


**Burroughs** 

**B 1800 / B 1700  
Systems  
HASP**

**REFERENCE MANUAL**

RELATIVE TO MARK VII.0 RELEASE

PRICED ITEM

Burroughs 

**B 1800 / B 1700**  
**Systems**  
**HASP**

REFERENCE MANUAL

RELATIVE TO MARK VII.0 RELEASE

Copyright © 1978, Burroughs Corporation, Detroit, Michigan 48232

PRICED ITEM

Burroughs believes that the software described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, can be accepted for any consequences arising out of the use of this material, including loss of profit, indirect, special, or consequential damages. There are no warranties which extend beyond the program specification.

The Customer should exercise care to assure that use of the software will be in full compliance with laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

## TABLE OF CONTENTS

Section		Page
1	<b>INTRODUCTION</b> . . . . .	1-1
	General . . . . .	1-1
	Related Documents . . . . .	1-1
	Notation Conventions . . . . .	1-1
	Left and Right Broken Brackets ( < > ) . . . . .	1-2
	Lower-Case N (n) . . . . .	1-2
	At Sign (@) . . . . .	1-2
2	<b>HASP SYSTEM OVERVIEW</b> . . . . .	2-1
	General . . . . .	2-1
	The Central System . . . . .	2-1
	HASPGEN Parameters . . . . .	2-1
	The Remote System . . . . .	2-2
3	<b>HASP/MODIFIER PROGRAM</b> . . . . .	3-1
	General . . . . .	3-1
	Functional Description . . . . .	3-1
	HASP/MODIFIER System Requirements . . . . .	3-2
	HASP/MODIFIER System Configuration . . . . .	3-2
	Operating Instructions and Information . . . . .	3-3
	Default Run-Time Parameters . . . . .	3-3
	Run-Time Parameters List . . . . .	3-3
	ADDRESS=P:C:A: . . . . .	3-3
	AUDIT . . . . .	3-3
	BUFFER=n . . . . .	3-4
	COMP.TYPE=n . . . . .	3-4
	COMPRESSION.MIN=n . . . . .	3-4
	DONT.EX . . . . .	3-4
	ID=n . . . . .	3-4
	LEASED . . . . .	3-5
	LIST . . . . .	3-5
	LIST.ONLY . . . . .	3-5
	MAX.PRT=n . . . . .	3-5
	MAX.PUN=ñ . . . . .	3-5
	MAX.RDR=n . . . . .	3-6
	MEM=nnnnnn . . . . .	3-6
	NONTRANS . . . . .	3-6
	PRINTER=nnn . . . . .	3-6
	PUNCH=nn . . . . .	3-6
	REMOTE . . . . .	3-7
	SIGNON=a . . . . .	3-7
	SIGNOFF=a . . . . .	3-7
	SPOOL . . . . .	3-7
	/*SIGNON . . . . .	3-8
	/*SIGNOFF . . . . .	3-8
	//LOGON . . . . .	3-8
	//LOGOFF . . . . .	3-8

## TABLE OF CONTENTS (Cont)

Section	Page
Overriding Default Run-Time Parameters . . . . .	3-8
Creating HASP/PARMS Disk File . . . . .	3-8
HASP/MODIFIER Program Execution . . . . .	3-9
Entering Run-Time Parameters . . . . .	3-9
< <i>mix-index</i> > AXYES . . . . .	3-10
< <i>mix-index</i> > AXR . . . . .	3-10
< <i>mix-index</i> > AX < <i>parameter list</i> > . . . . .	3-11
Sign-On Card . . . . .	3-11
Password . . . . .	3-11
HASP Modification . . . . .	3-12
File Modification . . . . .	3-12
Dynamic Memory Modification . . . . .	3-12
Buffer Size . . . . .	3-12
Maximum Printers . . . . .	3-12
Maximum Punches . . . . .	3-12
Punch Width . . . . .	3-12
Dynamic Memory Size . . . . .	3-13
HASP/SPOOL Modification . . . . .	3-13
Console Printer Output Messages . . . . .	3-13
HASP/SAVE.PARMS File . . . . .	3-15
4 <b>B 1800/B 1700 HASP PROGRAM</b> . . . . .	4-1
General . . . . .	4-1
Operating Modes . . . . .	4-1
Direct Mode . . . . .	4-1
HASP System Requirements . . . . .	4-2
Remote System Configuration – Direct Mode . . . . .	4-2
Spool Mode . . . . .	4-2
HASP System Requirements . . . . .	4-4
Remote System Configuration – Spool Mode . . . . .	4-4
Operating Instructions and Information . . . . .	4-4
Default Run-Time Parameters . . . . .	4-4
B 1800/B 1700 HASP Program Operation . . . . .	4-4
Establishing Communications with the Central System . . . . .	4-5
Direct Mode Operation . . . . .	4-6
Transmitting Information to the Central System . . . . .	4-6
Receiving Information from the Central System . . . . .	4-6
Terminating HASP – Direct Mode . . . . .	4-7
Spool Mode Operation . . . . .	4-8
Transmitting Spool Files to the Central System . . . . .	4-8
Receiving Spool Files from the Central System . . . . .	4-9
Terminating HASP – Spool Mode . . . . .	4-9
Console Keyboard Input Commands . . . . .	4-10
Central System Commands . . . . .	4-10
Local Commands . . . . .	4-11
.AUDIT . . . . .	4-11
.CLOSE . . . . .	4-11

## TABLE OF CONTENTS (Cont)

Section	Page
.FINI . . . . .	4-11
.EOFn . . . . .	4-12
.GOn . . . . .	4-12
.HRn . . . . .	4-12
.PUNCH=nn . . . . .	4-14
.San <i>&lt;filename&gt;</i> . . . . .	4-14
.NEW-SPOOL . . . . .	4-16
.SD1 <i>&lt;filename&gt;</i> . . . . .	4-16
.ST1 <i>&lt;filename&gt;</i> . . . . .	4-16
.RO <i>&lt;option&gt;</i> . . . . .	4-17
.SO <i>&lt;option&gt;</i> . . . . .	4-17
.WHAT . . . . .	4-17
Console Printer Output Messages . . . . .	4-18
Local Output Messages . . . . .	4-18
IBM/HASP Messages . . . . .	4-23
Operating Exceptions . . . . .	4-23
Channel Skips on Line Printer . . . . .	4-23
Printer/Punch Backup Capability . . . . .	4-24
Change Forms Processing . . . . .	4-24
Direct Mode . . . . .	4-24
Spool Mode . . . . .	4-24
Processing 96-Column IBM Object-Code Files . . . . .	4-25
Punching 96-Column IBM Object-Code Files . . . . .	4-25
Reading 96-Column IBM Object-Code Files . . . . .	4-25
HASP File Information . . . . .	4-26
File Number 0 – Direct Mode . . . . .	4-26
File Number 1 – Direct Mode . . . . .	4-27
File Number 2 – Direct Mode . . . . .	4-27
File Number 3 – Direct Mode . . . . .	4-27
File Number 4 – Direct Mode . . . . .	4-27
File Number 5 – Direct Mode . . . . .	4-28
File Number 6 – Direct Mode . . . . .	4-28
File Number 7 – Direct Mode . . . . .	4-28
File Number 8 – Direct Mode . . . . .	4-28
File Number 9 – Direct Mode . . . . .	4-29
File Number 10 – Direct Mode . . . . .	4-29
File Number 11 – Direct Mode . . . . .	4-29
File Number 12 – Direct Mode . . . . .	4-29
File Number 13 – Direct Mode . . . . .	4-29
File Number 14 – Direct Mode . . . . .	4-30
File Number 15 – Direct Mode . . . . .	4-30
File Number 16 – Direct Mode . . . . .	4-30
File Number 17 – Direct Mode . . . . .	4-30
File Number 18 – Direct Mode . . . . .	4-31
File Number 19 – Spool Mode . . . . .	4-31
File Number 20 – Spool Mode . . . . .	4-31

## TABLE OF CONTENTS (Cont)

Section	Page
File Number 21 – Direct Mode or Spool Mode . . . . .	4-31
File Number 22 – Direct Mode or Spool Mode . . . . .	4-32
File Number 23 – Direct Mode or Spool Mode . . . . .	4-32
Assigning RCB's to Files . . . . .	4-32
Summary Statistics Report . . . . .	4-33
5 <b>HASP/SPOOL PROGRAM . . . . .</b>	<b>5-1</b>
General . . . . .	5-1
Operating Modes . . . . .	5-1
Input Spool Creation Mode . . . . .	5-1
HASP/SPOOL System Requirements – Input Spool Mode . . . . .	5-2
Remote System Configuration – Input Spool Mode . . . . .	5-2
Output Spool Printing and Punching Mode . . . . .	5-2
HASP/SPOOL System Requirements – Output Spool Mode . . . . .	5-2
Remote System Configuration – Output Spool Mode . . . . .	5-3
Operating Instructions and Information . . . . .	5-3
Operating the HASP/SPOOL Program . . . . .	5-4
Executing the HASP/SPOOL Program . . . . .	5-4
Entry of .SR1, .SP1, or .BOTH Local Command . . . . .	5-5
Input Spool Mode Operation . . . . .	5-5
Output Spool Mode Operation . . . . .	5-7
Console Keyboard Input Commands . . . . .	5-8
.BOTH . . . . .	5-8
.CLOSE . . . . .	5-9
.SP1 < filename > . . . . .	5-9
.SR1 < filename > . . . . .	5-9
.RS1 . . . . .	5-9
Console Printer Messages . . . . .	5-9
Change Forms Processing . . . . .	5-11
Processing 96-Column IBM Object-Code Files . . . . .	5-11
Punching 96-Column IBM Object-Code Files . . . . .	5-12
Reading 96-Column IBM Object-Code Files . . . . .	5-12
HASP/SPOOL File Information . . . . .	5-13
File Number 0 – Output Spool Mode . . . . .	5-13
File Number 1 – Input Spool Mode . . . . .	5-14
File Number 2 – Input Spool Mode . . . . .	5-14
File Number 3 – Output Spool Mode . . . . .	5-14
File Number 4 – Output Spool Mode . . . . .	5-15
File Number 5 – Output Spool Mode . . . . .	5-15
File Number 6 – Output Spool Mode . . . . .	5-15
File Number 7 – Output Spool Mode . . . . .	5-16
File Number 8 – Output Spool Mode . . . . .	5-16
File Number 9 – Output Spool Mode . . . . .	5-16
File Number 10 – Output Spool Mode . . . . .	5-17
File Number 11 – Output Spool Mode . . . . .	5-17
File Number 12 – Output Spool Mode . . . . .	5-18

## TABLE OF CONTENTS (Cont)

Section	Page
File Number 13 – Output Spool Mode . . . . .	5-18
File Number 14 – Output Spool Mode . . . . .	5-18
File Number 15 – Output Spool Mode . . . . .	5-19
File Number 16 – Output Spool Mode . . . . .	5-19
File Number 17 – Output Spool Mode . . . . .	5-19
Operating Exceptions . . . . .	5-20
Channel Skips on Line Printer . . . . .	5-20
Printer/Punch Backup Capability . . . . .	5-20
Appendix A <b>GLOSSARY</b> . . . . .	A-1
Appendix B <b>HASP TRANSMISSION BLOCK FORMAT.</b> . . . . .	B-1
Typical HASP Transmission Block Format . . . . .	B-1
Details of Selected Control Characters . . . . .	B-2
Block Control Byte . . . . .	B-2
Function Control Sequence . . . . .	B-3
Record Control Byte . . . . .	B-3
Subrecord Control Byte . . . . .	B-5
String Control Byte . . . . .	B-8
Appendix C <b>TIPS AND TECHNIQUES</b> . . . . .	C-1
General . . . . .	C-1
B 1800/B 1700 Remote System Throughput . . . . .	C-1
Pad Character Strapping . . . . .	C-5
Transmitting Card Images . . . . .	C-6
Run-Time Parameter Max.RDR . . . . .	C-7
Appendix D <b>HASP CONTROL LOOP</b> . . . . .	D-1
General . . . . .	D-1
Simple Control Loop . . . . .	D-1
Complex Control Loop . . . . .	D-2
Index . . . . .	Index-1



## LIST OF ILLUSTRATIONS

Figure		Page
2-1	HASP System Hardware Components . . . . .	2-1
3-1	HASP/MODIFIER System Configuration . . . . .	3-2
4-1	Remote System Configuration – HASP Direct Mode . . . . .	4-3
4-2	Remote System Configuration – HASP Spool Mode . . . . .	4-5
5-1	Remote System Configuration – HASP/SPOOL Input Spool Mode . . . . .	5-3
5-2	Remote System Configuration – HASP/SPOOL Output Spool Mode . . . . .	5-4
D-1	Queue Buffer Location in Memory . . . . .	D-3

## LIST OF TABLES

Table		Page
3-1	Format of the HASP/SAVE.PARMS File . . . . .	3-15
4-1	RCB Assignment . . . . .	4-33

## 1. INTRODUCTION

### GENERAL

The purpose of this reference manual is to describe the B 1800/B 1700 Remote System and its use as a remote batch computer, on-line to an IBM Central System that is running the Houston Automatic Spooling Program (HASP). HASP is a program that works in conjunction with the IBM System 360/370 Operating System (OS) and controls certain input/output functions.

The B 1800/B 1700 HASP program is designed so that the B 1800/B 1700 functionally appears as an IBM System 360/30 to the IBM System 360/370 running HASP. The B 1800/B 1700 can communicate with the central system to take advantage of the computing power of the IBM System 360/370.

It is assumed that the reader has a working knowledge of the basic hardware and software functions of the B 1800/B 1700 Data Processing System.

Appendix A is a glossary of terms used in this manual.

Appendix B contains an example of the IBM HASP transmission block, including a detailed description of each item in the block.

Appendix C contains tips and techniques to aid the user in understanding and improving system throughput.

Appendix D contains a brief explanation of the control loop of HASP.

### RELATED DOCUMENTS

The following is a list of documents that are referenced in this manual:

B 1800/B 1700 Systems Software Operational Guide, Form Number 1068731.

B 1800/B 1700 Master Control Program (MCP) Reference Manual, Form Number 1088010.

Houston Automatic Spooling and Priority System, Version 3, 360D-05.1.014.

System 360 Operating System: Job Control Language Reference, Form Number GC28-6704.

B 1800/B 1700 Systems Data Communications Information Manual, Form Number 1089992.

### NOTATION CONVENTIONS

Words or letters that appear in uppercase type within the text of this manual represent one of the following:

- a. Program names.
- b. File names.
- c. Run-Time parameters

- d. Console printer input commands.
- e. Console printer output messages.
- f. Line printer messages.
- g. Line control characters (SOH, ETX, SYN, etc.).

### **Left and Right Broken Brackets (< >)**

Left and right broken brackets are used to enclose letters, digits, or letters and digits that can represent a variable, a number, run-time parameters, file names, and commands.

Example:

*<mix-index> AX <parameter list>*

### **Lower-Case N (n)**

The lower-case letter n represents a numeric character or characters.

### **At Sign (@)**

The “at” sign (@) is used to enclose hexadecimal information.

Example:

@F3@ is hexadecimal representation of the EBCDIC character 3.

## 2. HASP SYSTEM OVERVIEW

### GENERAL

The HASP system consists of an IBM System 360 or 370 central computer and one or more remote computers (the remote system) that communicate with each other over several classes of telephone lines. The user at a remote computer can use the capabilities of the central computer much as if the central computer were physically located at the remote computer site.

Data communications hardware and techniques are used with several classes of telephone lines to link the remote and central computer systems. A process called "spooling," unique to the B 1800/B 1700 among HASP remote systems, is advantageous for reducing line costs.

Figure 2-1 illustrates in block form the major hardware components of a simple HASP system whose remote computer is a B 1800/B 1700.

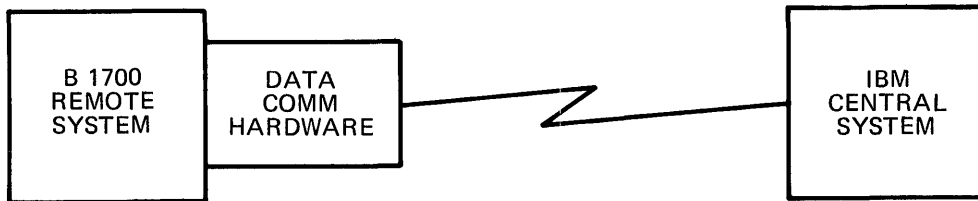


Figure 2-1. HASP System Hardware Components

### THE CENTRAL SYSTEM

HASP is a program that works in conjunction with the IBM 360/370 Operating System and controls certain input/output functions. When executing HASP, the IBM central computer logically controls the unit record devices of all remote computers as though they were local devices. The IBM central computer can simultaneously (while operating all local unit record devices) receive jobs from remote readers, place them in the queue of jobs awaiting processing, and transmit the results of previously executed jobs to several remote printers and/or punches.

### HASPGEN Parameters

HASP for the central system is a generative program created by a process called HASPGEN. Parameters are given to HASPGEN to describe each remote computer and its peripheral equipment, the number of teleprocessing buffers, the teleprocessing line identifier number, whether or not multi-leaving is wanted, and the multi-leaving buffer size when multi-leaving is specified. The following parameters are given to HASPGEN when the remote computer is a B 1800/B 1700.

\$BSCCPU	=	YES
\$MLBFSIZ	=	400 through 4095
\$NUMLNES	=	1 (minimum)
\$NUMRJE	=	1 (minimum)
\$NUMTPBF	=	2 (minimum)
\$NUMTPPR	=	1 (minimum) to 7 (maximum)
\$NUMTPPU	=	0 (minimum) to 7 (maximum)
\$NUMTPRD	=	1 (minimum) to 4 (maximum)

The combined total of the HASPGEN parameters \$NUMTPPR, \$NUMTPPU, and \$NUMTPRD must not be greater than 11 output streams.

## THE REMOTE SYSTEM

The B 1800/B 1700 is considered to be a remote batch system when operating on-line with an IBM System 360/370 central computer that is executing HASP.

A special program called HASP has been written for the B 1800/B 1700 to enable it to function as a remote computer that can communicate with an IBM central computer. This special HASP program performs the same functions as the IBM System 360/30 remote terminal program, and takes advantage of the "multi-leaving" transmission technique that is used with the IBM System 360/30 to improve throughput.

