated EOJ (Terminate B500) command. 5-13 - 5-14:Added EQU (Find Equivalent Address) command.

TABLE OF CONTENTS

GENERAL 1-1
RELATED PUBLICATIONS 1-1
SYSTEM AND MCP REQUIREMENTS 2-1
HARDWARE CONFIGURATIONS 2-2
8500 CONTROL PANEL 2-3
INSTRUCTIONS EMULATED 3-1
ARITHMETIC INSTRUCTIONS 3-2
EDITING INSTRUCTIONS 3-2
CONTROL INSTRUCTIONS 3-3
INTERROGATE INSTRUCTIONS 3-3
CARD READER AND CARD PUNCH 3-4
LINE PRINTER 3-4
MAGNETIC TAPE 3-4
CONSOLE PRINTER 3-4
DISK FILE 3-4
I/O EMULATION 4-1
CARD READER 4-1
TAPE 4-2
DISK 4-2
CONSOLE EMULATION 5-1
SEND SPO DATA TO 8500 PROGRAM 5-2
CLEAR 8500 INTERPRETER 5-3
CLEAR AND CONTINUE 5-4
CLEAR AND LOAD 8500 MEMORY 5-5
CONTINUE 8500 EXECUTION 5-6
DUMP 8500 TARGET MEMORY 5-7
DISPLAY 8500 SENSE SWITCHES 5-9
SIMULATE 8500 EOF ON DEVICE 5-10
TERMINATE 8500 5-11
FIND EQUIVALENT ADDRESS 5-13
STOP 8500 EXECUTION 5-15
LOAD 8500 MEMORY 5-16
LOAD CLEAR CONTINUE 5-17
READ 8500 MEMORY 5-18
SET 8500 MEMORY TO A SINGLE CHARACTER 5-19
8500 SINGLE INSTRUCTION MODE 5-20
SET 8500 SENSE SWITCHES 5-21
WRITE 8500 MEMORY 5-22
INTERNAL CODE 6-1

GENERAL

The B500 Interpreter allows a B1800/B1700 system to emulate a B500 system and still run under the B1800/B1700 MCPII. The Interpreter processes instructions in an operating environment that is the same as that defined for the original B500 program. In addition to the programming environment of the original object program, these B500 operating features are available:

1. Halt recovery
2. B500 arithmetic logic
3. B500 comparison indicator and sense switch logic
4. B500 instruction tracing
5. B500 memory dump

The following standard MCP features are also available:

1. Source input from B1800/B1700-supported languages
2. Limited file equation
3. Pseudo readers
4. Backup print and punch file capability
5. Program dump and dump analysis

The B500 Interpreter has at least the same capabilities as the B500 Emulator. Because of B1800/B1700 MCPII considerations, some features may be different from an operational standpoint. Performance will be approximately that of the B500 Emulator.

RELATED PUBLICATIONS

Name	Number
B500 Interpreter Environmental Program	P.S. 2212 5313
B1800/B1700 Software Operational Guide	1068731

SYSTEM AND MCP REQUIREMENTS

The B500 Interpreter requires the following:

1. 64 KB S-Memory or more
2. Console Printer (SPO)
3. Card Reader
4. Line Printer
5. Storage medium (Disk File, Pack or Cartridge)

The S-Memory requirement is a minimum. Those users who will multiprogram may want to exceed 64 KB. The needed S-Memory may be calculated from these requirements.

Each B500 environment takes:

6.5 KB	for a 4.8 K B500
10.0 KB	for a 9.6 K B500
17.0 KB	for a 19.2 K B500

The above figures include standard peripherals but exclude disk/tape. Each EU of B500 disk (ignoring number of mods) takes 2.5 KB of B1800/B1700 space. Each tape station open at the same time requires .2 KB. Any tape operations require a single maximum tape buffer (up to 19.2 KB) shared by all tapes in the B500 environment. This is user specified via the B500 IEP.

Disk requirements are the same as the B500 Emulator with the exception that since each pseudo B500 EU is divided into disk areas (if created by the Interpreter), only those areas that are needed are assigned by the B1800/B1700 MCP. The B500 Interpreter is also able to view each pseudo B500 EU as a single disk area file as created by the B500 emulator.

The interpreter requires a minimum of 2KB of M-Memory on a B1720 series processor. If this interpreter is multiprogrammed with other interpreters, like SOL or COBOL, the minimum M-Memory requirements of each active interpreter should be available.

It is recommended that SPO Control-2 with a B9348 (TD801) not be used for the console printer because of possible confusion of the character set.

If B500 programs using magnetic tape are to be interpreted, the B500 interpreter requires that seven-track, nine-track NRZ, or nine-track PE Tape units be attached to the B1800/B1700 system.

HARDWARE CONFIGURATIONS

Hardware devices available for each B500 environment are as follows:

Card Reader - one only
Card Punch - one only
Printers - two
Magnetic Tape units - up to six
Disk File
Supervisory Printer
Memory - 4.8K, 9.6K, or 19.2K

In instances where the specific hardware configuration of the B1800/B1700 site does not contain adequate hardware to support full requirements of the emulated system, device equation may be made. For instance, if the B1800/B1700 site has only one printer in its hardware configuration, and two B500 printers are emulated, the second printer may be directed to a backup print file on disk or tape. Similarly, if the line printer of the B1800/B1700 site is in use by one of the multiprogrammed jobs in the mix, both printer files may be directed to backup at the discretion of the operator. Thus, any B500 environment can be run in the "background" with little or no operator intervention since pseudo-readers may be used for card input and printer files may be sent to backup.

B500 CONTROL PANEL

The following B500 console control functions are emulated via the B1800/B1700 console printer (SPO):

Sense switches 1-6
Single instruct
Read memory
Write memory
Clear
Halt
Load
Continue
Memory address

In addition, since the B500 Interpreter operates in the environment of the B1800/B1700 MCP, these additional controls are available:

End of B500 emulation
Set B500 memory to a single character
Set/Reset B500 sense switches
Clear then load
Clear then continue
Load then clear then continue
End of file

Switches and indicators not emulated by the B500 Interpreter are:

Power off
Power on
Emergency off
Bit reset

INSTRUCTIONS EMULATED

B500 instruction classes executed by the virtual machine interpreter are:

Arithmetic
Editing *
Control
Interrogate *
Card Reader and Card Punch
Line Printer
Magnetic Tape
Supervisory Printer
Disk File

* Exceptions are noted in later pages.

Instruction classes not emulated by the virtual machine interpreter are:

Paper Tape
Sorter Reader
Lister
Data Communication
Data Transmission

The B500 Interpreter executes the following instructions, listed by function and device:

ARITHMETIC INSTRUCTIONS

Addition (ADD)
Subtraction (SUB)
Multiplication (MUL)
Division (DIV)

Note: The comparison indicator is maintained as a result of the arithmetic process.

EDITING INSTRUCTIONS

Transfer (TFR)
Transfer and Branch (TCB)
Transfer Zone (TFZ)
Transfer Zone and Branch (TZB)
Mask (MSK)
Data Compress (DCC)
Data Expand (DEC)

Note: Fiscal, fiscal inverted, and alphanumeric mask commands are emulated, as are the comparison indicators.

Edit commands not emulated:

Transfer and Translate (TT1, TT2, TT3)

CONTROL INSTRUCTIONS

Address Modification (ADM)
Branch Conditional (BRC)
Branch Unconditional (BRU)
Compare Alpha Equal and Unequal (CAE, CAU)
Compare Zone Equal and Unequal (CZE, CZU)
Compare Numeric Equal and Unequal (CNE, CNU)
Halt (HLT)
No-Operation (NOP)

Note: All comparison indicators are emulated.

The Halt command is emulated through the use of the host system Console Printer. The message will appear as HALT 9<MN><AAA><BBB><CCC>2<MEMORY LOCATION>, where <M> and <N> are the M and N variants of the halt; <AAA>, <BBB>, and <CCC> are the A, B, AND C addresses respectively and <MEMORY LOCATION> is the address of the halt which is normally displayed on the original system console panel.

INTERROGATE INSTRUCTIONS

Interrogate Card Reader (ICR)(Not ready only)
Interrogate Card Punch (ICP)(No OP-ED)
Interrogate Line Printer (ILP)(No OP-ED)
Interrogate Console Printer (ISP)(No OP-ED)
Interrogate Magnetic Tape Unit Ready (IMR)(No OP-ED)
Interrogate Magnetic Tape Unit Write (IMW)(No OP-ED)
Interrogate Sense Switches (TSS)
Branch Bit Equal (BBE)
Branch Bit Unequal (BBU)
Set Bit (SBT)
Reset Bit (RSB)

Interrogate instructions not emulated:

Paper Tape Reader (IPR)
Paper Tape Punch (IPP)
Lister (IPL)

CARD READER AND CARD PUNCH

Card Read (CRD)
Card Punch (PCH)(BCL only)
Card Read/Branch busy (CRI) (When the M-Variant equals 1.)

LINE PRINTER

Print on Printer (PRT)
Skip on Printer (SKP)

MAGNETIC TAPE

Magnetic Tape Read (TRD)
Magnetic Tape Write (TWR)
Magnetic Tape Erase (TER)
Magnetic Tape Backspace (BSP)
Magnetic Tape Rewind (RWD)
Magnetic Tape Binary Read (BRD)
Magnetic Tape Binary Write (BWR)

CONSOLE PRINTER

Console Printer Read (SPR)
Console Printer Write (SPO)

DISK FILE

Disk File Write (DFW)
Disk File Read (DFR)
Disk File Check (DFC)
Disk File Interrogate (DFI)

I/O EMULATION

I/O operation codes are recognized by the virtual machine interpreter and translated for execution in native (B1800/B1700) mode. Error recognition and recovery routines are performed in native mode by the MCP (except for tape operations). If an I/O operation exception condition is detected, the Interpreter will take the appropriate branch as indicated by the op code. If no exception branch is present in the op code, an error message will be displayed on the SPO. The object program may then be permitted to continue (See CON command).

CARD READER

The INTERROGATE CARD READER (T11) instruction is handled as follows:

A. If the card file has not been opened, an attempt is made to open the file. If the file is not present (ON.FILE.MISSING), the not-ready branch (BBB) is then taken. (Caution: see Note below.)

B. If the file has been opened, the TEST.AND.WAIT bit in the IOAT is examined. If true, the BBB branch will be taken.

The CARD READ instruction with a not-ready branch (#1) is handled as follows:

A. If the card file has not been opened, an attempt is made to open the file. If the file is not present (ON.FILE.MISSING), the not-ready branch (AAA) is taken. (Caution: see note below.)