The HASP program executing on the B 1800/B 1700 remote computer is responsible for maintaining the line discipline with the central computer and managing the local peripheral devices at the remote system during a HASP session. A HASP session consists of the transmission of job streams to the central computer from a remote computer. Job streams can be received and transmitted simultaneously by the remote computer when multi-leaving is specified in the HASPGEN parameters.

Unlike the IBM remote terminal program, the B 1800/B 1700 HASP program allows the parameters that describe each remote computer to be changed without recompiling the B 1800/B 1700 HASP programs. The B 1800/B 1700 HASP/MODIFIER program validates the run-time parameters for correct syntax. When all parameters are valid, HASP/MODIFIER makes the necessary modifications to the B 1800/B 1700 HASP system and creates a disk file that contains the run-time parameters. This procedure need only be done when the remote computer configurations are changed or the central computer requires a configuration change. A detailed explanation of the HASP/MODIFIER program is presented in section 3.

A process called "spooling," unique to the B 1800/B 1700 among HASP remote systems, is advantageous for reducing line costs. The spooling process enables the user at a B 1800/B 1700 remote computer to prepare data for transmission to the central computer and accumulate blocked data as spooled disk files while off-line from the central computer. The HASP/SPOOL program operates off-line and is used for this data preparation and dissemination.

The HASP/MODIFIER, HASP, and HASP/SPOOL programs for the B 1800/B 1700 run under the control of the B 1800/B 1700 Master Control Program II (MCP II), giving them all of the resources of a computer controlled by an operating system.

A detailed explanation of the HASP/MODIFIER, HASP, and HASP/SPOOL programs is presented in the following sections.

### 3. HASP/MODIFIER PROGRAM

#### GENERAL

The HASP/MODIFIER program is responsible for verifying all run-time parameters and creating a disk file that is used by HASP and HASP/SPOOL. This execution of the HASP/MODIFIER program need only be done when the remote or the central computer configurations are changed. In that case, the HASP/MODIFIER program must be executed before either HASP or HASP/SPOOL can be executed.

The HASP/MODIFIER program is described in this section. The items described and the order in which they are presented are listed as follows.

- a. General information, including system requirements.
- b. Run-time parameters.
- c. HASP/MODIFIER program execution.
- d. Modification to HASP.
- e. Modification to HASP/SPOOL.
- f. Console printer error messages.
- g. HASP/SAVE.PARMS File.

#### FUNCTIONAL DESCRIPTION

The functions of the HASP/MODIFIER program are as follows:

- a. Validate all run-time parameters.
- b. Create a disk file labeled HASP/SAVE.PARMS that contains the run-time parameters.
- c. Modify the HASP remote terminal programs to correspond to the run-time parameters as follows:
  1. HASP is modified when in Direct Mode.
  2. HASP and/or HASP/SPOOL are modified when in the Spool Mode.
- d. Zip-execute HASP in Direct Mode and HASP/SPOOL in Spool Mode if the DONT.EX run-time parameter was not used.

Run-time parameters may be entered from the console keyboard, card reader, or disk device. Each parameter is syntax checked as it is read. If an error has been detected, that error must be corrected and all previous parameters re-entered. All parameter errors are displayed on the console printer.

When all parameters are error free, a disk file labeled HASP/SAVE.PARMS is written to disk. This file contains one 180-byte record, containing all the run-time parameters.

In addition to creating the HASP/SAVE.PARMS file, HASP/MODIFIER modifies either HASP and/or HASP/SPOOL by changing the code file to conform to the dynamic memory required. The dynamic memory required is computed by using the variables supplied by the run-time parameters MAX.PRT, MAX.PUN, and BUFFERS. The HASP program is modified if more than one punch device is required.

The ZIP to HASP or HASP/SPOOL is dependent on the run-time parameter SPOOL. If the run-time parameter SPOOL is used, then HASP/SPOOL is executed. If run-time parameter SPOOL is not used, then HASP is executed. The ZIP statement can be overridden by using the run-time parameter DONT.EX.

In order for HASP and HASP/SPOOL to execute correctly, the disk file HASP/SAVE.PARMS must be present and the dynamic memory allocation made.

### HASP/MODIFIER SYSTEM REQUIREMENTS

The following hardware and software elements are required to execute the HASP/MODIFIER program:

- a. Master Control Program II (MCP II).
- b. 8KB of main memory, excluding the MCP II memory requirements.
- c. 80 or 96 column reader.
- d. Console Printer.
- e. Disk storage (cartridge, pack, or head-per-track).

### HASP/MODIFIER SYSTEM CONFIGURATION

Figure 3-1 is a block diagram that shows the relationship of the major hardware and software elements of the system executing HASP/MODIFIER.

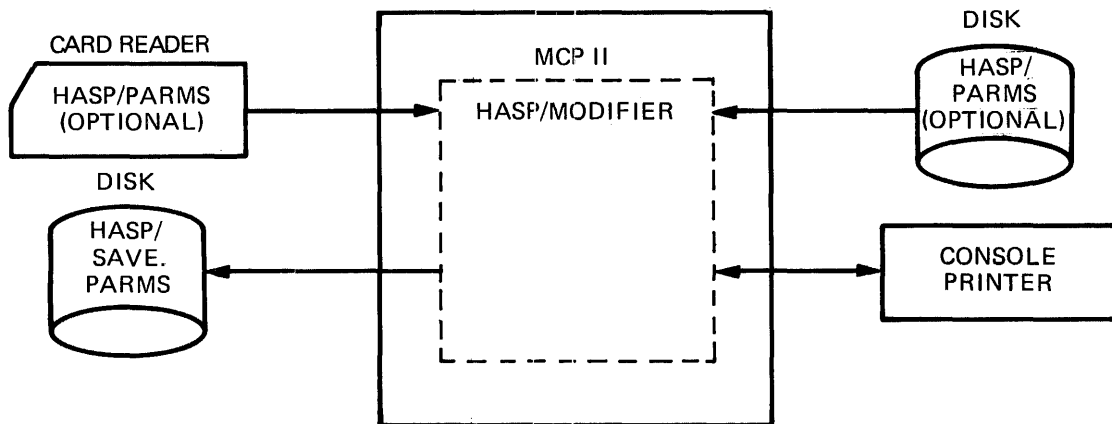


Figure 3-1. HASP/MODIFIER System Configuration



## OPERATING INSTRUCTIONS AND INFORMATION

This subsection provides operating instructions for the HASP/MODIFIER program, describes the default run-time parameters and how they can be overridden, lists and describes the console printer input commands and output messages, and includes other information related to operating the HASP/MODIFIER program.

### Default Run-Time Parameters

Unlike the IBM HASP remote terminal program, the HASP remote system allows parameters to be changed prior to execution. With the IBM HASP remote terminal program, the parameters must be stated at compile time. Therefore, any change in the remote system configuration and in the remote program, necessitates recompiling the program. The HASP remote terminal programs are more flexible in that parameters can be changed by the user prior to execution.

### RUN-TIME PARAMETERS LIST

When the HASP/MODIFIER program is compiled, a set of run-time parameters is included in the source program. At run-time, the default parameters are either used "as is" or overridden, depending on the action taken by the operator.

The following is a list of the run-time parameters, a description of each parameter, and the defaults they override.

#### ADDRESS=P:C:A:

Address of binary synchronous adapter, where:

P = one digit port number with a range of 0 through 7.

C = one or two digit channel number with a range of 0 through 15. When using a Multi-Line Control, C is 0.

A = one or two digit adapter number with a range of 0 through 15. When using a Single Line Control, A is 0.

Default: ADDRESS=7:12:0:

#### NOTE

The terminating colon is required.

### AUDIT

The file labeled DC/AUDIT.FILE is opened at the start of the HASP session, and all data communications I/O activity is monitored. A utility program named DC/AUDIT is provided to interpret the disk file and produce an audit report.

Default: No auditing.

## **BUFFER=n**

Specifies the teleprocessing buffer size in bytes, where n is a 3 or 4 digit number with a range of 132 through 4095.

Default: BUFFER=400

## **NOTE**

n must match the buffer size of the central system. The dynamic memory size of both the HASP and HASP/SPOOL programs are changed by HASP/MODIFIER if the default value of 400 is overridden.

## **COMP.TYPE=n**

Specifies the type of data compression, where n = 0, 1, 2, or 3. The data is processed according to the value of n, as follows:

- 0 = No data compression is desired.
- 1 = Compress trailing blanks.
- 2 = Compress two or more blanks occurring consecutively.
- 3 = Compress two or more blanks occurring consecutively. Also compress any occurrences of consecutive duplicate characters equal to or greater than the number specified by COMPRESSION.MIN=n.

Default: COMP.TYPE=3

## **COMPRESSION.MIN=n**

Consecutive duplicate characters numbering n or greater are compressed. The maximum allowable value for n is 31.

Default: COMPRESSION.MIN=4

## **DONT.EX**

After HASP and/or HASP/SPOOL has been modified, do not zip-execute HASP if in Direct Mode or HASP/SPOOL if in Spool Mode.

Default: Execute HASP if in Direct Mode or HASP/SPOOL if in Spool Mode.

## **ID=n**

This parameter is used to change the default identification number (REMOTE1) specified by the default sign-on card. n can be either a one or two digit number. If n is only one digit, do not include a leading zero.

For example:

Correct: ID=7

Incorrect: ID=07

If a unique sign-on card is specified, the "ID" parameter is not used, and the "ID" field passed to HASP and/or HASP/SPOOL contains whatever is in columns 22-23 of this unique sign-on card image.

Default: ID=1

### LEASED

The transmission line is a leased line.

Default: The transmission line is a switched line.

### LIST

Produces a listing on the printer of the HASP/SAVE.PARMS file just created. The LIST parameter is only valid when creating a new HASP/SAVE.PARMS file.

Default: No list.

### LIST.ONLY

Produces a listing on the printer of the HASP/SAVE.PARMS file that is already on disk.

Default: No list.

### MAX.PRT=n

n specifies the number of print files that the HASP program can receive at one time. n must be a value of 1 through 7. The sum of MAX.PRT, MAX.PUN, and MAX.RDR must not be greater than 11.

Default: MAX.PRT=1

### NOTE

n must match the number of print files specified at the central computer system.

### MAX.PUN=n

n specifies the number of punch files that the HASP program can receive at one time. n must be a value of 1 through 7. The sum of MAX.PRT, MAX.PUN, and MAX.RDR must not be greater than 11.

Default: MAX.PUN=1

### NOTE

n must be equal to or greater than the number of punch files specified at the central system.

MAX.RDR=n

n specifies the number of card image files that can be transmitted to the central computer at one time. n must be a value of 1 through 4. The sum of MAX.PRT, MAX.PUN, and MAX.RDR must not be greater than 11.

Default: MAX.RDR=1

#### NOTE

n must match the number of reader files specified at the central system.

MEM=nnnnnn

The MEM parameter is used to inform HASP/MODIFIER how much dynamic memory the HASP program is to use. *(nnnnnn)* must be equal to or greater than the minimum dynamic memory. The minimum dynamic memory can be calculated by using the following expression:

$$((\text{BUFFER} + 14) * 4 * 8 + ((\text{MAX.PRT} + \text{MAX.PUN}) * 1211) + (\text{IF PUNCH EQUAL } 96 \text{ THEN } 512 * 8))$$

If the MEM parameter is not used, the HASP/MODIFIER program calculates dynamic memory using the following expression:

$$((\text{BUFFER} + 14) * 4 * 8 + (((\text{MAX.PRT} + \text{MAX.PUN}) * 1211) * (10 * 1085)) + (\text{IF PUNCH EQUAL } 96 \text{ THEN } 512 * 8))$$

NONTRANS

Data transmission is in the non-transparent mode.

Default: Data transmission is in transparent mode.

PRINTER=nnn

nnn specifies the number of print positions for the line printer of the remote system, and must match the number specified at the central system for this remote system. nnn must be either 120 or 132.

Default: PRINTER=132

PUNCH=nn

nn specifies the number of columns that can be punched by the card punch. nn must be either 80 or 96. This parameter is only for HASP in Direct Mode. When nn is equal to 96, it refers to processing IBM object-code files on 96-column card equipment.

Default: PUNCH=80

## REMOTE

The REMOTE parameter is used to inform the HASP program to open a remote file and to process messages read from the remote file. A Network Controller and the Supervisory Message Control System (SMCS) program are required in order to use the remote terminal interface. The remote terminal interface cannot be used in the Spool Mode.

SIGNON=a

a specifies the unique sign-on card image. Card columns 8 through 80 contain the unique sign-on image. Card columns 8 through 80 are left-justified into an 80-position area and padded with trailing blanks.

Default: REMOTE1

SIGNOFF=a

a specifies the unique sign-off card image. Card columns 9 through 80 contain the unique sign-off image. Card columns 9 through 80 are left-justified into an 80-position area and padded with trailing blanks.

Default: /\*SIGNOFF

## SPOOL

This parameter specifies that the HASP and HASP/SPOOL programs are to operate in the Spool Mode.

Default: The HASP program is to operate in the Direct Mode.

## **/\*SIGNON**

Refer to the subsection titled Sign-On Card for detailed information about this parameter. Card columns 1 through 80 are transmitted to the central computer as the signon record.

Default: REMOTE1

## **/\*SIGNOFF**

The HASP program transmits the contents of card columns 1 through 80 to the central computer when the local command .CLOSE is used.

Default: /\*SIGNOFF

## **//LOGON**

The HASP program transmits the contents of card columns 1 through 80 to the central computer as the signon record.

Default: REMOTE1

## **//LOGOFF**

The HASP program transmits the contents of card columns 1 through 80 to the central computer when the local command .CLOSE is used.

Default: /\*SIGNOFF

## **OVERRIDING DEFAULT RUN-TIME PARAMETERS**

Several methods can be used when HASP/MODIFIER is executed to override the default run-time parameters. Another set or partial set of run-time parameters can be submitted to HASP/MODIFIER from one of the following:

- a. A disk file labeled HASP/PARMS.
- b. A card file labeled HASP/PARMS.
- c. The console printer.

## **CREATING HASP/PARMS DISK FILE**

Prior to a HASP session, a set of run-time parameters can be placed on a disk file labeled HASP/PARMS by the user. The HASP/PARMS disk file can be created by using the DMPALL utility program or a card-to-disk program written by the user.

There can be one or more parameter entries per card, but each parameter entry must be wholly contained on a single card (the continuation of a parameter on another card is not permitted). All parameter entries must be terminated with a blank character. All 80 columns of the card can be used.

Parameter entries are free-form in format, with the following exception: Any parameter that contains an equal sign (=) cannot contain embedded blanks (with the exception of SIGNON= and SIGNOFF=) and must be ended with a blank. The parameters /\*SIGNON, /\*SIGNOFF, //LOGON, //LOGOFF, SIGNON=a and SIGNOFF=a must be the only parameter on the card.

When a parameter error is detected, the parameter scan is terminated and the eight characters that contain the error are displayed on the console printer. The error must be corrected and the entire list of parameters resubmitted.

The following example illustrates the use of the DMPALL utility program to reproduce a disk file labeled HASP/PARMS from a card file labeled HASP/PARMS.

Example:

```
EX DMPALL
  DMPALL=1 BOJ.#9558 PR=4 TIME=10:10:05.3
  %DMPALL=1 ENTER SPECS
  DMPALL=1 ACCEPT
1AX COPY CRDDSK HASP/PARMS HASP/PARMS
  %DMPALL=1 ENTER SPECS
  DMPALL=1 ACCEPT
1AX
  DMPALL=1 EOJ.#9558 TIME=10:10:50.2

?DATA HASP/PARMS
LEASED
ADDRESS=1:00:10:
ID=87
SPOOL
?END
```

For additional information about the DMPALL utility program, refer to the System Software Operational Guide, form number 1068731.

### **HASP/MODIFIER Program Execution**

The first step is the entry of The EXECUTE statement from either the card reader or console printer. If SWITCH 0 is given a value, the appropriate entry must be included with the EXECUTE statement.

Example:

```
EXECUTE HASP/MODIFIER SWITCH 0=@F@
```

### **ENTERING RUN-TIME PARAMETERS**

At execution time of the HASP/MODIFIER program, the system operator has the option to set program control instruction attribute SWITCH 0 to @F@. The HASP/MODIFIER program interrogates SWITCH 0 and takes action based on the value found, as described on the following pages.

If SWITCH 0=0, HASP/MODIFIER attempts to read a disk file labeled HASP/PARMS, and one of the following occurs:

- a. If the HASP/PARMS file is present on disk, the syntax of the parameters is checked. If the parameter syntax is correct, HASP/MODIFIER proceeds to modify HASP and/or HASP/SPOOL to conform to the configuration specified by the run-time parameters.

If the parameter syntax is in error, the error message, PARAMETER FILE ON DISK NOT VALID, is displayed on the console printer. The action by HASP/MODIFIER is then the same as when SWITCH 0=@F@.

- b. If the HASP/PARMS file is not found on disk, the action taken is the same as when SWITCH 0=@F@.

If SWITCH 0=@F@, HASP/MODIFIER does not attempt to read the HASP/PARMS file, and the following message is displayed on the console printer:

USE DEFAULT PARAMETERS?

The HASP/MODIFIER program then does an ACCEPT and waits for the operator to enter a response on the console printer. If a unique sign-on and/or sign-off response message is used, it must be entered before any of the other response messages. The *<mix-index>* must be that of the HASP/MODIFIER program.

The following are examples of how to format response messages:

*<mix-index>*AX/\*SIGNON. . .

*<mix-index>*AX/\*SIGNOFF. . .

These responses are used to override the default sign-on and/or signoff card images.

The format of the standard HASP sign-on card image, and a description of its contents is presented in the subsection titled Sign-On Card.

Following the correct entry of a sign-on and/or sign-off response, the HASP/MODIFIER program does an ACCEPT and waits for the operator to enter either a YES, R, or *<parameter list>* response message on the console printer.

*<mix-index>* AXYES

This response indicates that the default run-time parameters are to be used. For details concerning these parameters refer to the subsection titled Default Run-Time Parameters.

*<mix-index>* AXR

This response indicates that further run-time parameters are to be read from a card file labeled HASP/PARMS.