B. After the file is opened, a not ready will cause a branch to the AAA address.

C. The B1800/B1700 ?END card is used to emulate the EOF button on the B500 card reader. The card file will be closed, the reader will be made available for use by other jobs, and the BBB branch will be taken. If the B500 program attempts to read more cards after the "?END" card, it is necessary that a "?DATA" card precede these cards.

Note: The process of requesting the MCP to open a file and then taking the ON.FILE.MISSING branch is costly and should be

avoided, especially if the not-ready branch address is the address of the READ or INTERROGATE instruction itself.

TAPE

The stranger tape mask communicated to the MCP sets only the bits for exceptions which the B500 Interpreter can handle. Fatal errors cause a message to be printed indicating the error condition and the B500 instruction being executed. Fatal error conditions and their messages are:

ERROR	MESSAGE
Write lockout	"NO WRITE RING"
Timeout	"BLANK TAPE TIMEOUT"
Long record	"TAPE.BUFFER TOO SHORT"
Dropout	"CHARACTER DROPOUT"
Invalid control id	"INVALID TAPE DEVICE"

Because of restrictions of the B1800/B1700 hardware, certain tape records must be made longer. All nine-track NRZ records which are written, or erased, are made to be at least 16 characters long by padding null (2002) characters at the end of the record. Nine-track PE records are made to contain an even number of characters by padding, at most, one null character at the end of the record.

Nine-track NRZ tapes created on systems other than a B1800/B1700 must have records of at least seven characters (the noise bit in the stranger tape mask is set). It is recommended that the record contain at least 16 characters. Nine-track PE records must be created with an even number of characters in the record.

Only seven-track, nine-track NRZ and nine-track PE devices may be used for B500 tape operations. (If an operator assigns a device other than those allowed, e.g., cassette, the "INVALID TAPE DEVICE" message will be displayed. The operator must "EOF" the device and "CON".

DISK

If a user tries to read a disk segment which has not yet been written, and the segment number is larger than the highest

segment number already written, the user will receive a segment of invalid characters, i.e., question marks. If the segment number is less than the highest segment number already written, the user will get whatever happens to be on the B1800/B1700 disk, with no change.

CONSOLE EMULATION

Through the B1800/B1700 system SPO, the user can send messages to the B500 Interpreter in order to emulate B500 console control functions, to modify B500 programs, or to send data to B500 programs. This section discusses the format and function of each of those messages and lists them in alphabetical order for easy reference.

The messages are formatted according to standard B1800/B1700 series SPO message format. Presentation of that format, with a corresponding explanation of its function, is presumed to be self-explanatory to those familiar with SPO message format. However, examples are given in selected cases.

SEND SPO DATA TO B500 PROGRAM

```
*****
* ? *
*****
```

Format: <job-number>AX?<message>

Function:

The ? command is used to send SPO data to B500 programs. The <message> field is sent in its entirety to the program on the next occurrence of a "SPO READ" operation.

The character "less than" (<) cannot be sent to a B500 program since it is a B1800/B1700 backspace (erase) character.

The <message> field may be up to 71 characters in length.

Note: The "?" separates messages that are directed to the user program as opposed to messages directed to the B500 Interpreter itself.

CLEAR 8500 INTERPRETER

```
******
* CL *
******
```

Format: <job-number>AXCL

Function:

The CL command is used to emulate the clear button on the 8500 console. It sets the 8500 comparison indicator to "EQUAL", resets all disk indicators and sets the instruction address to 000.

CLEAR AND CONTINUE

```
*******
* CLC *
*******
```

Format: <job-number>AXCLC
CLC <memory location>

Function:

The CLC command is used to emulate the B500 sequence clear followed by continue. If the <memory location> field is omitted, the B500 virtual machine interpreter will continue execution at location 000. If the <memory location> field is a valid three character B500 memory address, the B500 will continue execution at the given <memory location>. If the <memory location> is not a valid B500 memory location, an INVALID SPO INPUT message will be printed on the console.

CLEAR AND LOAD 8500 MEMORY

```
*******
* CLL *
*******
```

Format: <job-number>AXCLL

Function:

The CLL command is used to emulate the B500 sequence of pushing the clear button followed by the load button. It releases the device assigned to the "CARDS" file before reading from the "LOAD" file.

The B500 memory is loaded from the "LOAD" file beginning at location 000 and continuing until either the cards are exhausted or the end of B500 memory has been reached.

If the B1800/B1700 program SWitch 0 is non-zero, the interpreter will do an implied CLL followed by an implied CLC at time of initialization. This allows programs to process without operator intervention.

Example:

```
   % COPYSUO =1 READY.
     COPYSUO =1 ACCEPT.
 1AXCLL
   % COPYSUO =1 MEMORY LOADED FROM 000 TC 070.
     COPYSUO =1 ACCEPT.
```

CONTINUE B500 EXECUTION

****
* CON *
****

Format: <job-number>AXCON
CON <memory location>

Function:

This command is used to emulate the B500 continue console button.

If the <memory location> field is omitted, the B500 virtual machine will continue at the location at which it was stopped. If the <memory location> field is a valid three character B500 memory address, the B500 virtual machine interpreter will continue execution at the given <memory location>. If, however, the <memory location> is not a valid B500 memory location, an INVALID SPO INPUT message will be printed on the SPO.

DUMP 8500 TARGET MEMORY

```
*******
* DMP *
*******
```

Format: <job-number>AXDMP
AXDMP<title>

Function:

The DMP command is used to produce a printout of a B500 target memory if, at compile time, the $SET DMP.EQU is included. Users may, if they wish, have a title printed on the memory dump.

Examples:

If the $SET DMP.EQU is not included, the following message will result:

```
SPO INPUT -----3279AXDMP THIS IS A TITLE FOR A 8500 MEMORY DUMP.
SPO OUTPUT   :  3279 = * * NOT INCLUDED IN THIS RELEASE * *
```

$SET DMP.EQU card included:

```
SPO INPUT -----123AXDMP THIS IS A TITLE FOR A 8500 MEMORY DUMP.
SPO OUTPUT   :   123 = *** DUMP NOW IN PROGRESS ***
SPO OUTPUT   :   123 = *** DUMP EOJ ***
```

A printout of target memory is then produced. A sample dump is shown on the next page.

```
* * * B500 MEMORY DUMP * * *
          THIS IS A TITLE FOR A B500 MEMORY DUMP.
00000 - 000  000000000000  010  000000000000  020  000000000000  030  000000000000  040  000000000000
00060 - 050  000000000000  060  000000000000  070  000000000000  080  000000000000  090  000000000000
00120 - 100  000000000000  110  000000000000  120  000000000000  130  000000000000  140  000000000000
00180 - 150  000000000000  160  000000000000  170  000000000000  180  000000000000  190  000000000000
00240 - 200  000000000000  210  000000000000  220  000000000000  230  000000000000  240  000000000000
00300 - 250  000000000000  260  000000000000  270  000000000000  280  000000000000  290  000000000000
00360 - 300  000000000000  310  000000000000  320  000000000000  330  000000000000  340  000000000000
00420 - 350  000000000000  360  000000000000  370  000000000000  380  000000000000  390  000000000000
00480 - 400  000000000000  410  000000000000  420  000000000000  430  000000000000  440  000000000000
00540 - 450  000000000000  460  000000000000  470  000000000000  480  000000000000  490  000000000000
00600 - 500  000000000000  510  000000000000  520  000000000000  530  000000000000  540  000000000000
00660 - 550  000000000000  560  000000000000  570  000000000000  580  000000000000  590  000000000000
00720 - 600  000000000000  610  000000000000  620  000000000000  630  000000000000  640  000000000000
00780 - 650  000000000000  660  000000000000  670  000000000000  680  000000000000  690  000000000000
00840 - 700  000000000000  710  000000000000  720  000000000000  730  000000000000  740  000000000000
00900 - 750  000000000000  760  000000000000  770  000000000000  780  000000000000  790  000000000000
00960 - 800  000000000000  810  000000000000  820  000000000000  830  000000000000  840  000000000000
01020 - 850  000000000000  860  000000000000  870  000000000000  880  000000000000  890  000000000000
01080 - 900  000000000000  910  000000000000  920  000000000000  930  000000000000  940  000000000000
01140 - 950  000000000000  960  000000000000  970  000000000000  980  000000000000  990  000000000000
01200 - +00  000000000000  +10  000000000000  +20  000000000000  +30  000000000000  +40  000000000000
01260 - +50  000000000000  +60  000000000000  +70  000000000000  +80  000000000000  +90  000000000000
01320 - A00  000000000000  A10  000000000000  A20  000000000000  A30  000000000000  A40  000000000000
01380 - A50  000000000000  A60  000000000000  A70  000000000000  A80  000000000000  A90  000000000000
01440 - B00  000000000000  B10  000000000000  B20  000000000000  B30  000000000000  B40  000000000000
01500 - B50  000000000000  B60  000000000000  B70  000000000000  B80  000000000000  B90  000000000000
01560 - C00  000000000000  C10  000000000000  C20  000000000000  C30  000000000000  C40  000000000000
01620 - C50  000000000000  C60  000000000000  C70  000000000000  C80  000000000000  C90  000000000000
01680 - D00  000000000000  D10  000000000000  D20  000000000000  D30  000000000000  D40  000000000000
01740 - D50  000000000000  D60  000000000000  D70  000000000000  D80  000000000000  D90  000000000000
01800 - E00  000000000000  E10  000000000000  E20  000000000000  E30  000000000000  E40  000000000000
01860 - E50  000000000000  E60  000000000000  E70  000000000000  E80  000000000000  E90  000000000000
01920 - F00  000000000000  F10  000000000000  F20  000000000000  F30  000000000000  F40  000000000000
01980 - F50  000000000000  F60  000000000000  F70  000000000000  F80  000000000000  F90  000000000000
02040 - G00  000000000000  G10  000000000000  G20  000000000000  G30  000000000000  G40  000000000000
02100 - G50  000000000000  G60  000000000000  G70  000000000000  G80  000000000000  G90  000000000000