There can be one or more parameter entries per card, but each parameter entry must be wholly contained on a single card (the continuation of a parameter on another card is not permitted). All parameter entries must be terminated with a blank character. All 80 columns of the card can be used. Parameter entries are



free-form in format with the following exception: Any parameter that contains an equal sign cannot contain embedded blanks (with the exception of the SIGNON= and SIGNOFF=), and must be ended with a blank. The parameters /\*SIGNON, /\*SIGNOFF, //LOGON, //LOGOFF, SIGNON=a, or SIGNOFF=a must be the only parameter on the card.

When a parameter error is detected, the parameter scan is terminated, and the eight characters that contain the error are displayed on the console printer. The error must be corrected and the entire set of parameters must be resubmitted.

*< mix-index >* AX *< parameter list >*

This response indicates that a parameter list of run-time parameters are entered on the console printer by the operator. The operator can enter up to 72 characters of parameters to override the default parameters. If more than 72 characters are required for any *< parameter list >* entry, one of the other methods must be used.

The parameter entry from the console printer can be in free-form format with the following exception: any parameter that contains an equal sign cannot contain embedded blanks and must be ended with a blank.

When a parameter error is detected, the parameter scan is terminated and the eight characters that contain the error are displayed on the console printer. The error must be corrected and the entire *< parameter list >* resubmitted.

If the R, YES, or sign-on/sign-off response messages are entered incorrectly, the following message is displayed on the console printer:

INVALID RESPONSE TRY AGAIN

If the *< parameter list >* response message contains an error, the eight characters in which the error was detected are displayed on the console printer, and the following message is also displayed:

INVALID PARAMETER CORRECT AND RE-ENTER.

## SIGN-ON CARD

The default run-time parameter list contains a sign-on card image with the following format:

CARD COLUMN 1	CARD COLUMN 16
/*SIGNON	REMOTE1

## PASSWORD

If the user wants to add a password to further identify the remote system a sign-on card can be included in the run-time parameters. The format is as follows:

CARD COLUMN 1	CARD COLUMN 16	CARD COLUMN 25
/*SIGNON	REMOTEn	<i>&lt; password &gt;</i>

n is the identification number assigned to the remote system at the central system.

The password is assigned at the central system, and its use is optional.

## HASP Modification

When the input run-time parameters are error free, HASP/MODIFIER makes modifications to the respective code files on disk by using the ZIP statement. For this reason the program names of HASP and HASP/SPOOL cannot be changed until after the modification has been completed.

## FILE MODIFICATION

Both HASP and HASP/SPOOL have internal file names of HASP/OUTPUT0 through HASP/OUTPUT7. The files named HASP/OUTPUT0 through HASP/OUTPUT6 contain the device type code of the printer. HASP/OUTPUT7 contains the device type code of the 96-column card punch. If more than one card punch is desired, HASP/MODIFIER starts with the external file name HASP/OUTPUT6 and changes the device type to 96-column card punch. If still more card punch devices are needed, the same procedure is performed on files HASP/OUTPUT5, HASP/OUTPUT4, and so on, until the required number of punch devices have been made available.

## DYNAMIC MEMORY MODIFICATION

After all file modification has been done, HASP/MODIFIER calculates the required dynamic memory needed for HASP and HASP/SPOOL.

The HASP and HASP/SPOOL programs are modified regardless of whether Direct Mode or Spool Mode is specified.

The following is a list of variables and a description of each that is used by the HASP/MODIFIER program to compute dynamic memory.

### Buffer Size

This is the teleprocessing buffer size for the remote terminal system, as determined by the run-time parameter BUFFER=n, where n is the size of the buffer in bytes. The default value for BUFFER is 400.

### Maximum Printers

The number of printers assigned to the remote terminal system, as determined by the run-time parameter MAX.PRT=n, where n is the number of printers. The default value for MAX.PRT is 1.

### Maximum Punches

The number of punch devices assigned to the remote terminal system, as determined by the run-time parameter MAX.PUN=n, where n is the number of punch units. The default value for MAX.PUN is 1.

### Punch Width

When the run-time parameter PUNCH=n is assigned a value of 96, then two 256-byte translate tables are required.

In addition to the preceding variables, the constant Memory Queue Size is also used in computing the dynamic memory size requirements. Memory Queue Size specifies the number of unpacked print and/or punch records that can be contained in memory before being written to disk, and always has a value of ten.

## DYNAMIC MEMORY SIZE

The dynamic memory size for the HASP program can be calculated by using the following expression:

$$((\text{BUFFER} + 14) * 4 * 8 + (\text{MAX.PRT} + \text{MAX.PUN}) * (10 * 1085 + 1211) \\ + \text{IF PUNCH EQUAL } 96 \text{ THEN } 512 * 8)$$

For example, when the default buffer size is 400 bytes, MAX.PRT=1, MAX.PUN=1, and PUNCH=80, the result of the dynamic memory size computation is 34,788 bits. For any buffer size, MAX.PRT, or MAX.PUN other than the default sizes, or when PUNCH=96, the HASP/MODIFIER program computes a different dynamic memory size requirement.

In the following example, MAX.PRT=4, MAX.PUN=2, and BUFFER=512 are used to calculate the dynamic memory size:

Example:

$$((512 + 14) * 4 * 8) + (4 + 2) * (10 * 1085 + 1211)$$

The result of the calculation is 89198 bits of dynamic memory.

## HASP/SPOOL Modification

The dynamic memory size for the HASP/SPOOL program can be calculated by using the following expression:

$$(\text{BUFFER}) * 2 * 8$$

For example, when the default buffer size of 400 is used, the result of the dynamic memory size computation is 6400 bits.

## Console Printer Output Messages

When a run-time parameter error is detected, a message is printed to indicate the faulty parameter. A list of the possible messages and a description of each follows:

COMPRESSION TYPE INVALID  
<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for COMP.TYPE= is not 0, 1, 2, or 3.

INVALID ADDRESS PARAMETER  
<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for ADDRESS= is not in the form ADDRESS=P:C:A:

INVALID BUFFER PARAMETER  
<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for BUFFER= is not in the range of 132 to 4095 inclusive, or is not followed by a blank.

INVALID COMPRESSION MIN

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for COMPRESSION.MIN= is not in the range of 2 to 31 inclusive.

INVALID ID PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for ID= is not a one or two-digit entry, or is not followed by a blank.

INVALID MAX.PRT PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for MAX.PRT is not in the range of 1 to 7 inclusive, or the sum of MAX.PRT, MAX.RDR, and MAX.PUN is greater than 11.

INVALID MAX.PUN PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for MAX.PUN is not in the range of 1 to 7 inclusive, or the sum of MAX.PRT, MAX.RDR, and MAX.PUN is greater than 11.

INVALID MAX.RDR PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for MAX.RDR is not in the range of 1 to 4 inclusive.

INVALID PRINTER PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for PRINTER= is not 120 or 132.

INVALID PUNCH PARAMETER

<nnnnnnnn> INVALID PARAMETER, PLEASE CORRECT AND RE-ENTER

Parameter for PUNCH= is not 80 or 96

INVALID RESPONSE, TRY AGAIN

None of the following parameters are entered in response to the use default parameters message: R, YES, /\*SIGNON, or <parameter list>.

<nnnnnnnn> INVALID PARAMETER PLEASE CORRECT AND RE-ENTER

Displayed, when the first error in a parameter list is found. <nnnnnnnn> is the invalid entry. All must be re-entered.

NO PARAMETER ENTERED

No entries in file HASP/PARMS.

## PARAMETER FILE ON DISK NOT VALID

HASP/PARMS disk file contains invalid parameters which are not used for run-time parameters.

## USE DEFAULT PARAMETERS?

Request for run-time parameter entry. Refer to the subsection titled RUN-TIME PARAMETERS for additional information.

## PARITY ERROR DISK FILE HASP/PARMS – FILE NOT USED

This message indicates that a disk file read parity error has occurred. The HASP/PARMS file on disk is closed, and the device type is changed to cards. The entire parameters list may still be entered through the console printer or card reader.

## PARITY ERROR ON READ, PLEASE RE-READ LAST CARD AGAIN.

This message indicates that a parity error was detected while reading the parameter list from cards. Place the last card read back into the read hopper and press the RESET/START button.

## HASP/SAVE.PARMS File

The HASP/MODIFIER program creates a disk file which contains the run-time parameters used by HASP and HASP/SPOOL. The format of this file is described in table 3-1.

Table 3-1. Format of the HASP/SAVE.PARMS File

Field Name	Type	Length	Description
01 RUN.OPTIONS	CHAR	180	
02 DC.BUF.SIZE	BIT	12	Contains the number of characters that can be transmitted and/or received. Minimum of 132. Maximum of 4096.
02 COMPRESSION.TYPE	BIT	2	0 = No compression. 1 = Compress trailing blanks. 2 = Compress all blanks. 3 = Compress all duplicate characters including blanks.
02 COMPRESSION.MIN	BIT	5	Number of continuous duplicate characters before compression is allowed. Minimum of 1. Maximum of 31.
02 TRANS.MODE	BIT	1	0 = Non-transparent 1 = Transparent

Table 3-1. Format of the HASP/SAVE.PARMS File (Cont)

Field Name	Type	Length	Description
02 REMOTE.ID	CHAR	2	Contains the remote computers identification number. This number is placed in the sign-on card if no unique sign-on record is used.
02 CONNECT.TYPE	BIT	2	0 = Switched Line. 1 = Leased Line.
02 MAX.READERS	BIT	3	Contains the number of card image files that can be read and transmitted to the central computer at one time. Minimum of 1. Maximum of 4.
02 MAX.PRINTERS	BIT	3	Contains the number of print streams that can be received by the remote computer at one time. Minimum of 1 Maximum of 7.
02 MAX.PUNCHES	BIT	3	Contains the number of punch streams that can be received by the remote computer at one time. Minimum of 1. Maximum of 7.
02 PRINTER.WIDTH	BIT	8	Contains the number of physical records that can be written on the printer. Either 120 or 132.
02 PUNCH.WIDTH	BIT	7	Indicates the number of columns of the punch device that the code file being sent from the central computer is to be punched on, as follows: 80 = 80 column device. 96 = 96 column device.
02 FILLER	BIT	1	Value of 0.
02 SPOOL.MODE	BIT	1	0 = Direct Mode 1 = Spool Mode
02 PORT	BIT	3	Contains the port address. Minimum of 0. Maximum of 7.
02 CHANNEL	BIT	4	Contains the channel address. Minimum of 0. Maximum of 15.
02 ADAPTER	BIT	4	Contains the adapter address. Minimum of 0. Maximum of 15.

Table 3-1. Format of the HASP/SAVE.PARMS File (Cont)

Field Name	Type	Length	Description
02 AUDITING	BIT	1	0 = No auditing 1 = Auditing
02 REMOTE	BIT	1	0 = No remote terminal interface. 1 = Remote terminal interface.
02 FILLER	BIT	4	Value of 0.
02 SIGN.ON.CARD	CHAR	80	Contains the sign-on card image.
02 SIGN.OFF.CARD	CHAR	80	Contains the sign-off card image.

## 4. B 1800/B 1700 HASP PROGRAM

### GENERAL

The HASP program executing on the B 1800/B 1700 remote computer is responsible for maintaining the line discipline with the IBM central computer and managing the local peripheral devices of the B 1800/B 1700 system.

This section describes the B 1800/B 1700 HASP program and includes operating instructions. The items described in this section are listed below in the order in which they are presented.

- a. Operating modes, including system requirements.
- b. Operating instructions.
- c. Console printer input commands.
- d. Console printer output messages.
- e. Operating exceptions.
- f. Change forms processing.
- g. File information.
- h. Summary statistics report.

### OPERATING MODES

There are two modes of operation for the B 1800/B 1700 HASP program, and they are hereafter referred to in this manual as the Direct Mode and the Spool Mode. In either mode of operation the B 1800/B 1700 remote computer is on-line with the central computer. The B 1800/B 1700 HASP program can process up to four input job streams and up to seven output job streams.

#### Direct Mode

When operating in the Direct Mode, the B 1800/B 1700 HASP program directly maintains the line discipline with the central computer and handles the local card reader, card punch, line printer, and console printer at the remote site. The B 1800/B 1700 HASP program receives input data from the card device, disk device, tape device, or console printer, blocks the input records, and transmits the blocks of data to the central system. Data records received from the central system are deblocked and routed to the appropriate card punch, line printer, or console printer.



## HASP SYSTEM REQUIREMENTS

The following hardware and software elements are required to execute the B 1800/B 1700 HASP program in the Direct Mode:

- a. Master Control Program II (MCP II).
- b. 20K bytes of main memory, excluding the MCP II memory requirements.
- c. 80 or 96 column card reader.
- d. 120 or 132 print position line printer.
- e. Console printer.
- f. Disk storage (cartridge, pack, or head-per-track).
- g. Single Line Control (SLC) or Multi-Line Control (MLC).
- h. Binary synchronous adapter.

A card punch is optional, and if included in the remote system it can be either an 80 or 96 column device.

The maximum number of local card readers is four. The input media may be changed to read from either a card device, disk device or tape device.

The maximum number of local line printers is seven. The output media may be directed to printer backup, thus eliminating the need for separate hardware devices for each print file.

The maximum number of local card punches is seven. The output media may be directed to punch backup, thus eliminating the need for separate hardware devices for each punch file.

## REMOTE SYSTEM CONFIGURATION – DIRECT MODE

Figure 4-1 is a block diagram that shows the relationship of the major hardware and software elements of a B 1800/B 1700 remote system, executing HASP in the Direct Mode.

### Spool Mode

The Spool Mode of the B 1800/B 1700 HASP program is advantageous when the data communications hardware between the remote and central computers can transfer data at a faster rate than can the peripheral devices of the remote computer. Logical records must be blocked and spooled to a disk file by the HASP/SPOOL program before the remote system goes on-line with the central computer, resulting in minimal data preparation overhead during job stream transmission and decreased line cost. Detailed information about the spooling process can be found in the section titled HASP/SPOOL.

In the Spool Mode, the B 1800/B 1700 HASP program directly maintains the line discipline with the central computer, reads spooled files and transmits the blocked data to the central computer, and receives data from the central computer. When data is received from the central computer, it is spooled directly to disk without deblocking.

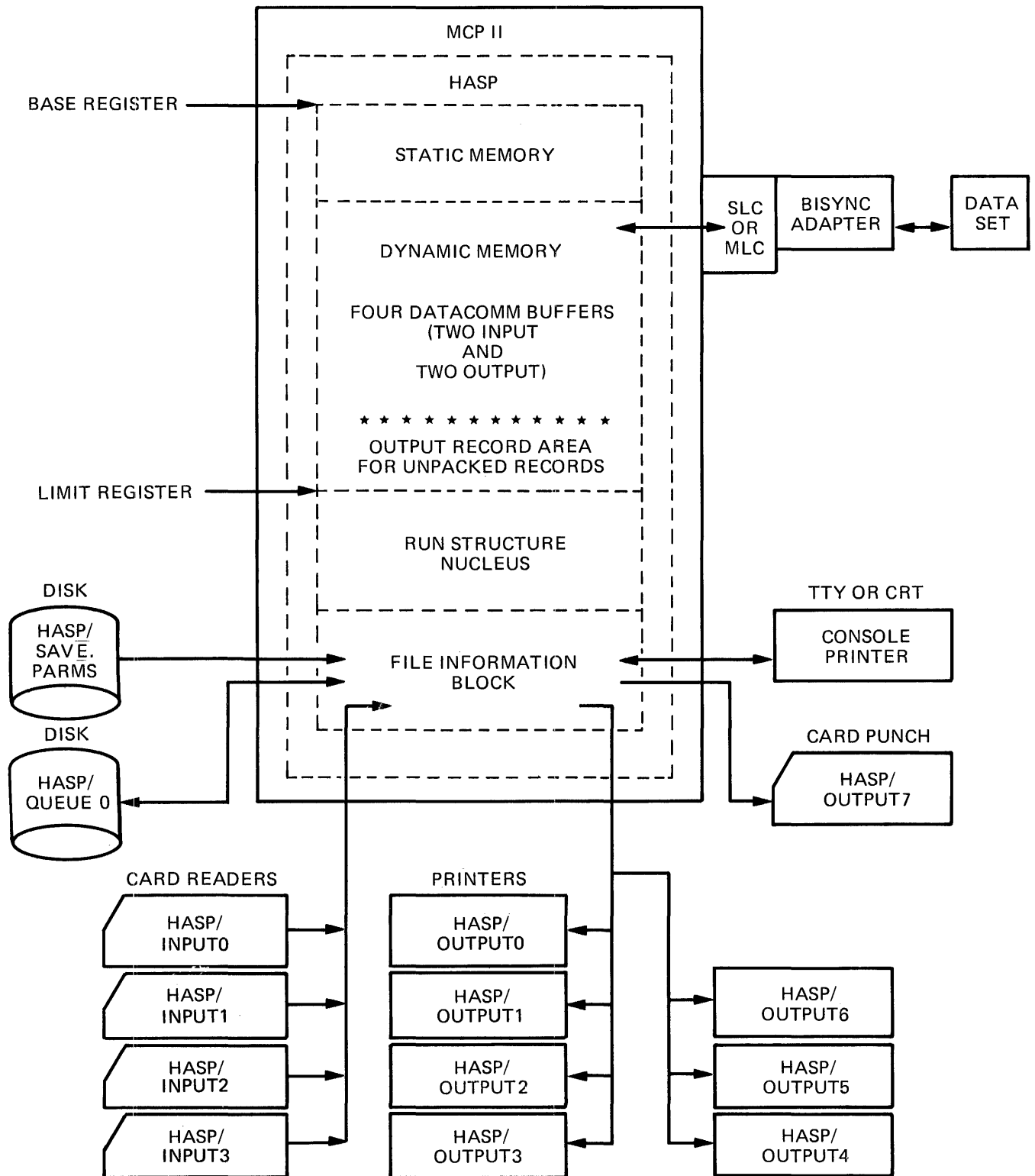


Figure 4-1. Remote System Configuration – HASP Direct Mode

## HASP SYSTEM REQUIREMENTS

The following hardware and software elements are required to execute the B 1800/B 1700 HASP program in Spool Mode.

- a. Master Control Program II (MCP II).
- b. 16K bytes of main memory, excluding the MCP II memory requirements.
- c. Console printer.
- d. Disk storage (cartridge, pack, or head-per-track).
- e. Single Line Control (SLC) or Multi-Line Control (MLC).
- f. Binary synchronous adapter.