02160 - H00  000000000000  H10  000000000000  H20  000000000000  H30  000000000000  H40  000000000000
02220 - H50  000000000000  H60  000000000000  H70  000000000000  H80  000000000000  H90  000000000000
02280 - I00  000000000000  I10  000000000000  I20  000000000000  I30  000000000000  I40  000000000000
02340 - I50  000000000000  I60  000000000000  I70  000000000000  I80  000000000000  I90  000000000000
02400 - !00  000000000000  !10  000000000000  !20  000000000000  !30  000000000000  !40  000000000000
02460 - !50  000000000000  !60  000000000000  !70  000000000000  !80  000000000000  !90  000000000000
02520 - J00  000000000000  J10  000000000000  J20  000000000000  J30  000000000000  J40  000000000000
02580 - J50  000000000000  J60  000000000000  J70  000000000000  J80  000000000000  J90  000000000000
02640 - K00  000000000000  K10  000000000000  K20  000000000000  K30  000000000000  K40  000000000000
02700 - K50  000000000000  K60  000000000000  K70  000000000000  K80  000000000000  K90  000000000000
02760 - L00  000000000000  L10  000000000000  L20  000000000000  L30  000000000000  L40  000000000000
02820 - L50  000000000000  L60  000000000000  L70  000000000000  L80  000000000000  L90  000000000000
02880 - M00  000000000000  M10  000000000000  M20  000000000000  M30  000000000000  M40  000000000000
02940 - M50  000000000000  M60  000000000000  M70  000000000000  M80  000000000000  M90  000000000000
03000 - N00  000000000000  N10  000000000000  N20  000000000000  N30  000000000000  N40  000000000000
03060 - N50  000000000000  N60  000000000000  N70  000000000000  N80  000000000000  N90  000000000000
03120 - O00  000000000000  O10  000000000000  O20  000000000000  O30  000000000000  O40  000000000000
03180 - O50  000000000000  O60  000000000000  O70  000000000000  O80  000000000000  O90  000000000000
03240 - P00  000000000000  P10  000000000000  P20  000000000000  P30  000000000000  P40  000000000000
03300 - P50  000000000000  P60  000000000000  P70  000000000000  P80  000000000000  P90  000000000000
03360 - Q00  000000000000  Q10  000000000000  Q20  000000000000  Q30  000000000000  Q40  000000000000
03420 - Q50  000000000000  Q60  000000000000  Q70  000000000000  Q80  000000000000  Q90  000000000000
03480 - R00  000000000000  R10  000000000000  R20  000000000000  R30  000000000000  R40  000000000000
03540 - R50  000000000000  R60  000000000000  R70  000000000000  R80  000000000000  R90  000000000000
03600 -  00  000000000000   10  000000000000   20  000000000000   30  000000000000   40  000000000000
03660 -  50  000000000000   60  000000000000   70  000000000000   80  000000000000   90  000000000000
03720 - /00  000000000000  /10  000000000000  /20  000000000000  /30  000000000000  /40  000000000000
03780 - /50  000000000000  /60  000000000000  /70  000000000000  /80  000000000000  /90  000000000000
03840 - S00  000000000000  S10  000000000000  S20  000000000000  S30  000000000000  S40  000000000000
03900 - S50  000000000000  S60  000000000000  S70  000000000000  S80  000000000000  S90  000000000000
03960 - T00  000000000000  T10  000000000000  T20  000000000000  T30  000000000000  T40  000000000000
04020 - T50  000000000000  T60  000000000000  T70  000000000000  T80  000000000000  T90  000000000000
04080 - U00  000000000000  U10  000000000000  U20  000000000000  U30  000000000000  U40  000000000000
04140 - U50  000000000000  U60  000000000000  U70  000000000000  U80  000000000000  U90  000000000000
04200 - V00  000000000000  V10  000000000000  V20  000000000000  V30  000000000000  V40  000000000000
04260 - V50  000000000000  V60  000000000000  V70  000000000000  V80  000000000000  V90  000000000000
04320 - W00  000000000000  W10  000000000000  W20  000000000000  W30  000000000000  W40  000000000000
04380 - W50  000000000000  W60  000000000000  W70  000000000000  W80  000000000000  W90  000000000000
04440 - X00  000000000000  X10  000000000000  X20  000000000000  X30  000000000000  X40  000000000000
04500 - X50  000000000000  X60  000000000000  X70  000000000000  X80  000000000000  X90  000000000000
04560 - Y00  000000000000  Y10  000000000000  Y20  000000000000  Y30  000000000000  Y40  000000000000
04620 - Y50  000000000000  Y60  000000000000  Y70  000000000000  Y80  000000000000  Y90  000000000000
04680 - Z00  000000000000  Z10  000000000000  Z20  000000000000  Z30  000000000000  Z40  000000000000
04740 - Z50  000000000000  Z60  000000000000  Z70  000000000000  Z80  000000000000  Z90  000000000000
```