## REMOTE SYSTEM CONFIGURATION – SPOOL MODE

Figure 4-2 is a block diagram that shows the relationship of the major hardware and software elements of a B 1800/B 1700 remote system executing HASP in the Spool Mode.

## OPERATING INSTRUCTIONS AND INFORMATION

This subsection provides operating instructions for the B 1800/B 1700 HASP program, lists and describes the console printer input commands and output messages, and includes other information related to operating the B 1800/B 1700 HASP program.

Included in this section are examples of the console printer displays following execution of the B 1800/B 1700 HASP program in both the Direct and Spool Modes.

### Default Run-Time Parameters

Unlike the IBM HASP Remote Terminal Program, the B 1800/B 1700 HASP program allows parameters to be changed prior to execution. With the IBM HASP Remote Terminal Program, the parameters must be stated at compile time. Therefore, any change in the remote system configuration and in the remote program necessitates recompiling the program. The B 1800/B 1700 HASP program is more flexible in that parameters can be changed by the user prior to program execution without recompiling. Refer to section 3 for detailed information concerning changing parameters.

### B 1800/B 1700 HASP Program Operation

This subsection sequentially describes the initiation, communications, and termination functions that occur prior to and during a HASP session. The portion concerning the initiation functions is applicable to both the Direct and Spool Modes, but the other functions are described separately for each mode.

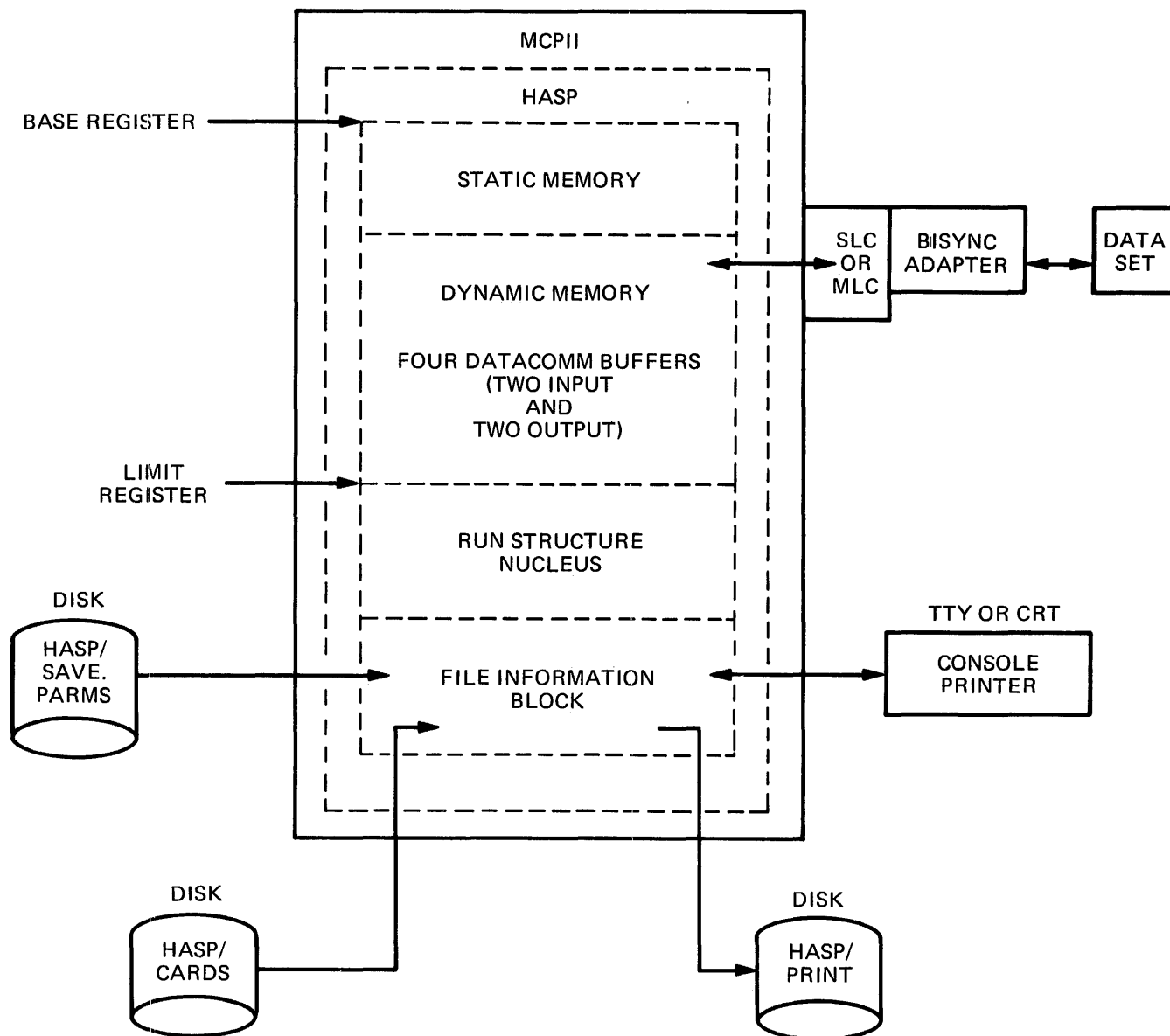


Figure 4-2. Remote System Configuration – HASP Spool Mode

#### ESTABLISHING COMMUNICATIONS WITH THE CENTRAL SYSTEM

At Beginning-of-Job, the B 1800/B 1700 HASP program:

- a. Sets the run-time parameters.
- b. Verifies whether the data communications control (SLC or MLC) and binary synchronous adapter are present.
- c. Waits for a data-set-ready condition, which is always present when leased lines are used and present when the central computer answers the call when switched lines are used.

- d. Transmits a SOHENQ message to the central computer. The SOH, ENQ, and other control characters are described in Appendix A.

When an ACKO message sequence is received from the central computer, the following message is displayed on the console printer:

#### COMMUNICATIONS ESTABLISHED

The sign-on card is then transmitted to the central computer. If the sign-on card is valid, the B 1800/B 1700 and the central computer enter a "handshaking" mode of operation until either one proceeds to transmit to the other. If the format of the sign-on card is invalid, the central computer disconnects the remote system.

#### DIRECT MODE OPERATION

##### Transmitting Information to the Central System

Job streams can be sent to the central computer by entering the appropriate local command (.SD, .SR, or .ST) on the console keyboard. Detailed information concerning local commands can be found in the subsection titled CONSOLE PRINTER INPUT COMMANDS.

After receiving a valid local command from the console printer, the B 1800/B 1700 HASP program:

- a. Requests permission to send to the central system.
- b. Receives permission from the central system to send.
- c. Opens the input file.
- d. Displays the following message on the console printer:

READER n OPEN

The number of the reader that has just been opened is n. Next, the input file is read; the data is compressed, blocked, and transmitted to the central system. When end-of-file is detected, the B 1800/B 1700 HASP program:

- a. Transmits an end-of-file message to the central system.
- b. Displays the following message on the console printer:

READER n CLOSED

The number of the reader that has just been closed is n.

##### Receiving Information from the Central System

All data and control information received from the central system is decompressed, and then goes to either the line printer, card punch, or console printer.

The B 1800/B 1700 system operator can communicate with the central computer by entering a message in the following form on the console printer:

*<mix-index>* AX *<input message>*

The *<mix-index>* must be that of the B 1800/B 1700 HASP program, and the *<input message>* consists of whatever characters are entered on the console printer by the operator. The input message must be in the format expected by the central computer. Refer to the documents listed in the subsection title RELATED DOCUMENTS in section 1 for detailed information about the input message format.

#### Terminating HASP — Direct Mode

When HASP is operating in the Direct Mode, the session can be terminated by sending a sign-off card through the card reader or by entering the following message on the console printer:

*<mix-index>* AX.CLOSE

Either of the methods described above cause HASP to:

- a. Sign-off (disconnect communications).
- b. Print a summary statistics report (HEREP).
- c. Go to End-of-Job.

#### Example:

The following example shows typical console printer input commands and output messages for a HASP session in Direct Mode. Note that any console printer input command whose first character is a period is interpreted as a local command by the B 1800/B 1700 HASP program.

## HASP Direct Mode – Console Printer Log

```
EX HASP
  HASP = 1 BOJ.# = 1357 PR=4 TIME = 16:20:11.5
  %HASP = 1 SWITCHED MODE - - NOW DIAL HOST
  %HASP = 1 COMMUNICATION ESTABLISHED
1 AX.SR
  %HASP = 1 READER 1 OPEN
  %HASP = 1 $*19.00.55 JOB 546 ON RM16.RDI - - STAR
  %HASP = 1 READER 1 CLOSED
1 AX$DJ546
  %HASP = 1 $ 19.01.24 JOB 546 STAR EXECUTING A PRIO 0
1 AX$DJ546
  %HASP = 1 $ 19.02.28 JOB 546 STAR ON RM16.PR1 PRIO 9
1 AX.SD HASP/TESTONE
  %HASP = 1 READER 1 OPEN
  %HASP = 1 $*19.02.50 JOB 547 ON RM16.RD1 - - STAR

  %HASP = 1 READER 1 CLOSED
1 AX$DJ547
  %HASP = 1 $ 19.03.34 JOB 547 STAR EXECUTING A PRIO 0
1 AX$DJ547
  %HASP = 1 $ 19.04.13 JOB 547 STAR ON RM16.PRI PRIO 9
1 AX.ST HASP/TESTTWO
  %HASP = 1 READER 1 OPEN
  %HASP = 1 $*19.04.36 JOB 548 ON RM16.RD1 - - STAR
  %HASP = 1 READER 1 CLOSED
1 AX$DJ1-999
  %HASP = 1 $ 19.05.10 JOB 548 STAR EXECUTING A PRIO 0
1 AX$DJ1-999
  %HASP = 1 $ 19.06.10 JOB 548 STAR ON RM16.PR1 PRIO 9
1 AX$DJ1-999
  %HASP = 1 19.06.43 JOB(S) NOT FOUND
1 AX.CLOSE
  HASP = 1 EOJ. # =1357 TIME = 16:28:26.1
```

### SPOOL MODE OPERATION

#### Transmitting Spool Files to the Central System

To transmit an input spool file to the central system, the operator starts an input reader by entering either the .SD1 or .ST1. local command on the console keyboard. The B 1800/B 1700 HASP program verifies whether the input spool file has been created by the HASP/SPOOL program using the identical run-time parameters as HASP. The message READER 1 OPEN is displayed and the spool file is transmitted.

After the file has been transmitted, the following message is displayed on the console printer:

```
READER 1 CLOSED
```

If additional spool files are to be transmitted, the above procedure is repeated for each file.

## Receiving Spool Files from the Central System

When executing HASP in the Spool Mode, all data and control information received from the central system is written in blocked, compressed form to the output spool disk file named HASPOUT. HASPOUT is initially labeled HASP/PRINT01. Whenever the HASPOUT file is closed and reopened, the integer portion of the file label part is incremented by 1.

The B 1800/B 1700 system operator can communicate with the central computer by entering a message in the following form on the console printer:

*<mix-index>* AX *<input message>*

The *<mix-index>* must be that of the B 1800/B 1700 HASP program, and the *<input message>* consists of whatever characters are entered on the console printer by the operator. The input message must be in the format expected by the central computer. Refer to the documents listed in the subsection titled RELATED DOCUMENTS in Section 1 for detailed information about the input message format.

Any console printer messages received from the central computer are displayed on the console printer, and written to the disk spool files for subsequent display by the HASP/SPOOL program when it processes the spool files.

When special forms are required, the B 1800/B 1700 HASP program responds automatically to the central computer and continues receiving the data stream.

## Terminating HASP – Spool Mode

When HASP is operating in Spool Mode, the session can be terminated by entering the following message on the console printer:

*<mix-index>* AX.CLOSE

Entry of this message causes HASP to:

- a. Sign-off (disconnect communications).
- b. Close the output file.
- c. Print a summary statistics report (HEREP).
- d. Go to End-of-Job.

Example:

The following example shows typical console printer input commands and output messages for a HASP session in Spool Mode. Note that any console printer input command whose first character is a period is interpreted as a local command by the B 1800/B 1700 HASP program.



## HASP Spool Mode – Console Printer Log

EX HASP

HASP = 1 BOJ. # =1354 PR = 4 TIME = 15:56:03.7

%HASP = 1 SWITCHED MODE - - NOW DIAL HOST

%HASP = 1 CONNECTION FAILED WILL RETRY

%HASP = 1 COMMUNICATION ESTABLISHED

1 AX.SD1 HASP/SPOOLONE

%HASP = 1 READER 1 OPEN

%HASP = 1 \$\*18.36.27 JOB 535 ON RM16.RD1 - - STAR

%HASP = 1 READER 1 CLOSED

1 AX\$DJ535

%HASP = 1 18.37.02 JOB 535 STAR EXECUTING A PRIO 0

1 AX\$DJ535

%HASP = 1 \$ 18.38.43 JOB 535 STAR ON RM16.PR1 PRIO 9

1 AX.NEW-SPOOL

1 AX.ST1 HASP/SPOOLTWO

%HASP = 1 READER 1 OPEN

%HASP = 1 \$\*18.39.15 JOB 537 ON RM16.RD1 - - STAR

1 AX\$DS537

%HASP = 1 HASP/PRINT01 CLOSED - HASP/PRINT02 OPENED

%HASP = 1 \$ 18.39.40 JOB 537 STAR ON RM16.RD1 PRIO 0

%HASP = 1 \$\*18.39.41 JOB 538 ON RM16.RD1 - STAR

%HASP = 1 READER 1 CLOSED

1 AX\$DJ537

%HASP = 1 \$ 18.41.01 JOB 537 STAR ON RM16.PR1 PRIO 9

1 AX.NEW-SPOOL

%HASP = 1 HASP/PRINT02 CLOSED - HASP/PRINT03 OPENED

1 AX\$DJ1-999

%HASP = 1 \$ 18.41.57 JOB 538 STAR ON RM16.PR1 PRIO 9

1 AX\$DJ1-999

%HASP = 1 \$ 18.43.14 JOB(S) NOT FOUND

1 AX.CLOSE

%HASP = 1 0002 SPOOLS TRANSMITTED

HASP = 1 EOJ. # =1354 TIME = 16:04:54.5

### Console Keyboard Input Commands

Two types of console keyboard input commands are used with the B 1800/B 1700 HASP program: Central system and local.

### CENTRAL SYSTEM COMMANDS

Central system commands are entered on the B 1800/B 1700 console keyboard to instruct the central computer system to do something. The format of the central system command is:

*<mix-index> AX <central system command>*

For the syntax of the central system commands, refer to the HASPII System Manual, form number 360D-05.1.014 (Version 3).

## LOCAL COMMANDS

Local commands are entered on the B 1800/B 1700 console keyboard to instruct the B 1800/B 1700 HASP program to perform some function. Some of the local commands are used only in Direct Mode or Spool Mode, while others are used in both modes. The format of the local command is:

*<mix-index> AX <local command>*

Refer to the B 1800/B 1700 System Software Operational Guide, form number 1068731, for detailed information about the AX command.

The following local commands can be used when the B 1800/B 1700 HASP program is executing in either Direct Mode or Spool Mode:

### .AUDIT

The function of this command is to open a disk file labeled DC/AUDIT.FILE, on which all Data Comm I/O activity is then recorded in a format recognizable by the DC/AUDIT program. To terminate the audit function, re-enter .AUDIT on the console printer.

### .CLOSE

A sign-off message in card-image form is stored in the B 1800/B 1700 HASP program. The function of the .CLOSE command is to transmit the sign-off message in card-image form to the central system. When current data streams are exhausted, the central system disconnects the line and the B 1800/B 1700 HASP program terminates.

### .FINI

The .FINI command can be used to terminate a HASP session that does not terminate normally when the .CLOSE command is used. This failure to terminate normally can occur when the central system does not positively acknowledge the sign-off message sent by the B 1800/B 1700 remote system as a result of the .CLOSE command.

Execution of the .FINI command causes the following:

- a. The summary statistics report (HEREP) is printed immediately, followed by the termination of the B 1800/B 1700 HASP program.
- b. Processing of active card and print streams is halted.
- c. Switched lines are not disconnected.
- d. No sign-off procedures are initiated.

The following local commands can be used when the B 1800/B 1700 HASP program is executing in the Direct Mode.

## .EOFn

The .EOFn local command instructs the B 1800/B 1700 HASP program to stop reading from the input reader specified by n, and send the end-of-stream indicator to the central system. The .EOFn command also deactivates the reader specified by the letter n. This command is only valid when a secondary file specifier was not found. The .SD, .SR, and .ST commands explain the use of secondary file specifiers.

## .GOn

The .GOn local command instructs the B 1800/B 1700 HASP program to continue reading from the input reader specified by n. This command is only valid when a secondary file specifier was not found. The .SD, .SR, and .ST commands explain the use of secondary file specifiers.

## .HRn

The .HRn local command allows the user to permanently assign a card reader device to the HASP program. The letter n is optional and, when used, specifies the reader number of the card reader to read from. The value of n must be from 1 to 4, inclusive. The reader number cannot exceed the value of the MAX.RDR run-time parameter.

Each card reader has a separate file whose internal file name is INPUT0, INPUT1, INPUT2, or INPUT3. The numeric portion of the name is the reader number minus one.

If the .HR local command is used without specifying the specific reader to be used, the HASP program uses the first non-active reader.

The HRn command allows the user to dedicate a card reader to the HASP program. The user can transmit a card deck to the central computer by entering the deck into the card reader, thereby eliminating the need to enter the .S local command on the console keyboard. The following rules must be adhered to when using the .HR local command:

- a. When the .HR command is initiated, the user must feed a card with the following information into the card reader assigned by the .HR local command:

?STREAM HASP/CARDS

The label HASP/CARDS cannot be changed.

- b. Any card deck can be transmitted to the central computer by placing the card deck in the input hopper of the card reader and pressing STOP/START. The first card of the card deck must have the following format:

?DATA *<filename>*

Card columns 1 through 5 must contain ?DATA. The file name can be one name or two names separated by a slash (/).