Figure 5.1 Sample B500 Target Memory Dump

DISPLAY B500 SENSE SWITCHES

```
*******
* DSW *
*******
```

Format: <job-number>AXDSW

Function:

The DSW command is used to display the current B500 sense switch settings.

The sense switches are displayed in numerically descending order on the SPO unless there are no sense switches set, in which case the message NO SWITCHES SET is displayed.

Examples:

```
1AXDSW
 % B500/B500/ =1 SW=6431
```

```
1AXDSW
 % B500/B500/ =1 SW=-NO SWITCHES SET-
 B500/B500/ =1 ACCEPT.
```

SIMULATE B500 EOF ON DEVICE

```
*******
* EOF *
*******
```

Format: <job-number>AXEOF<mnemonic>

Function:

The EOF command is used to simulate an end-of-file condition on specified devices. It is useful when multiprogramming the B500 virtual machine with other B1800/B1700 jobs which require peripheral devices in use by the interpreter. If the B500 program attempts to read more cards after the card reader has been EOF-ed, it is necessary for a B1800/B1700 "?DATA" card to precede these cards.

The following devices may be freed by the EOF command:

CARD.READER	(CRD)
PUNCH	(PCH)
PRINTER1	(PR1)
PRINTER2	(PR2)
TAPE n	(TPn) where n = 1-6

Examples:

```
1AXEOFCRD
```

TERMINATE 8500

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

Format: <job-number>AXEOJ[9mn]
[OFF]

Function:

The EOJ command is used to terminate the 8500 virtual machine in a normal fashion. The EOJ option can be set (EOJ9mn) or reset (EOJOFF) from the SPO. If EOJ9mn is set and the 8500 virtual machine executes a halt machine instruction (9mn), the mn variants of the halt instruction are compared to the mn variants of the EOJ command. If the variants are equal, the B1800/B1700 job will terminate.

If B1800/B1700 program switch 9 is non-zero, EOJ999 is set. This allows batch programs to process without operator intervention.

Examples:

EOJ999

```
**** SPO OUTPUT    :   E/E/ =3498 BOJ. PP=4, MP=4 TIME = 14:30:11.4
**** SPO OUTPUT    : %  E/E/ =3498 READY.
**** SPO OUTPUT    :   E/E/ =3498 ACCEPT.
**** SPO INPUT -----3498AXEOJ999
**** SPO OUTPUT    :   E/E/ =3498 ACCEPT.
**** SPO INPUT -----3498AXWRI000999
**** SPO OUTPUT    : %  E/E/ =3498 000=999000000000
**** SPO OUTPUT    :   E/E/ =3498 ACCEPT.
**** SPO INPUT -----3498AXCON000
**** SPO OUTPUT    :   E/E/ =3498 EOJ. TIME = 14:32:33.9
```

EOJ9ST

```
**** SPO OUTPUT    :   E/E/ =3499 BOJ. PP=4, MP=4 TIME = 14:33:13.7
**** SPO OUTPUT    : %  E/E/ =3499 READY.
**** SPO OUTPUT    :   E/E/ =3499 ACCEPT.
**** SPO INPUT -----3499AXWRI0009ST
**** SPO OUTPUT    : %  E/E/ =3499 000=9ST000000000
**** SPO OUTPUT    :   E/E/ =3499 ACCEPT.
**** SPO INPUT -----3499AXCON000
```

```
**** SPO OUTPUT    : %  E/E/ =3499 HALT 9ST 000 000 000 0000
**** SPO OUTPUT    :    E/E/ =3499 ACCEPT.
**** SPO INPUT -----3499AX99AXEOJ9ST
**** SPO OUTPUT    :    E/E/ =3499 ACCEPT.
**** SPO INPUT -----3499AXCON000
**** SPO OUTPUT    :    E/E/ =3499 EOJ. TIME = 14:34:34.4

EOJ9,,

**** SPO INPUT -----3505AXEOJ9,,
**** SPO OUTPUT    :    E/E/ = 3505 ACCEPT.
**** SPO INPUT -----3505AXWRI0009,,
**** SPO OUTPUT    : %  E/E/ =3505 000=9,,000000000
**** SPO OUTPUT    :    E/E/ =3505 ACCEPT.
**** SPO INPUT -----3505AXCON000
**** SPO OUTPUT    :    E/E/ =3505 EOJ. TIME = 15:22:23.0
```

Example of EOJ:

Invalid EOJ input. EOJ must be followed by a 9 and two 8500 printable characters.

```
**** SPO INPUT -----EX E/E/
**** SPO OUTPUT    :    E/E/ =3505 BOJ. PP=4, MP=4 TIME = 15:20:12.5
**** SPO OUTPUT    : %  E/E/ =3505 READY.
**** SPO OUTPUT    :    E/E/ =3505 ACCEPT.
**** SPO INPUT -----3505AXEOJ456
**** SPO OUTPUT    : %  E/E/ =3505 ***INVALID SPO INPUT
**** SPO OUTPUT    :    E/E/ =3505 ACCEPT.
```

FIND EQUIVALENT ADDRESS

```
*******
* EQU *
*******
```

Format: <job-number>AXEQU<5-digit>
AXEQU<3-char 8500 address>

Function:

The EQU command is used to perform address translation from internal 8500 to 5-digit decimal or vice versa.

Two things should be noted.

1. At compile time, the card $SET DMP.EQU must be included.

2. The command is valid for that range of target memory that had been initially declared at environment compile time.

Examples:

This example illustrates when the card $SET DMP.EQU is not included:

```
SPO INPUT -----3279AXEQUM00
SPO OUTPUT  :  %      3279 = * * NOT INCLUDED IN THIS RELEASE * *
```

The following examples illustrate when the EQU command is used for values greater than the memory size declared (in this case, a memory size of 4.8K was initially declared):

```
SPO INPUT -----123AXEQU04800
SPO OUTPUT   : %  123 = * * ADDR REQUESTED IS OUT OF MEMORY BOUNDS *
SPO OUTPUT   : %  123 = * *  * INVALID ADDRESS EQU REQUEST * * *

SPO INPUT -----123AXEQU09600
SPO OUTPUT   : %  123 = * * ADDR REQUESTED IS OUT OF MEMORY BOUNDS *
SPO OUTPUT   : %  123 = * * * INVALID ADDRESS EQU REQUEST * * *
```