- c. The last card in the card deck must have the following format:

?END

Card columns 1 through 4 must contain ?END.

d. To release a reader from the .HR local command, enter the following card through the card reader:

```
?TERMINATE HASP/CARDS
```

Example:

```
?STREAM HASP/CARDS
?DATA CARDS
.SD MICR/JCL1
.SD MICR/FILE1
.SD MICR/JCLEND1
?END
?DATA CARDS
.SD MICR/JCL2
.SD MICR/FILE2
.SD MICR/JCL2
.SD MICR/FILE2
.SD MICR/JCLEND2
?END
?DATA CARDS
//TEST1 JOB (123456,123456,HDH),CLASS=A, TIME=8
//LIST EXEC COBUCLG, PARM.COB='LOAD'
//COB.SYSPUNCH DD SYSOUT=B
//COB.SYSIN DD *
```

(source deck)

```
/*
//LKED.SYSDUMP DD SYSOUT=A
//GO.SYSPRINT DD SYSOUT=A
//GO.SYSIN DD *
```

(input data)

```
/*
//
?END
?TERMINATE HASP/CARDS
```

The correct format and sequence of the job stream transmitted in this manner is the user's responsibility.

#### **.PUNCH=nn**

The purpose of this local command is to change the processing of IBM object-code files from the manner in which they are currently being processed. Use of the .PUNCH=nn command allows the user to change:

- a. Whatever was specified by the PUNCH=nn run-time parameter at Beginning-of-Job, or
- b. Whatever was last specified by the .PUNCH=nn local command during the present HASP session.

For example, if the PUNCH=96 run-time parameter was specified at BOJ and the user wants to read or punch 80-column IBM object-code files, the local command .PUNCH=80 can be entered on the console printer to dynamically change the original parameter.

The entry of .PUNCH=80 can be used to dynamically terminate the processing of 96-column IBM object-code files. No translation occurs for card decks that are punched or read following the entry of .PUNCH=80.

If the .PUNCH=96 local command is entered during a HASP session for which the PUNCH=96 run-time parameter was not specified, the command is not executed and an error message is displayed.

#### NOTE

Responsibility for the proper use of the .PUNCH=nn local command rests with the B 1800/B 1700 operator, who must be certain that the proper option is set for each job stream.

#### **.San *<filename>***

The .S local command causes the HASP program to read a file, compress the data, and transmit the file to the central computer.

The letter a in the .S command specifies the device type of the device the HASP program is to read from. The valid one-character mnemonics and their equivalent device types follow:

- D = The file is located on a DISK device.
- R = The file is located on a CARD device.
- T = The file is located on a TAPE device.

The letter n in the .S command specifies the reader number of the reader to read from, and must have a value of 1 to 4 inclusive, if used.

The *<filename>* specified in the .S command is the label of the file. The HASP program uses the file name to locate the file on the device type specified.

The maximum number of readers that can be active at one time is determined by the run-time parameter MAX.RDR. Each reader has a separate file; the internal file name is INPUT0, INPUT1, INPUT2, or INPUT3. When the START reader command is entered to the HASP program, a search is done to find a non-active reader.

If a non-active reader is found, the file is marked active, and the file is opened using the file name and device type specified. If the file is not found, the HASP program displays a message and the operator must re-enter the command.

If all readers are active and the reader number was not specified in the .S command, the HASP program queues the command and notifies the operator that the command is queued. When a reader becomes non-active, the queued command is read by the HASP program and the operator is notified that the command is in process.

If the reader specified in the .S command is active, the HASP program notifies the operator that the reader is active and the .S command is not queued. The operator must re-enter the command when the reader is non-active.

Multiple job steps can also be built and transmitted to the central computer as a single job that is executed serially by the central computer. The data to be sent to the central computer can be on disk, tape, or cards.

In order to create multiple job steps to be executed serially, the local commands .SD, .SR, and .ST can be included as a part of the input file. The .SD, .SR, or .ST commands start in position one of the card image. Position four must be blank. The file name of the data file starts in position five. Any number of .SD, .SR, and .ST local commands can be included as a part of the input file.

All card images that do not have .SD, .SR, or .ST in the first three character positions and a blank in position four are transmitted to the central computer.

When either .SD, .SR, or .ST is detected, the B 1800/B 1700 HASP program stops reading from the primary input file and a secondary file is modified to the device type and file name specified. This secondary file is opened and transmitted to the central computer. When the secondary file has reached the end-of-file, that file is closed and input is read from the primary file. If the next card image contains a local command, this same procedure is performed; otherwise, the card is transmitted to the central computer. If the secondary file contains .SD, .SR, or .ST local commands, they are treated as data and transmitted to the central computer.

The end-of-stream control message is sent to the central computer when the primary input file has reached end-of-file, or when explicitly commanded by the B 1800/B 1700 system operator. Refer to the .EOFn local command within this section for additional information.

If the secondary file is not present, the B 1800/B 1700 HASP program waits for an operator response to continue reading from the input file or to terminate the job stream and send the end-of-stream message to the central computer. When the secondary file is not present, the system operator has three options:

- a. Correct the card and place it in the input hopper of the card reader.
- b. Bypass the record and continue. The system operator must enter the .GOn command for this option.
- c. Close the input file and send the end-of-stream message to the central computer. The system operator must enter the .EOFn command for this option.

Refer to the .GOn and .EOFn local commands within this section for additional information.

The correct format and sequence of the job stream transmitted in the manner just described is the responsibility of the user.

The following example describes the use of the .S command.

Example:

The B 1800/B 1700 system operator can enter the following command to merge secondary files:

```
<job-number> AX.SR CARDS
```

The HASP program opens a card file labeled CARDS which contains the following information:

```
?DATA CARDS
'SD MICR/STARTJCL
.SD MICR/FILE1
.SD MICR/FILE2
.ST MICR/FILE3
.SD MICR/ENDJCL
?END CARDS
```

The file labeled MICR/STARTJCL is transmitted to the central computer, and the files labeled MICR/FILE1 and MICR/FILE2 are transmitted to the central system from a disk device. The file labeled MICR/FILE3 is then transmitted from a tape device. Next, the file labeled MICR/ENDJCL is transmitted from a disk device. When the HASP program labeled CARDS has detected the EOF on the file labeled CARDS, the end-of-stream indicator is transmitted to the central system, and the file labeled CARDS is closed.

The following local command can be used when the B 1800/B 1700 HASP program is executing in the spool mode:

**.NEW-SPOOL**

This command causes the current output spool to be closed as soon as all current output streams are received from the central computer, and a new output spool is opened with a new name. Any new output streams are suspended until the new output spool is opened.

**.SD1 *<filename>***

When the B 1800/B 1700 HASP program is executing in the Spool Mode, .SD1 *<filename>* causes the program to read a disk file labeled HASP/CARDS (or an alternate file name specified for the file), and transmit the file to the central computer.

The numeral 1 in the .SD command specifies the reader number to be read from, and is the only valid value. Only one reader may be open at a time.

The external file name can be overridden by specifying an alternate file name; otherwise, the external file name reverts back to the default file name.

**.ST1 *<filename>***

When the B 1800/B 1700 HASP program is executing in the Spool Mode, .ST1 *<filename>* causes the program to read a tape file labeled HASP/CARDS (or an alternate file name specified for the file), and transmits the file to the central computer.

The numeral 1 in the .ST command specifies the reader number to read from and is the only valid value. Only one reader may be open at a time.

The external file name can be overridden by specifying an alternate file name; otherwise, the external file name reverts back to the default file name.

#### *.RO <option>*

The .RO local command allows the B 1800/B 1700 HASP operator to reset the NEWSPOOL option of the B 1800/B 1700 HASP program. Once this option is reset, all streams received from the central computer are written to a single disk file. The only time a new disk file is opened is when all allocated disk space is exhausted and a new file is required in order to continue processing. The syntax for the .RO command can be any of the following:

```
.RO NS
:RO NEW.SPOOL
.RO NEWSPOOL
.RO NEW/SPOOL
```

#### *.SO <option>*

The .SO input command allows the B 1800/B 1700 HASP operator to set the NEWSPOOL option of the B 1800/B 1700 HASP program. Once this option is set, it remains in effect until reset by the .RO input command. The syntax for the .SO command can be any of the following:

```
.SO NS
.SO NEW.SPOOL
.SO NEWSPOOL
.SO NEW/SPOOL
```

Entry of the .SO command causes the B 1800/B 1700 HASP program to automatically:

- a. Scan for the End-of-File indicator for each stream being received from the central computer.
- b. Close the current spool file if the End-of-File was found and there are no other active streams.
- c. Change the external file name and reopen the spool file that was just closed.

All subsequent console printer messages, print streams, and punch streams are written to the newly-opened spool file.

Console printer messages are not considered a stream and do not invoke this mechanism.

#### *.WHAT*

The .WHAT local command allows the HASP user to inquire about the status of all readers. The typical response is formatted as follows:

```
READER 1 HOT READER
READER 2 NOT ACTIVE
READER 3 CARDS ON RECORD 3425.
READER 4 NOT ACTIVE
```



## Console Printer Output Messages

### LOCAL OUTPUT MESSAGES

The console printer messages that can occur while executing the B 1800/B 1700 HASP program are listed in alphabetical order and explained as follows:

#### ?ME <nnnnnnnn> REQUIRED

Insufficient dynamic memory area is allocated for the buffers. The HASP program terminates. <nnnnnnnn> is the amount of dynamic memory required.

#### “PUNCH=96” WAS NOT A RUN TIME PARAMETER; REQUEST IGNORED

This message indicates that the .PUNCH=96 local command was entered on the console printer, but the PUNCH=96 run-time parameter was not specified at initialization.

#### ADAPTER IS ‘LEASED’ . . . YOU SPECIFIED SWITCHED

The B 1800/B 1700 HASP program was executed with run-time parameters specifying a switched line, but the adapter is strapped for leased line operation. The B 1800/B 1700 HASP program terminates. The adapter can be removed to change the strapping option, a CLEAR/START operation performed, or the run-time parameters may be changed to allow a HASP session using the leased adapter. In both cases, the B 1800/B 1700 HASP program must be re-executed.

#### ADAPTER IS ‘SWITCHED’ . . . YOU SPECIFIED LEASED

The B 1800/B 1700 HASP program was executed with run-time parameters specifying a leased line, but the adapter is strapped for a switched line. The B 1800/B 1700 HASP program terminates. The adapter can be removed to change the strapping option and a CLEAR/START operation performed, or the run-time parameters can be changed to allow a HASP session using the switched adapter. In both cases, the B 1800/B 1700 HASP program must be re-executed.

#### AUDIT FILE CLOSED AT HH:MM:SS.T ON MM/DD/YY

This message indicates that all Data Comm auditing has been stopped and the audit file is closed.

#### AUDIT FILE OPENED AT HH:MM:SS.T ON MM/DD/YY

This message indicates that auditing of all Data Comm activity has begun.

**BCB CONTROL FIELD IN ERROR. VALUE RECEIVED = @n@**

This message indicates that an invalid BCB (Block Control Byte) has been received from the central computer. n is the hexadecimal value of bits 1–3 of the BCB.

**BCB RECEIVED INDICATES LOST TEXT**

This message indicates that the value of the BCB (Block Control Byte) received is either equal to or greater by 2 than the value of the BCB expected. A lost text message is transmitted to the central computer, and it is the central computer's responsibility to correct the BCB.

**CAN'T SEND INPUT SIGNING OFF**

This message indicates that an attempt was made to start an input reader while the B 1800/B 1700 HASP program was signing off.

**CENTRAL SYSTEM HAS YET TO GIVE PERMISSION TO SEND SIGNOFF CARD**

This message indicates that either the local command .CLOSE, or a signoff card has been read through the card device and streams are active.

**COMMUNICATIONS ESTABLISHED**

This message indicates an ACK0 message has been received from the central computer in response to an SOHENQ message sent from the B 1800/B 1700 HASP program.

**ENCOUNTERED EOF BEFORE THE SECOND OF A PAIR OF OBJECT CARDS WAS FOUND THE 96 COL DECK JUST TRANSMITTED WAS NOT COMPLETE**

This message indicates that the End-of-File was detected before both cards of the last pair of cards in the object code file were found.

**END OF OUTPUT SPOOL**

**HASP/PRINT nn FULL – RESUMED AS HASP/PRINTnn**

This message indicates that all allocated disk space for an output spool file has been used. The old file is closed and the nn portion of the external label is incremented by one and opened.

**ERRONEOUS RCB RECEIVED. RCB = @nn@**

This message indicates that the B 1800/B 1700 HASP program has detected an invalid RCB (Record Control Byte). @nn@ is the hexadecimal value of the RCB received.

**ERROR: CONTROL NOT PRESENT**

This message indicates that neither a Single Line Control nor Multi-Line Control was found at the port, channel, and adapter number specified in the HASP/SAVE.PARMS file.

**ERROR: INVALID ADAPTER TYPE**

This message indicates that the adapter identification at the port, channel, and adapter number specified in the HASP/SAVE.PARMS file was not a BISYNC adapter.

**ERROR: INVALID RCB DETECTED, BLOCK SKIPPED**

This message indicates that the B 1800/B 1700 HASP program has detected an invalid RCB (Record Control Byte) sent from the central computer.

#### ERROR: INVALID SCB LENGTH ENCOUNTERED

This message indicates that the length of the String Control Byte (SCB) is greater than the output record size. Only the portion of the record that is equal to the record size is processed.

#### ESTABLISHMENT RETRIES UP. n

This message indicates that ten consecutive attempts were made to send SOHENQ to the central computer without receiving a response from the central computer. n is the number of times this message has been displayed.

#### FILE "DC/AUDIT.FILE" LOCKED

This message indicates that the file labeled DC/AUDIT.FILE is in use. The audit function is turned off.

#### FILE LABELED '<filename>' MISSING. TRY AGAIN.

This message indicates that a reader was started but the file was not present.

#### FILE <filename> TOO LONG OR MISSING SLASH BETWEEN NAMES

This message indicates that the alternate file name supplied has an incorrect format. The local command must be re-entered.

#### HASP/PRINTnn CLOSED – HASP/PRINTnn OPENED

This message indicates that a new output spool file is created in response to the local command .NEW-SPOOL.

#### HASP/SAVE.PARMS NOT ON DISK; EX HASP/MODIFIER

This message indicates that the disk file containing the run-time parameters is not present on disk. The program named HASP/MODIFIER must be executed to build this disk file.

#### INVALID INPUT STREAM ID

This message indicates that the reader number specified is greater than the number of readers specified in the HASP/SAVE.PARMS file.

#### INVALID PUNCH PARAMETER

This message indicates that the local command .PUNCH=nn is in error, where nn must be either 80 or 96.

#### INVALID SPO MESSAGE

This message indicates that a console printer message was entered and its first character was a period (.), but the entire message does not match any of the local commands.

#### LOSS OF CLEAR TO SEND

This message indicates that the Clear to Send signal supplied by the Data Comm hardware has been lost.

#### LOSS OF DATA SET READY

This message indicates that the Data Set Ready signal supplied by the Data Comm hardware has been lost.

#### MEMORY PARITY ERROR

This message indicates that a memory parity error has occurred during the last Data Comm write or read.

#### ONLY '.SD' OR '.ST' ALLOWED IN SPOOL MODE

This message indicates that the .SR local command was used to open a reader when HASP is in the Spool Mode.

#### PERMISSION-TO-SEND RECEIVED FOR NON-EXISTING READER (n)

This message indicates that the B 1800/B 1700 HASP program has been given permission to send a data file without having a hardware device to read from.

#### PERMISSION-TO-SEND RECEIVED FOR UNSOLICITED DEVICE.TYPE=@n@ STREAM=n

This message indicates that the B 1800/B 1700 HASP program has received permission to send a data file it did not request permission to send. TYPE is the low order four bits of the RCB received and STREAM is the reader number of the requested file.

#### PLEASE ENTER "\$Sn

This message indicates that a change of forms message has been received from the central computer. n is the stream number of the response.

#### PUNCH FILE OPENED; REQUEST IGNORED

This message indicates that the local command .PUNCH was entered when a punch file was still open.

#### READER 1 CLOSED

This message indicates that the end of a spool file has been detected and another spool file may be sent.

#### READER n CLOSED

This message indicates that the End-of-File has been detected and another file may be sent using the same reader number. n is the reader number closed.

#### READER n OPENED

This message indicates that the specified reader has been opened and is being transmitted to the central computer. n is the reader number opened.

#### READER n STILL ACTIVE

This message indicates that an attempt was made to open a reader that is still active. n is the reader number of the active stream.

#### READER n WAITING PERMISSION TO SEND FROM HOST

This message indicates that the B 1800/B 1700 HASP program has sent to the central computer a request to send a file, but the central computer has not responded with permission to send. n is the reader number of the reader waiting permission to send.

#### REQUEST-TO-SEND FOR NON-EXISTENT PRINT STREAM (n)

This message indicates that the central computer has requested to send a print stream whose stream number is greater than the number of printers specified by MAX.PRT in the run-time parameters. (n) is the stream number requested.

#### REQUEST-TO-SEND FOR NON-EXISTENT PUNCH STREAM (n)

This message indicates that the central computer has requested to send a punch stream whose stream number is greater than the number of punches specified by MAX.PCH in the run-time parameters. (n) is the stream number requested.

#### RETRIES UP. n

This message indicates that there have been ten unsuccessful attempts to send a message to the central computer. n is the number of times this message has been displayed.

#### RETRY/LIMIT EXCEEDED . . . RE-ESTABLISHING AGAIN

This message indicates that there have been 50 attempts to send a message to the central computer without success. The Data Comm connection is broken. An attempt is made to re-establish the Data Comm line again. This message suggests a hardware problem.

#### SIGNING-OFF

This message indicates that a request has been given to the B 1800/B 1700 HASP program to sign off twice. This message appears on the second request to sign off.

#### TRY BUFFER SIZE nnnn BYTES

This message indicates that the buffer size in the HASP/SAVE.PARMS is not equal to the buffer size used to create the spool file, where nnnn is the buffer size used to create the spool file.

#### SPOOL WAS EMPTY

This message indicates that the spool file that was requested to be sent to the central computer has no data. No action is taken and the file is purged.

#### SWITCH.MODE - - NOW DIAL HOST

This message indicates that the B 1800/B 1700 HASP program is waiting for the B 1800/B 1700 operator to dial the central computer.

#### TOO MANY SLASHES IN FILE NAME

This message indicates that the alternate file name given with the local commands, .SD, .SR, or .ST, has more than two slashes.