The following output occurs when the addresses requested for translation are not a 5-digit decimal or a 3-char 8500 ADDR:

```
SPO INPUT -----123AXEQU0479
SPO OUTPUT   :  123 = * * * INVALID ADDRESS EQU REQUEST * * *
```

```
SPO INPUT -----3295AXEQUAOMO
SPO OUTPUT    :  3295 = * * * INVALID ADDRESS EQU REQUEST * * *
```

The following occurs when the card $SET DMP.EQU is included, the address requested for translation is a 5-digit decimal or a 3-char 8500 ADDR and the value of address requested is within the range of the memory size declared:

```
SPO INPUT -----123AXEQU03429
SPO OUTPUT     :     123 = 8500 ADDR EQU   03429 = 059

SPO INPUT -----123AXEQUK00
SPO OUTPUT     :     123 = 8500 ADDR EQU   K00 = 02640
```

STOP B500 EXECUTION

```
*******
* HLT *
*******
```

Format: <job-number>AXHLT
HLT<address>

Function:

The HLT command is used to emulate the B500 halt console button. After printing the last executed instruction and its location, control is returned to the operator.

The instruction is printed with three leading blanks and is followed by "2<memory location>". The instruction at the memory location has already been executed at the time it is printed.

If <address> is specified, the B500 will halt after the instruction at the specified address has been executed. To reset the check for an address match, enter HLT, then CON, on the console printer.

LOAD B500 MEMORY

```
*******
* LOA *
*******
```

Format: <job-number>AXLOA
LOA<memory location>

Function:

The LOA command is used to load B500 memory.

If the <memory location> field is omitted, the B500 virtual machine interpreter will begin loading memory at location 000, otherwise memory will be loaded at the location given in the <memory location> field. It releases the device assigned to the "CARDS" file before reading from the "LOAD" file.

Memory is loaded from the "LOAD" file until either the cards are exhausted or the end of B500 memory has been reached.

The address of the last character loaded is displayed on the console printer. The next instruction pointer is the character after the last loaded. In the case where all of memory is loaded, the next instruction is at 000. After the load is complete, the B500 interpreter will display the following message to describe the load:

MEMORY LOADED FROM <xxx> TO <yyy>

LOAD CLEAR CONTINUE

```
*******
* LCC *
*******
```

Format: <job-number>LCC

Function:

The LCC command is used to load 8500 memory, clear the 8500, and continue at address 000. It is the equivalent of "LOA" followed by "CLC".

READ B500 MEMORY

```
*******
* REA *
*******
```

Format: <job-number>AXREA<memory location>

Function:

The REA command is used to display on the SPO the contents of B500 memory beginning at the location given by the <memory location> field and continuing for twelve succeeding locations.

The <memory location> field is a three character field specifying a valid B500 memory location.

If the <memory location> field is omitted, or is not three characters in length, an "INVALID INPUT" message is displayed.

If the <memory location> field is valid, the virtual machine will respond by displaying the address followed by its contents.

SET 8500 MEMORY TO A SINGLE CHARACTER

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

Format: <job-number>AXSET
SET<char>

Function:

The SET command is used to set all 8500 memory locations to the single character given in the <char> field. If the <char> field is omitted, all 8500 memory locations are set to the character "blank".

B500 SINGLE INSTRUCTION MODE

```
*******
* SIN *
*******
```

Format: <job-number>AXSIN
SIN<memory location>

Function:

The SIN command is used to enable the operator to execute the next B500 instruction and examine the instruction and memory location after execution. After displaying the instruction and its location on the B1800/B1700 SPO, the control is returned to the operator.

The instruction is printed with three leading blanks and is followed by "2<memory location>". The instruction and its memory location have already been executed at the time they are displayed.

If the <memory location> field is omitted, the next sequential instruction is executed by the SIN command, otherwise, the single instruction at the <memory location> is executed and the next instruction pointer is set to reflect the <memory location>.

SET B500 SENSE SWITCHES

```
*******
* SSW *
*******
```

Format: <job-number>AXSSW<switches>

Function:

The SSW command is used to set or display the current B500 sense switch settings.

When the <switches> field is omitted (i.e., SSW<ETX>), the sense switches are displayed in numerically descending order on the SPO.

When the <switches> field is a <valid switch group>, each switch in the group enumeration is set.

When the <switches> field is not a <valid switch group> (i.e., contains an invalid switch number), an "INVALID INPUT" message is displayed and none of the switches are changed.

A <valid switch group> is defined by:

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

<VALID SWITCH>       ::=  1/2/3/4/5/6
<DELIMITER>          ::=  BLANK/ ,
```

Note: As the above BNF implies, switch numbers may be entered in any order (i.e., SSW 1, 5, 3 <ETX>).

WRITE B500 MEMORY