#### 96 COL CARD SEQUENCE NOT IN ORDER EXPECTED CARD NO. n BUT FOUND CARD NO. n

This message indicates that a sequence check error was detected as a result of one of a pair of object code file cards being out of sequence or missing.

#### n, IN SPOOL MODE IS AN INVALID INPUT STREAM ID

This message indicates that a reader number other than one was used to open a spool file, where n is the reader number used.

#### IBM/HASP MESSAGES

All operator messages received from the central computer are displayed on the B 1800/B 1700 console printer. These messages are always preceded by a currency (dollar) sign.

#### **Operating Exceptions**

The following differences exist between the B 1800/B 1700 and IBM 360/30 HASP systems.

#### CHANNEL SKIPS ON LINE PRINTER

Not all Burroughs line printers are capable of handling a "skip to channel 12" request. If such a skip is to be attempted, a check is needed to verify whether such a skip is possible on the B 1800/B 1700 system being used.

## PRINTER/PUNCH BACKUP CAPABILITY

Whenever a line printer or card punch is not available to the B 1800/B 1700 HASP program when the print or card punch file is open, the following message is displayed on the console printer:

PRINTER OR BACKUP DISK REQUIRED FOR HASP/OUTPUT

Then if the PBD option of the MCP is set, or if the output is directed to disk by the MCP input command OU, all subsequent output data goes to a disk backup file.

### Change Forms Processing

During execution of the B 1800/B 1700 HASP program, a special forms request message can be received from the central system. The message is then displayed on the console printer to instruct the B 1800/B 1700 operator to load the proper cards in the card punch or paper forms in the line printer. The mode in which the B 1800/B 1700 HASP program is operating determines the action to be taken.

### DIRECT MODE

When the B 1800/B 1700 HASP program is in Direct Mode and a special forms request message is received, one of the following messages is displayed on the console printer:

JOB XXX LOAD XXX FORMS IN RM  $\langle n \rangle$  .PR $n$

JOB XXX LOAD XXX FORMS IN RM  $\langle n \rangle$  .PU $n$

$n$  is the stream number that requires the special forms.  $\langle n \rangle$  is the identification number of the remote system.

The central computer waits for the B 1800/B 1700 operator to respond by entering one of the two following AX input messages after the forms are changed:

$\langle \text{mix-index} \rangle$  AX\$SRM  $\langle n \rangle$  .PR $n$  (For line printer forms change)

$\langle \text{mix-index} \rangle$  AX\$SRM  $\langle n \rangle$  .PU $n$  (For card punch forms change)

$\langle \text{mix-index} \rangle$  is that of the B 1800/B 1700 HASP program.

$n$  is the form number that requires the special forms.

$\langle n \rangle$  is the identification number of the remote system.

### SPOOL MODE

When the B 1800/B 1700 HASP program is in Spool Mode and a special forms request message is received, the appropriate message is written on the output spool, displayed on the console printer, and an appropriate response to the central system is generated automatically by the B 1800/B 1700 HASP program. When the output spool is processed by the HASP/SPOOL program, the same message is displayed on the console printer to allow the operator to load the proper forms in the output device.

## Processing 96-Column IBM Object-Code Files

96-column IBM object-code files can be punched or read when the B 1800/B 1700 HASP program is in Direct Mode. In order to punch or read IBM object-code files with 96-column card devices at the B 1800/B 1700 remote system, the PUNCH=96 run-time parameter must be specified. The PUNCH=96 run-time parameter causes two translation tables to be brought into dynamic memory. The translation tables are used for translating object-code files only.

The dynamic memory size must be increased by 4096 bits to provide space for the translation tables. The dynamic memory size is set by the HASP/MODIFIER program to include the tables if .PUNCH=96 is specified in the run-time parameters.

For detailed information about the .PUNCH=nn local command, refer to the subsection of this manual titled Console Keyboard Input Commands.

For detailed information about the PUNCH=nn run-time parameter, refer to Section 3 titled HASP/MODIFIER PROGRAM.

## PUNCHING 96-COLUMN IBM OBJECT-CODE FILES

When the PUNCH=96 run-time parameter has been specified, the B 1800/B 1700 HASP program tests column 1 of each card record received from the central system. If card column 1 contains a hexadecimal @02@, that card record is considered an IBM object-code file record.

For card records that prove to be IBM object-code card records, the following operations are performed:

- a. Card columns 1 through 40 of the card record are translated into 96 columns of data.
- b. Row 1 of column 81 is punched.
- c. An ASCII-graphic translation is performed on the 96 columns of data, and those 96 columns are punched.
- d. Card columns 41 through 80 of the card record are translated into 96 columns of data.
- e. Row 2 of column 81 is punched.
- f. An ASCII-graphic translation is performed on the 96 columns of data, and those 96 columns are punched.

The performance of the operations described above result in punching two 96-column cards from each 80-column card record received from the central system. The 96-column cards are put in sequence by pairs as indicated by the row 1 or row 2 punch in column 81. Starting with the first card punched, row 1 of column 81 is punched. The second card is punched in row 2 of column 81. Rows 1 and 2 of column 81 are then punched alternately so that column 81 can be tested for sequence by pairs when the cards are read.

## READING 96-COLUMN IBM OBJECT-CODE FILES

When the PUNCH=96 run-time parameter has been specified, the B 1800/B 1700 HASP program tests column 81 of every 96-column card read. Column 81 is tested to determine if the cards are IBM object-code records and if the cards are in sequence by pairs.



Files of 96-column cards are processed as follows:

- a. The first card in the card deck is read and column 81 is tested for a punch in row 1.
  1. If row 1 is punched, a translation is performed to translate the 96 columns of data from the card to 40 columns of data for columns 1 through 40 of an 80-column card-image record.
  2. If row 1 is not punched, the card deck is out of sequence and processing is suspended. The following error message is then displayed on the console printer:

96 COL CARD SEQUENCE NOT IN ORDER  
EXPECTED CARD NO. 1 BUT FOUND CARD NO. 2

- b. The second card is read and column 81 is tested for a punch in row 2.
  1. If row 2 is punched, a translation is performed to translate the 96 columns of data from the card to 40 columns of data for columns 41 through 80 of an 80-column card image record. At this point, two 96-column cards have been read and translated to produce one 80-column card image record.
  2. If row 2 is not punched, the card deck is out of sequence and processing is suspended. The following error message is then displayed on the console printer:

96 COL CARD SEQUENCE NOT IN ORDER  
EXPECTED CARD NO. 2 BUT FOUND CARD NO. 1

- c. The third card and all subsequent odd-numbered cards in the card deck are read and processed in the same manner as the first card.
- d. The fourth card and all subsequent even-numbered cards in the card deck are read and processed in the same manner as the second card.

## HASP FILE INFORMATION

The attributes of the files used with the B 1800/B 1700 HASP program are listed under the subheadings that follow. Note that some of the files are used in both Direct and Spool Mode.

### File Number 0 – Direct Mode

The following are the file attributes for file number 0.

INTERNAL NAME:	INPUT0
EXTERNAL NAME:	HASP/INPUT0
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	96 bytes (defaults to input file attributes)
NUMBER OF BUFFERS:	3
FILE OPENED:	INPUT

The B 1800/B 1700 HASP operator can change the device type of INPUT0 by using the .S local command.

### **File Number 1 – Direct Mode**

The following are the file attributes for file number 1.

INTERNAL NAME:	INPUT1
EXTERNAL NAME:	HASP/INPUT1
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	96 bytes (defaults to input file attributes)
NUMBER OF BUFFERS:	3
FILE OPENED:	INPUT

The B 1800/B 1700 HASP operator can change the device type of INPUT1 by using the .S local command.

### **File Number 2 – Direct Mode**

The following are the file attributes for file number 2.

INTERNAL NAME:	INPUT2
EXTERNAL NAME:	HASP/INPUT2
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	96 bytes (defaults to input file attributes)
NUMBER OF BUFFERS:	3
FILE OPENED:	INPUT

The B 1800/B 1700 HASP operator can change the device type of INPUT2 by using the .S local command.

### **File Number 3 – Direct Mode**

The following are the file attributes for file number 3.

INTERNAL NAME:	INPUT3
EXTERNAL NAME:	HASP/INPUT3
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	96 bytes (defaults to input file attributes)
NUMBER OF BUFFERS:	3
FILE OPENED:	INPUT

The B 1800/B 1700 HASP operator can change the device type of INPUT3 by using the .S local command.

### **File Number 4 – Direct Mode**

The following are the file attributes for file number 4.

INTERNAL NAME:	OUTPUT0
EXTERNAL NAME:	HASP/OUTPUT0
HARDWARE TYPE:	PRINTER or BACKUP DISK

RECORD SIZE: 132  
NUMBER OF BUFFERS: 10  
FILE OPENED: OUTPUT

#### **File Number 5 – Direct Mode**

The following are the file attributes for file number 5.

INTERNAL NAME: OUTPUT1  
EXTERNAL NAME: HASP/OUTPUT1  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 10  
FILE OPENED: OUTPUT

#### **File Number 6 – Direct Mode**

The following are the file attributes for file number 6.

INTERNAL NAME: OUTPUT2  
EXTERNAL NAME: HASP/OUTPUT2  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 10  
FILE OPENED: OUTPUT

#### **File Number 7 – Direct Mode**

The following are the file attributes for file number 7.

INTERNAL NAME: OUTPUT3  
EXTERNAL NAME: HASP/OUTPUT3  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 10  
FILE OPENED: OUTPUT

#### **File Number 8 – Direct Mode**

The following are the file attributes for file number 8.

INTERNAL NAME: OUTPUT4  
EXTERNAL NAME: HASP/OUTPUT4  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 10  
FILE OPENED: OUTPUT

### **File Number 9 – Direct Mode**

The following are the file attributes for file number 9.

INTERNAL NAME:	OUTPUT5
EXTERNAL NAME:	HASP/OUTPUT5
HARDWARE TYPE:	PRINTER or BACKUP DISK
RECORD SIZE:	132
NUMBER OF BUFFERS:	10
FILE OPENED:	OUTPUT

### **File Number 10 – Direct Mode**

The following are the file attributes for file number 10.

INTERNAL NAME:	OUTPUT6
EXTERNAL NAME:	HASP/OUTPUT6
HARDWARE TYPE:	PRINTER or BACKUP DISK
RECORD SIZE:	132
NUMBER OF BUFFERS:	10
FILE OPENED:	OUTPUT

### **File Number 11 – Direct Mode**

The following are the file attributes for file number 11.

INTERNAL NAME:	OUTPUT7
EXTERNAL NAME:	HASP/OUTPUT7
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	10
FILE OPENED:	OUTPUT

### **File Number 12 – Direct Mode**

The following are the file attributes for file number 12.

INTERNAL NAME:	SF0
EXTERNAL NAME:	SF0
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	Defaults to input file attributes
NUMBER OF BUFFERS:	2
FILE OPENED:	INPUT

### **File Number 13 – Direct Mode**

The following are the file attributes for file number 13.

INTERNAL NAME:	SF1
EXTERNAL NAME:	SF1
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	Defaults to input file attributes
NUMBER OF BUFFERS:	2
FILE OPENED:	INPUT

#### **File Number 14 – Direct Mode**

The following are the file attributes for file number 14.

INTERNAL NAME:	SF2
EXTERNAL NAME:	SF2
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	Defaults to input file attributes
NUMBER OF BUFFERS:	2
FILE OPENED:	INPUT

#### **File Number 15 – Direct Mode**

The following are the file attributes for file number 15.

INTERNAL NAME:	SF3
EXTERNAL NAME:	SF3
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	Defaults to input file attributes
NUMBER OF BUFFERS:	2
FILE OPENED:	INPUT

#### **File Number 16 – Direct Mode**

The following are the file attributes for file number 16.

INTERNAL NAME:	REMOTE
EXTERNAL NAME:	REMOTE
HARDWARE TYPE:	REMOTE
RECORD SIZE:	100
NUMBER OF BUFFERS:	1
FILE OPENED:	INPUT/OUTPUT
NUMBER OF STATIONS:	50
MAXIMUM MESSAGES:	255

#### **File Number 17 – Direct Mode**

The following are the file attributes for file number 17.

INTERNAL NAME:	HASPQ0
EXTERNAL NAME:	HASP/QUEUE0
HARDWARE TYPE:	DISK
RECORD SIZE:	400
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	240
AREAS:	8
NUMBER OF BUFFERS:	1
FILE OPENED:	INPUT/OUTPUT/NEW

### File Number 18 – Direct Mode

The following are the file attributes for file number 18.

INTERNAL NAME:	SPOQ
EXTERNAL NAME:	SPOQ
HARDWARE TYPE:	DISK RANDOM
RECORD SIZE:	135
NUMBER OF BUFFERS:	1
FILE OPENED:	INPUT/OUTPUT

### File Number 19 – Spool Mode

The following are the file attributes for file number 19.

INTERNAL NAME:	HASPIN
EXTERNAL NAME:	HASP/CARDS
HARDWARE TYPE:	DISK
RECORD SIZE:	400
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	100
AREAS:	25
NUMBER OF BUFFERS:	1
FILE OPENED:	INPUT

The B 1800/B 1700 HASP operator can request HASPIN with another external file by the use of the .SD1 *<filename>* or .ST *<filename>* local command.

### File Number 20 – Spool Mode

The following are the file attributes for file number 20.

INTERNAL NAME:	HASPOUT
EXTERNAL NAME:	HASP/PRINT
HARDWARE TYPE:	DISK
RECORD SIZE:	400
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	50
AREAS:	100
NUMBER OF BUFFERS:	1
FILE OPENED:	OUTPUT

The initial HASPOUT output spool file is labeled HASP/PRINT01. Subsequent spools or execution of the .NEW-SPOOL local command causes the numeric part of the external file name to be incremented by 1. For example: HASP/PRINT02 and HASP/PRINT03 follows HASP/PRINT01 sequentially.

### File Number 21 – Direct Mode or Spool Mode

The following are the file attributes for file number 21.

INTERNAL NAME:	AUDIT
EXTERNAL NAME:	DC/AUDIT.FILE

HARDWARE TYPE: DISK RANDOM, LOCK  
RECORD SIZE: 180  
RECORDS PER BLOCK: 1  
BLOCKS PER AREA: 100  
AREAS: 10  
NUMBER OF BUFFERS: 1  
FILE OPENED: INPUT/OUTPUT/NEW

The audit mechanism can be started by entering AUDIT as a run-time parameter, or entering the .AUDIT local command on the console keyboard.

#### **File Number 22 – Direct Mode or Spool Mode**

The following are the file attributes for file number 22.

INTERNAL NAME: HEREP  
EXTERNAL NAME: HEREP/REPORT  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 1  
FILE OPENED: OUTPUT

#### **File Number 23 – Direct Mode or Spool Mode**

The following are the file attributes for file number 23.

INTERNAL NAME: PARAMETERS  
EXTERNAL NAME: HASP/SAVE.PARMS  
HARDWARE TYPE: DISK  
RECORD SIZE: 180 (defaults to input file attributes)  
RECORDS PER BLOCK: 1  
BLOCKS PER AREA: 1  
AREAS: 1  
NUMBER OF BUFFERS: 1  
FILE OPENED: INPUT

### **ASSIGNING RCB'S TO FILES**

When the B 1800/B 1700 HASP program is executing in the Direct Mode and sending card images to the central computer, each record is assigned a Record Control Byte (RCB). This assignment is dependent on which reader is opened as illustrated by table 4-1, RCB Assignment.

When the B 1800/B 1700 HASP program is executing in the Direct Mode and receiving print and/or punch streams from the central computer, each buffer is unpacked to create a fully-expanded record to be written to a peripheral device. Each record has a Record Control Byte that controls the selection of the unit to which the unpacked record is to be written. The selection of the unit by the RCB is fixed for the B 1800/B 1700 HASP program. Table 4-1, RCB Assignment, illustrates what file is used for which RCB.

Table 4-1. RCB Assignment

File Number	Internal Filenames	Reader Number	Card RCB	Printer RCB	Punch RCB
0	INPUT0	1	@93@		
1	INPUT1	2	@A3@		
2	INPUT2	3	@B3@		
3	INPUT3	4	@C3@		
4	OUTPUT0			@94@	
5	OUTPUT1			@A4@	@F5@
6	OUTPUT2			@B4@	@E5@
7	OUTPUT3			@C4@	@D5@
8	OUTPUT4			@D4@	@C5@
9	OUTPUT5			@E4@	@B5@
10	OUTPUT6			@F4@	@A5@
11	OUTPUT7				@95@

**SUMMARY STATISTICS REPORT**

During the termination process of a HASP session, a summary statistics report, HEREP (HASP Environment Recording Error Printout) is printed on the line printer in the following format:

ERROR IDENTIFICATION	CUMULATIVE COUNT FROM LAST HASP SESSION
BLOCK SEQ CK	nnnnnnnn
ERROR REPLY	nnnnnnnn
INVALID RESP	nnnnnnnn
TOTAL LINE ERRORS	nnnnnnnn
TIMEOUTS	nnnnnnnn
CRC	nnnnnnnn
OTHER LINE ERRORS	nnnnnnnn
NAKS RCVD	nnnnnnnn
NAKS SENT	nnnnnnnn
BLOCKS WRITTEN	nnnnnnnn
BLOCKS READ	nnnnnnnn



## 5. HASP/SPOOL PROGRAM

### GENERAL

The HASP/SPOOL program operates off-line from the central computer under MCP II control. The HASP/SPOOL program is unlike the B 1800/B 1700 HASP program in that the HASP/SPOOL program does not communicate with the central computer.

The HASP/SPOOL program performs the following main functions:

- a. Creates input spool disk files.
- b. Reads output spool disk files and processes them.

The process of creating input and output spool disk files is referred to as "spooling." Spooling is advantageous for reducing transmission line costs when the spooled files are sent to and from the central computer, and enables the user at a B 1800/B 1700 remote computer to prepare data prior to transmission to the central computer, and process data received from the central computer after completion of an "on-line" HASP session.

The HASP/SPOOL program is described in this section. The items described are listed below in the order in which they are presented:

- a. Operating modes.
- b. Operating instructions.
- c. Console printer input commands.
- d. Console printer messages.
- e. Operating exceptions
- f. Change forms processing.
- g. File information.

### OPERATING MODES

The two modes of operation for the HASP/SPOOL program are the Input Spool Creation Mode (Input Spool Mode), and the Output Spool Printing and Punching Mode (Output Spool Mode). In either mode of operation the B 1800/B 1700 remote computer is off-line from the central computer in all activities regarding the HASP/SPOOL program.