```
*******
* WRI *
*******
```

Format: <job-number>AXWRI<memory location><new contents>

Function:

The WRI command is used to write the <new contents> field at the B500 memory location given by the <memory location field>.

The <memory location> field is a three-character B500 address.

The <new contents> field may be from 1 to 48 characters in length. The B500 memory will be written starting at the address given in the <memory location> field and continuing until the <new contents> field is exhausted.

If the <memory location> field contains an address such that the end of the B500 memory is reached before the <new contents> field is exhausted, memory wrap around will occur.

INTERNAL CODE

B100/200/300/500 PRINT SYMBOL	B100/200/300/400/500 INTERNAL CODE P	B	A	8	4	2	1	B1800/B1700 EQUIVALENT
Blank	1	1	1	0	0	0	0	Blank
.	0	0	1	1	0	1	0	.
[	1	0	1	1	0	1	1	[
(	1	0	1	1	1	0	1	(
<	1	0	1	1	1	1	0	<
← (≠)	0	0	1	1	1	1	1	← (SPO)
&	0	0	1	1	1	0	0	&
$	0	1	0	1	0	1	0	$
*	1	1	0	1	0	1	1	*
)	1	1	0	1	1	0	1	)
;	1	1	0	1	1	1	0	;
≤	0	1	0	1	1	1	1	¬ ↑ (SPO)
-	0	1	0	1	1	0	0	-
/	0	1	1	0	0	0	1	/
,	1	1	1	1	0	1	0	,
%	0	1	1	1	0	1	1	%
=	0	1	1	1	1	0	1	=
]	0	1	1	1	1	1	0	]
▪	1	1	1	1	1	1	1	▪
# (≠)	1	0	0	1	0	1	0	#
@ (↑)	0	0	0	1	0	1	1	@
: (≠)	0	0	0	1	1	0	1	:
> (▪)	0	0	0	1	1	1	0	>
≥ (≠)	1	0	0	1	1	1	1	* EOF-tape
+	0	0	1	0	0	0	0	+

A	1	0	1	0	0	0	1		A
B	1	0	1	0	0	1	0		B
C	0	0	1	0	0	1	1		C
D	1	0	1	0	1	0	0		D
E	0	0	1	0	1	0	1		E
F	0	0	1	0	1	1	0		F
G	1	0	1	0	1	1	1		G
H	1	0	1	1	0	0	0		H
I	0	0	1	1	0	0	1		I
x (See note)	0	1	0	0	0	0	0		@D0@
J	1	1	0	0	0	0	1		J
K	1	1	0	0	0	1	0		K
L	0	1	0	0	0	1	1		L
M	1	1	0	0	1	0	0		M
N	0	1	0	0	1	0	1		N
O	0	1	0	0	1	1	0		O
P	1	1	0	0	1	1	1		P
Q	1	1	0	1	0	0	0		Q
R	0	1	0	1	0	0	1		R
≠	1	1	1	1	1	0	0		_underscore\(SPO
S	0	1	1	0	0	1	0		S
T	1	1	1	0	0	1	1		T
U	0	1	1	0	1	0	0		U
V	1	1	1	0	1	0	1		V
W	1	1	1	0	1	1	0		W
X	0	1	1	0	1	1	1		X
Y	0	1	1	1	0	0	0		Y
Z	1	1	1	1	0	0	1		Z

0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	1	1
2	0	0	0	0	0	0	1	0	1	2
3	1	0	0	0	0	0	1	1	1	3
4	0	0	0	0	0	1	0	0	1	4
5	1	0	0	0	0	1	0	1	1	5
6	1	0	0	0	0	1	1	0	1	6
7	0	0	0	0	0	1	1	1	1	7
8	0	0	0	0	1	0	0	0	1	8
9	1	0	0	0	1	0	0	1	1	9
?	1	0	0	0	1	1	0	0	1	?

Note: A minus zero (x) is translated to 2002 on nine-track tape, allowing 81700 COBOL J-signed items to be read from that type of tape.

BURROUGHS CORPORATION
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

INDEX

? 5-1
ARITHMETIC INSTRUCTIONS 3-2
8500 CONTROL PANEL 2-3
8500 SINGLE INSTRUCTION MODE 5-20
CARD READER 4-1
CARD READER AND CARD PUNCH 3-4
CL 5-3
CLC 5-4
CLEAR AND CONTINUE 5-4
CLEAR AND LOAD 8500 MEMORY 5-5
CLEAR 8500 INTERPRETER 5-3
CLL 5-5
CON 5-6
CONSOLE EMULATION 5-1
CONSOLE PRINTER 3-4
CONTINUE 8500 EXECUTION 5-6
CONTROL INSTRUCTIONS 3-3
DISK 4-2
DISK FILE 3-4
DISPLAY 8500 SENSE SWITCHES 5-9
DSW 5-9
DUMP 8500 TARGET MEMORY 5-7
EDITING INSTRUCTIONS 3-2
EOF 5-10
EOJ 5-11
EQU 5-13
FIND EQUIVALENT ADDRESS 5-13
GENERAL 1-1
HARDWARE CONFIGURATIONS 2-2
HLT 5-15
I/O EMULATION 4-1
INSTRUCTIONS EMULATED 3-1
INTERNAL CODE 6-1
INTERROGATE INSTRUCTIONS 3-3
LCC 5-17
LINE PRINTER 3-4
LOA 5-16
LOAD 8500 MEMORY 5-16
LOAD CLEAR CONTINUE 5-17
MAGNETIC TAPE 3-4
REA 5-18
READ 8500 MEMORY 5-18
RELATED PUBLICATIONS 1-1
SET 5-19

SET B500 MEMORY TO A SINGLE CHARACTER 5-19
SET B500 SENSE SWITCHES 5-21
SIMULATE B500 EOF ON DEVICE 5-10
SIN 5-20
SSW 5-21
STOP B500 EXECUTION 5-15
SYSTEM AND MCP REQUIREMENTS 2-1
TAPE 4-2
TERMINATE B500 5-11
WRI 5-22
WRITE B500 MEMORY 5-22