#### Input Spool Creation Mode

The purpose of the Input Spool Creation Mode of the HASP/SPOOL program is to read 80 column card image files, and store them on disk as an input spool file. The card images are read, the data is compressed,

blocked, and stored on disk. The spool files created are transmitted to the central system by the B 1800/B 1700 HASP program at a later time. The HASP/SPOOL program must use the same run-time parameters when creating the input spool file as the B 1800/B 1700 HASP program uses when transmitting the spool file to the central system in Spool Mode.

Many 80 column card image files can be concatenated to form one input spool file labeled HASP/CARDS. A subsequent execution of HASP/SPOOL in the Input Spool Creation Mode creates another file labeled HASP/CARDS. If more than one input spool disk file is to be transmitted during a HASP session, they must have unique names. Therefore, in order to differentiate between subsequent HASP/CARDS files created by HASP/SPOOL, the conventions for changing disk file names must be followed. For detailed information concerning file name changing, refer to the B 1700 SYSTEM SOFTWARE OPERATIONAL GUIDE, form number 1068731.

#### HASP/SPOOL SYSTEM REQUIREMENTS — INPUT SPOOL MODE

The following hardware and software elements are required to execute the HASP/SPOOL program in the Input Spool Creation Mode:

- a. Master Control Program II (MCP II).
- b. 16K bytes of main memory excluding the MCP II memory requirements.
- c. 80 or 96 column card reader.
- d. Disk storage (cartridge, pack, or head-per-track).
- e. Console printer.

#### REMOTE SYSTEM CONFIGURATION — INPUT SPOOL MODE

Figure 5-1 is a block diagram that shows the major hardware and software elements of a B 1800/B 1700 remote system capable of executing the HASP/SPOOL program in the Input Spool Creation Mode.

#### Output Spool Printing and Punching Mode

The purpose of Output Spool Printing and Punching Mode of the HASP/SPOOL program is to read the output spool disk file named HASP/PRINT01, deblock and decompress the records, and send the records to the line printer or the card punch. If the output spool disk file created by the B 1800/B 1700 HASP program was closed and reopened during the HASP session, the label HASP/PRINT01 was incremented by 1 for each open operation. The sequential output spool disk files (HASP/PRINT02, HASP/PRINT03, for example) are also processed in this mode.

#### HASP/SPOOL SYSTEM REQUIREMENTS — OUTPUT SPOOL MODE

The following hardware and software elements are required to execute the HASP/SPOOL program in the Output Spool Printing and Punching Mode:

- a. Master Control Program II (MCP II).
- b. 16K bytes of main memory, excluding the MCP II memory requirements.
- c. 120 or 132-position line printer.

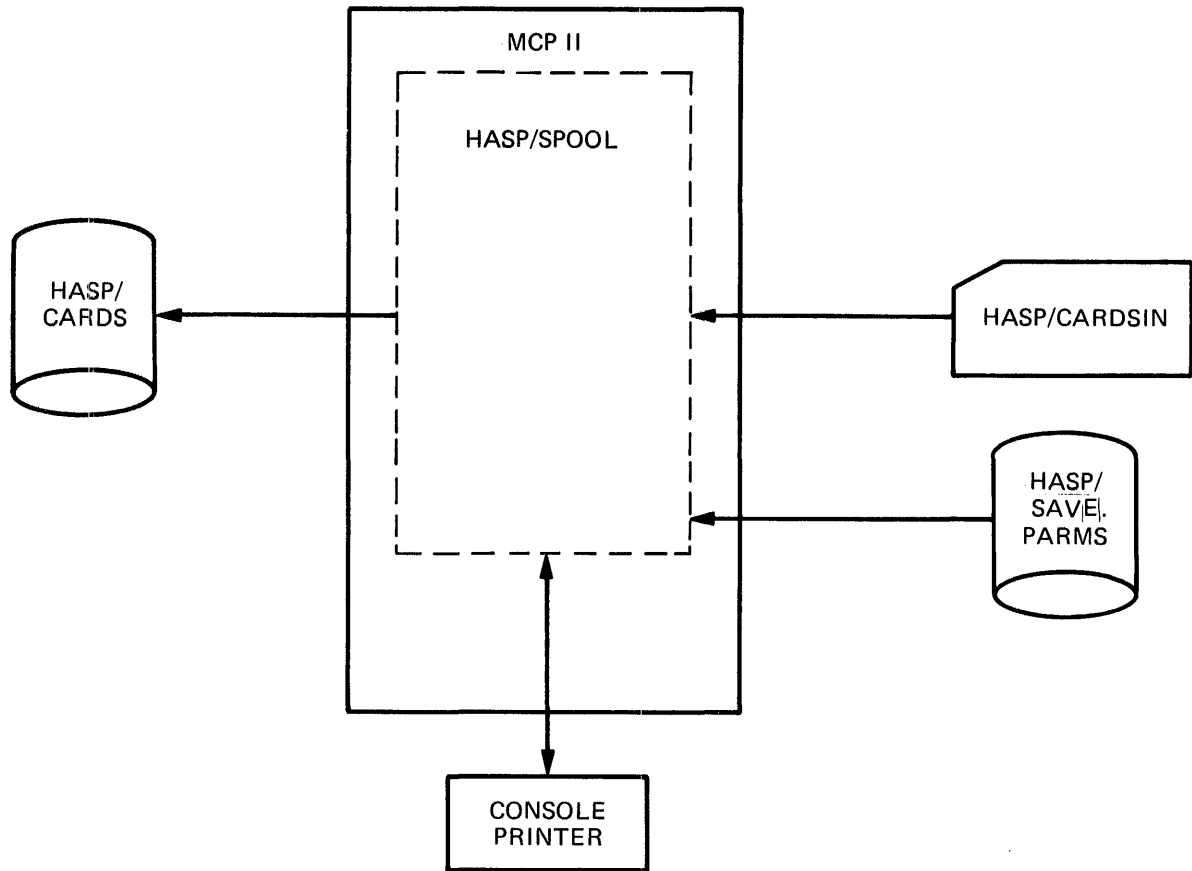


Figure 5-1. Remote System Configuration – HASP/SPOOL Input Spool Mode

- d. Disk storage (cartridge, pack, or head-per-track).
- e. Console printer.
- f. Card punch is optional – 80 or 96 column.

**REMOTE SYSTEM CONFIGURATION – OUTPUT SPOOL MODE**

Figure 5-2 is a block diagram that shows the major hardware and software elements of a B 1800/B 1700 remote system capable of executing the HASP/SPOOL program in the Output Spool Printing and Punching Mode.

**OPERATING INSTRUCTIONS AND INFORMATION**

This subsection provides operating instructions for the HASP/SPOOL program, lists and describes the console printer input commands and output messages, and includes other information related to operating the HASP/SPOOL program.

Examples of the console printer displays following execution of the HASP/SPOOL program in both the Input Spool Mode and Output Spool Mode are included in this section .

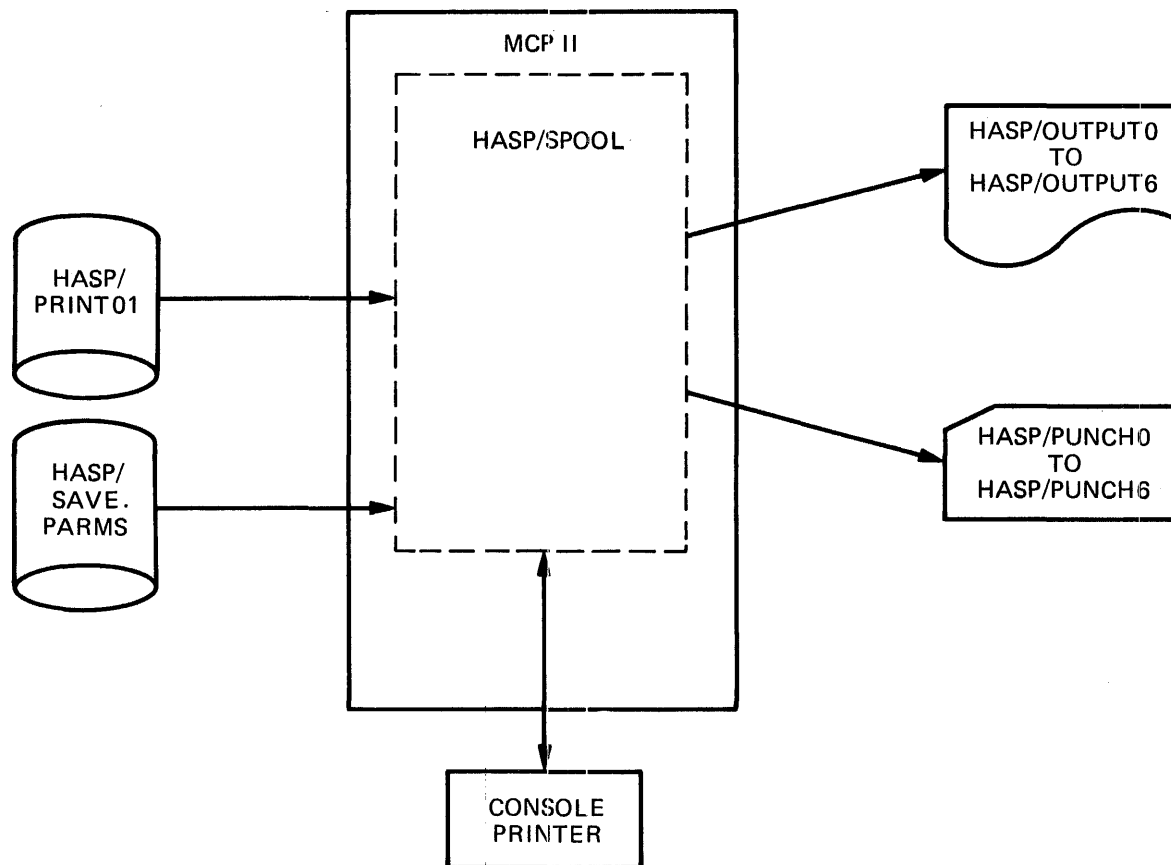


Figure 5-2. Remote System Configuration – HASP/SPOOL Output Spool Mode

### Operating the HASP/SPOOL Program

The HASP/SPOOL program has two modes of operation, the execution of which is controlled by the B 1800/B 1700 operator. At beginning-of-job, entry of the .BOTH local command causes the HASP/SPOOL program to concurrently execute in both Input Spool Mode and Output Spool Mode.

### EXECUTING THE HASP/SPOOL PROGRAM

The first step in executing the HASP/SPOOL program is the entry of an EXECUTE control instruction from either the card reader or console printer.

Example:

```
EXECUTE HASP/SPOOL
```

Following the entry of the EXECUTE statement, a series of messages is displayed on the console printer, as illustrated in the following example.

Example:

```
HASP/SPOOL = 1 BOJ. #=1470 PR = 4 TIME = 13:18:07.9
```

```
% HASP/SPOOL = 1 SPOOLED MODE:
```

```
% HASP/SPOOL = 1 ENTER .SR1 FOR INPUT OR .SP1 FOR OUTPUT OR .BOTH FOR INPUT  
AND OUTPUT TOGETHER
```

```
HASP/SPOOL = 1 ACCEPT.
```

#### ENTRY OF .SR1, .SP1, OR .BOTH LOCAL COMMAND

Following the ACCEPT, the HASP/SPOOL program waits for the operator to enter either the .SR1, .SP1, or .BOTH local commands.

Example:

```
1AX.SR1
```

The entry of .SR1 causes the HASP/SPOOL program to read a card file labeled HASP/CARDSIN and create a spool file on disk labeled HASP/CARDS.

The entry of .SP1 causes the HASP/SPOOL program to read a spool file from disk labeled HASP/PRINT01 and print or punch the records on an output device.

The entry of .BOTH causes the HASP/SPOOL program to read a card file labeled HASP/CARDSIN and create a spool file on disk labeled HASP/CARDS, and concurrently read a spool file from disk labeled HASP/PRINT01 and print or punch the records on an output device.

For a detailed description of the .SR1, .SP1, and .BOTH commands refer to the subsection titled Console Printer Input Commands.

If the local command is incorrect, the following message is displayed on the console printer:

```
INVALID RESPONSE.
```

The HASP/SPOOL program waits for the operator to enter the local command message again.

The operation of the HASP/SPOOL program has been the same up to this point for both the Input Spool Creation Mode and the Output Spool Printing and Punching Mode, with the single exception of the local command entry. The description of the operation of the two modes is presented in the separate subsections that follow.

#### INPUT SPOOL MODE OPERATION

The HASP/SPOOL program is executed in the Input Spool Mode if the operator entered either the .SR1 or .BOTH local command on the console printer.

An input card file labeled HASP/CARDSIN and an output disk file named HASP/CARDS are opened. The input card file can optionally have an alternate file name *<filename>*, in which case the operator must have entered .SR1 *<filename>* on the console printer.

The input card file is read and the data is compressed, blocked, and written to the output disk file. When the end-of-file is detected on the card file, the following message is displayed on the console printer:

END OF FILE, ENTER COMMAND

The HASP/SPOOL program waits for the operator to enter either a .CLOSE or .SR1 local command on the console printer. To continue in the Input Spool Mode, the .SR1 command is entered and causes the HASP/SPOOL program to read another card file. The processed records are added to the output disk file labeled HASP/CARDS.

Examples:

```
<mix-index>AX.SR1
<mix-index>AX.SR1  HASP/TESTONE
```

After the last card file has been read and the end-of-file message displayed on the console printer, the .CLOSE local command must be entered on the console printer, as follows:

```
<mix-index>AX.CLOSE
```

The entry of the .CLOSE local command causes the HASP/SPOOL program to close all open files, and the HASP/SPOOL program to terminate.

The following example illustrates typical console printer commands and messages that occur during the process of initiating and executing the HASP/SPOOL program in the Input Spool Creation Mode.

Example of HASP/SPOOL – Input Spool Mode:

```
EX HASP/SPOOL
  HASP/SPOOL = 1 BOJ.# = 1187 PR=4 TIME = 16:49:13.6
  % HASP/SPOOL = 1 SPOOLED MODE:
  % HASP/SPOOL = 1 ENTER.SR1 FOR INPUT OR .SP1 FOR OUTPUT OR .BOTH FOR INPUT
    AND OUTPUT TOGETHER
  HASP/SPOOL = 1 ACCEPT.
1 AX.SR1
  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.CLOSE
  HASP/SPOOL = 1 EOJ. # = 1187 TIME = 16:50:37.8

CH HASP/CARDS HASP/SPOOLONE
  "HASP/CARDS" CHANGE TO "HASP/SPOOLONE"
```

```

EX HASP/SPOOL
  HASP/SPOOL = 1 BOJ. # =1188 PR = 4 TIME = 16:51:31.3
  % HASP/SPOOL = 1 SPOOLED MODE:
  %HASP/SPOOL = 1 ENTER.SR1 FOR INPUT OR .SP1 FOR OUTPUT OR .BOTH FOR
    INPUT AND OUTPUT TOGETHER
  HASP/SPOOL = 1 ACCEPT.
1 AX.SR1 HASP/TESTONE
  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.SR1 HASP/TESTTWO

  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.CLOSE
  HASP/SPOOL = 1 EOJ. # =1188 TIME = 16:53:59.1

CH HASP/CARDS HASP/SPOOLTWO
  "HASP/CARDS" CHANGED TO "HASP/SPOOLTWO"

```

## OUTPUT SPOOL MODE OPERATION

The HASP/SPOOL program is executed in the Output Spool Mode if the operator entered either the .SP1 or .BOTH local command on the console printer.

The HASP/SPOOL program opens an input disk file specified by the operator, or the default file labeled HASP/PRINT01, if no file was specified. The output printer file is labeled HASP/OUTPUT0 through HASP/OUTPUT6 and the output punch file is labeled HASP/PUNCH0 through HASP/PUNCH6.

The first record of the input disk file contains the run-time parameters used when the file was created by the B 1800/B 1700 HASP program. The HASP/SPOOL program compares its run-time parameters with the parameters from the input disk file, and any differences result in appropriate warning or error messages on the console printer.

The input disk file is read, the data is decompressed, and the records written to the line printer, the card punch, or both.

When the end-of-file is detected on the input disk file, the following messages are displayed on the console printer:

```
SPOOL OUTPUT COMPLETE
```

```
END OF FILE, ENTER COMMAND
```

The HASP/SPOOL program waits for the operator to enter either a .SP1, .RS1, or .CLOSE local command.

After the last input disk file has been read and processed, and the end-of-file message is displayed, the .CLOSE local command must be entered on the console printer, as follows:

```
<mix-index>AX.CLOSE
```

The entry of the .CLOSE local command causes the HASP/SPOOL program to close all open files, and the HASP/SPOOL program to terminate.

If the special forms request message is displayed during execution, the HASP/SPOOL program waits for the operator to load the special forms in the line printer or card punch. Refer to the subsection titled CHANGE FORMS PROCESSING in this section of the manual for additional information about special forms processing.

The following example illustrates typical console printer commands and messages that occur during the process of initiating and executing the HASP/SPOOL program in the Output Spool Mode.

#### Example of HASP/SPOOL – Output Spool Mode:

```
EX HASP/SPOOL
  HASP/SPOOL = 1 BOJ. #=1356 PR = 4 TIME = 16:16:15.3
  % HASP/SPOOL = 1 SPOOLED MODE:
  % HASP/SPOOL = 1 ENTER .SR1 FOR INPUT OR .SP1 FOR OUTPUT OR .BOTH FOR
    INPUT AND OUTPUT TOGETHER
  HASP/SPOOL = 1 ACCEPT.
1 AX.SP1
  % HASP/SPOOL = 1 $*18.36.27 JOB 535 ON RM16.RD1 – STAR
  % HASP/SPOOL = 1 $ 18.37.02 JOB 535 STAR EXECUTING A PRIO 0
  % HASP/SPOOL = 1 $ 18.38.43 JOB 535 STAR ON RM16.PR1 PRIO 9
  % HASP/SPOOL = 1 $*18.39.15 JOB 537 ON RM16.RD1 – STAR
  % HASP/SPOOL = 1 SPOOL OUTPUT COMPLETE
  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.SP1 HASP/PRINT02
  % HASP/SPOOL = 1 $ 18.39.40 JOB 537 STAR ON RM16.RD1 PRIO 0
  % HASP/SPOOL = 1 $*18.39.41 JOB 538 ON RM16.RD1 – STAR
  % HASP/SPOOL = 1 $ 18.41.01 JOB 537 STAR ON RM16.PR1 PRIO 9
  % HASP/SPOOL = 1 SPOOL OUTPUT COMPLETE
  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.SP1 HASP/PRINT03
  % HASP/SPOOL = 1 $ 18.41.57 JOB 536 STAR ON RM16.PR1 PRIO 9
  % HASP/SPOOL = 1 $ 18.43.14 JOB(S) NOT FOUND
  % HASP/SPOOL = 1 SPOOL OUTPUT COMPLETE
  % HASP/SPOOL = 1 END OF FILE, ENTER COMMAND.
  HASP/SPOOL = 1 ACCEPT.
1 AX.CLOSE
  HASP/SPOOL = 1 EOJ. #=1356 TIME = 16:18:54.5
```

#### Console Keyboard Input Commands

Local commands are the only type of console keyboard input command used with the HASP/SPOOL program. Some of the commands can be used only when the HASP/SPOOL program is in the Input Spool Creation Mode or the Output Spool Printing and Punching Mode, while other commands can be used in both modes.

The following local commands can be used when the HASP/SPOOL program is to execute in either mode:

**.BOTH**

This command is the only valid response to the ACCEPT message at the beginning of execution of the HASP/SPOOL program.



The .BOTH command causes a card file labeled HASP/CARDSIN to be read and a spool file on disk labeled HASP/CARDS to be created. Simultaneously with the above operation, a spool file labeled HASP/PRINT01 is read, and either prints the records on the line printer, punches them on the card punch, or displays them on the console printer.

No alternate file name is permitted with the use of the .BOTH command.

#### .CLOSE

The .CLOSE command causes all files currently open to be closed, and terminates the HASP/SPOOL program.

The use of the .CLOSE command is permitted only when the end-of-file message is displayed on the console printer.

#### .SP1<filename>

The .SP1 command causes the HASP/SPOOL program to open an input spool file labeled HASP/PRINT01 or an input spool file with an alternate file name. The HASP/SPOOL program then does a parameter comparison of the first record of HASP/PRINT01 (which contains the run-time parameters with which the spool file was created) and the current run-time parameters.

If the buffer sizes specified for the HASP and HASP/SPOOL files are not the same, the HASP/SPOOL program terminates. The HASP/SPOOL program must be executed with run-time parameters that are compatible with the parameters used for the B 1800/B 1700 HASP program.

If the buffer sizes specified for the HASP and HASP/SPOOL files are the same, the HASP/SPOOL program continues to read the spool file and decompress the data.

The following local command can be used when the B 1800/B 1700 HASP/SPOOL program is to be executed in the Input Spool Creation mode.

#### .SR1<filename>

The .SR1 command causes the HASP/SPOOL program to read and compress the information from a card file labeled HASP/CARDSIN or from a file with an alternate file name and create a disk file labeled HASP/CARDS for transmission to the central computer at a later time.

The following local command can be used when the HASP/SPOOL program is to be executed in the Output Spool Printing and Punching mode.

#### .RS1

The .RS1 command causes the HASP/SPOOL program to continue processing the second and successive HASP/PRINTnn files that were created by the B 1800/B 1700 HASP program when the size of the disk spool file was exceeded. The .RS1 command cannot be used to cause continued processing of successive files created by use of the .NEW-SPOOL command with the B 1800/B 1700 HASP program. For processing such files, the .SR1 <filename> command must be used.

### Console Printer Messages

The console printer messages that can occur while executing the B 1700 HASP/SPOOL program are listed below in alphabetical sequence.

ENCOUNTERED EOF BEFORE THE SECOND OF A PAIR OF OBJECT CARDS FOUND  
THE 96 COL DECK JUST INPUT WAS NOT COMPLETE

The End-of-File was detected before both cards of the last pair of cards in the object code file were found.

END OF FILE, ENTER COMMAND

Either the input card file or the input spool file reached the End-of-File.

ENTER .SR1 FOR INPUT .SP1 FOR OUTPUT OR .BOTH FOR THE INPUT AND  
OUTPUT TOGETHER

This message is a request for a response. Refer to the subsection titled CONSOLE  
PRINTER INPUT COMMANDS for additional information.

ERROR: "BUFFER" PARAMETER DOES NOT AGREE WITH SPOOL

The current size of the HASP/SPOOL buffer does not match the spool buffer size.

ERROR: INVALID RCB DETECTED, BLOCK SKIPPED

The HASP/SPOOL program detected an invalid Record Control Byte (RCB). The data block is skipped.

ERROR: INVALID SCB LENGTH ENCOUNTERED REST OF BLOCK SKIPPED

The length of the String Control Byte (SCB) is greater than the output record size. Only the portion of the record that is equal to the record size is processed.

ERROR: SPOOL FILE EMPTY

The input spool file does not contain any records.

ERROR: INVALID "SCRB", RECORD SKIPPED

This message indicates that the block contains an invalid Sub-Record Control Byte (SRCB).

FILE HASP/PRINTnn NOT FOUND RE-ENTER COMMAND

The input spool file is not present on disk.

FILE NAME *<filename>* TOO LONG OR MISSING SLASH BETWEEN NAMES.

The alternate file name is not formatted correctly.

HASP/SAVE.PARMS NOT ON DISK RUN THE HASP/MODIFIER PROGRAM

This message indicates that the disk file containing the run-time parameter is not present on disk.

## INVALID RESPONSE

An invalid response was given to any of the ACCEPT messages. The HASP/SPOOL program waits for another input message.

## SPOOLED MODE

This message is displayed after the run-time parameters have been read by HASP/SPOOL.

## SPOOL OUTPUT COMPLETE

The End-of-File has been detected on the input spool file.

## TOO MANY SLASHES IN FILE NAME

The alternate file name has more than two slashes.

## 96 COL CARD SEQUENCE NOT IN ORDER EXPECTED CARD NO. *n* BUT FOUND CARD NO. *n*

This message indicates that a sequence check error was detected as a result of one of a pair of object code file cards being out of sequence or missing.

### Change Forms Processing

When the HASP/SPOOL program is in the Output Spool Printing and Punching Mode, and a special forms request message is encountered that was placed in the spool file by the central system, one of the following messages is displayed on the console printer and an ACCEPT is executed.

```
JOB XXX LOAD XXX FORMS IN RM <n> .PR 1  
JOB XXX LOAD XXX FORMS IN RM <n> .PU 1  
<n> is the identification number of the remote system.
```

The HASP/SPOOL program waits for the B 1800/B 1700 operator to load the proper forms in either the card punch or the line printer. The operator must enter one of the following AX input messages on the console printer:

```
<mix-index> AX$SRM <n> .PR 1           (For line printer forms change)  
<mix-index> AX$SRM <n> .PU 1           (For card punch forms change)  
<mix-index> is that of the HASP/SPOOL program  
<n> is the identification number of the remote system.
```

All other console printer messages received for the B 1800/B 1700 HASP program from the central system while the output spool file was being created are displayed.

### Processing 96-Column IBM Object-Code Files

96-column IBM object-code files can be punched or read with the HASP/SPOOL program. In order to punch or read IBM object-code files with 96-column card devices at the B 1800/B 1700 remote system, the PUNCH=96 run-time parameter must be specified. Translation tables are used for translating the object-code files.

For detailed information about the PUNCH=nn run-time parameter, refer to the subsection titled Default Run-Time Parameters.

## PUNCHING 96-COLUMN IBM OBJECT-CODE FILES

When the PUNCH=96 run-time parameter has been specified, the HASP/SPOOL program tests column 1 of each card record read from the spool file. If card column 1 contains a hexadecimal @02@, that card record is considered an IBM object-code file record.

For card records that prove to be IBM object-code card records, the following operations are performed:

- a. Card columns 1 through 40 of the card record are translated into 96 columns of data.
- b. Row 1 of column 81 is punched.
- c. An ASCII-graphic translation is performed on the 96 columns of data, and those 96 columns are punched.
- d. Card columns 41 through 80 of the card record are translated into 96 columns of data.
- e. Row 2 of column 81 is punched.
- f. An ASCII-graphic translation is performed on the 96 columns of data, and those 96 columns are punched.

The performance of the operations previously described result in punching two 96-column cards from each 80-column card record read from the spool file. The 96-column cards are put in sequence by pairs as indicated by the row 1 or row 2 punch in column 81. Starting with the first card punched, row 1 of column 81 is punched. The second card is punched in row 2 of column 81. Rows 1 and 2 of column 81 are then punched alternately so that column 81 can be tested for sequence by pairs when the cards are read.

## READING 96-COLUMN IBM OBJECT-CODE FILES

When the PUNCH=96 run-time parameter has been specified, the HASP/SPOOL program tests column 81 of every 96-column card read. Column 81 is tested to determine if the cards are IBM object-code records and if the cards are in sequence by pairs.

Files of 96-column cards are processed as follows:

- a. The first card in the card deck is read and column 81 is tested for a punch in row 1.
  1. If row 1 is punched, a translation is performed to translate the 96 columns of data from the card to 40 columns of data for columns 1 through 40 of an 80-column card-image record.
  2. If row 1 is not punched, the card deck is out of sequence and processing is suspended. The following error message is then displayed on the console printer:

```
96 COL CARD SEQUENCE NOT IN ORDER  
EXPECTED CARD NO. 1 BUT FOUND CARD NO. 2
```

- b. The second card is read and column 81 is tested for a punch in row 2.
  1. If row 2 is punched, a translation is performed to translate the 96 columns of data from the card to 40 columns of data for columns 41 through 80 of an 80-column card image record. At this point, two 96-column cards have been read and translated to produce one 80-column card image record.
  2. If row 2 is not punched, the card deck is out of sequence and processing is suspended. The following error message is then displayed on the console printer:

96 COL CARD SEQUENCE NOT IN ORDER  
 EXPECTED CARD NO. 2 BUT FOUND CARD NO. 1

- c. The third card and all subsequent odd-numbered cards in the card deck are read and processed in the same manner as the first card.
- d. The fourth card and all subsequent even-numbered cards in the card deck are read and processed in the same manner as the second card.

### HASP/SPOOL FILE INFORMATION

The attributes of the files used with the B 1800/B 1700 HASP/SPOOL program are listed under the sub-headings that follow.

#### File Number 0 – Output Spool Mode

The following are the file attributes for file number 0.

INTERNAL NAME:	HASP.IN
EXTERNAL NAME:	HASP/PRINT01
HARDWARE TYPE:	DISK
RECORD SIZE:	400
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	100
AREAS:	25
NUMBER OF BUFFERS:	1
FILE OPENED:	INPUT

A request can be made for HASP.IN with an external name other than HASP/PRINTnn by use of the local command `.SP1 <filename>`

### File Number 1 – Input Spool Mode

The following are the file attributes for file number 1.

INTERNAL NAME:	HASP.OUT
EXTERNAL NAME:	HASP/CARDS
HARDWARE TYPE:	DISK
RECORD SIZE:	400
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	100
AREAS:	25
NUMBER OF BUFFERS:	1
FILE OPENED:	OUTPUT

### File Number 2 – Input Spool Mode

The following are the file attributes for file number 2.

INTERNAL NAME:	HASPCARDS
EXTERNAL NAME:	HASP/CARDSIN
HARDWARE TYPE:	96-COLUMN DATA RECORDER
RECORD SIZE:	96 bytes (defaults to input file attributes)
NUMBER OF BUFFERS:	3
FILE OPENED:	INPUT

A request can be made for HASPCARDS with an external name other than HASP/CARDSIN by the use of the local command `.SR1 <filename>`.

### File Number 3 – Output Spool Mode

The following are the file attributes for file number 3.

INTERNAL NAME:	PRINT0
EXTERNAL NAME:	HASP/OUTPUT0
HARDWARE TYPE:	PRINTER or BACKUP DISK

RECORD SIZE: 132  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for print streams with a RCB of @94@.

**File Number 4 – Output Spool Mode**

The following are the file attributes for file number 4.

INTERNAL NAME: PRINT1  
EXTERNAL NAME: HASP/OUTPUT1  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE 132  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for print streams with a RCB of @A4@.

**File Number 5 – Output Spool Mode**

The following are the file attributes for file number 5.

INTERNAL NAME: PRINT2  
EXTERNAL NAME: HASP/OUTPUT2  
HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for print streams with a RCB of @B4@.

**File Number 6 – Output Spool Mode**

The following are the file attributes for file number 6.

INTERNAL NAME: PRINT3  
EXTERNAL NAME: HASP/OUTPUT3

HARDWARE TYPE:           PRINTER or BACKUP DISK  
RECORD SIZE:             132  
NUMBER OF BUFFERS:       3  
FILE OPENED:             OUTPUT

This file is used only for print streams with a RCB of @C4@.

**File Number 7 – Output Spool Mode**

The following are the file attributes for file number 7.

INTERNAL NAME:            PRINT4  
EXTERNAL NAME:            HASP/OUTPUT4  
HARDWARE TYPE:            PRINTER or BACKUP DISK  
RECORD SIZE:             132  
NUMBER OF BUFFERS:       3  
FILE OPENED:             OUTPUT

This file is used only for print streams with a RCB of @D4@.

**File Number 8 – Output Spool Mode**

The following are the file attributes for file number 8.

INTERNAL NAME:            PRINT5  
EXTERNAL NAME:            HASP/OUTPUT5  
HARDWARE TYPE:            PRINTER or BACKUP DISK  
RECORD SIZE:             132  
NUMBER OF BUFFERS:       3  
FILE OPENED:             OUTPUT

This file is used only for print streams with a RCB of @E4@.

**File Number 9 – Output Spool Mode**

The following are the file attributes for file number 9.

INTERNAL NAME:            PRINT6  
EXTERNAL NAME:            HASP/PRINT6



HARDWARE TYPE: PRINTER or BACKUP DISK  
RECORD SIZE: 132  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for print streams with a RCB of @F4@.

**File Number 10 – Output Spool Mode**

The following are the file attributes for file number 10.

INTERNAL NAME: PUNCH0  
EXTERNAL NAME: HASP/PUNCH0  
HARDWARE TYPE: 96-COLUMN CARD PUNCH or BACKUP DISK  
RECORD SIZE: 96  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for punch streams with a RCB of @95@.

**File Number 11 – Output Spool Mode**

The following are the file attributes for file number 11.

INTERNAL NAME: PUNCH1  
EXTERNAL NAME: HASP/PUNCH1  
HARDWARE TYPE: 96-COLUMN CARD PUNCH or BACKUP DISK  
RECORD SIZE: 96  
NUMBER OF BUFFERS: 3  
FILE OPENED: OUTPUT

This file is used only for punch streams with a RCB of @A5@.

### **File Number 12 – Output Spool Mode**

The following are the file attributes for file number 12.

INTERNAL NAME:	PUNCH2
EXTERNAL NAME:	HASP/PUNCH2
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	3
FILE OPENED:	OUTPUT

This file is used only for punch streams with a RCB of @B5@.

### **File Number 13 – Output Spool Mode**

The following are the file attributes for file number 13.

INTERNAL NAME:	PUNCH3
EXTERNAL NAME:	HASP/PUNCH3
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	3
FILE OPENED:	OUTPUT

This file is used only for punch streams with a RCB of @C5@.

### **File Number 14 – Output Spool Mode**

The following are the file attributes for file number 14.

INTERNAL NAME:	PUNCH4
EXTERNAL NAME:	HASP/PUNCH4
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	3
FILE OPENED:	OUTPUT

This file is used only for punch streams with a RCB of @D5@.

### **File Number 15 – Output Spool Mode**

The following are the file attributes for file number 15.

INTERNAL NAME:	PUNCH5
EXTERNAL NAME:	HASP/PUNCH5
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	3
FILE OPENED:	OUTPUT

This file is used only for punch streams with a RCB of @E5@.

### **File Number 16 – Output Spool Mode**

The following are the file attributes for file number 16.

INTERNAL NAME:	PUNCH6
EXTERNAL NAME:	HASP/PUNCH6
HARDWARE TYPE:	96-COLUMN CARD PUNCH or BACKUP DISK
RECORD SIZE:	96
NUMBER OF BUFFERS:	3
FILE OPENED:	OUTPUT

This file is used only for punch streams with a RCB of @F5@.

### **File Number 17 – Input Spool Mode or Output Spool Mode**

The following are the file attributes for file number 17.

INTERNAL NAME:	PARAMETERS
EXTERNAL NAME:	HASP/SAVE.PARMS
HARDWARE TYPE:	DISK
RECORD SIZE:	180 (defaults to input file attributes)
RECORDS PER BLOCK:	1
BLOCKS PER AREA:	1

AREAS: 1  
NUMBER OF BUFFERS: 1  
FILE OPENED: INPUT

### **OPERATING EXCEPTIONS**

The following differences exist between the B 1800/B 1700 and IBM 360/30 HASP systems.

#### **Channel Skips on Line Printer**

Not all Burroughs line printers are capable of handling a "skip to channel 12" request. If such a skip is to be attempted, a check is needed to verify that such a skip is possible on the B 1800/B 1700 system being used.

#### **Printer/Punch Backup Capability**

Whenever a line printer or card punch is not available to the HASP/SPOOL program, the following message is displayed on the console printer:

PRINTER OR BACKUP DISK REQUIRED FOR HASP/PRINT

PUNCH OR BACKUP DISK REQUIRED FOR HASP/PUNCH

Then if the PBD option of the MCP is set, or if the output is directed to disk by the MCP input command OU, all subsequent output data goes to a disk backup file.

## APPENDIX A

### GLOSSARY

#### ACK0

ACK0 indicates that the previous block received was accepted without error, and that the receiver is ready to accept the next block of the transmission.

ACK0 is represented by a DLE character followed by a hexadecimal 70.

#### Bandwidth

An expression of Hertz measurement which states the difference between the high and low frequencies in a communication channel.

#### Baud

A unit of signaling speed equal to the number of discrete conditions of signal events per second.

#### Binary Synchronous (BSC)

A method of data transmission which allows sending or receiving of data streams. The data streams can contain bit patterns which would normally be detected as control character sequences. Therefore, data streams containing object program code may be transmitted. The transmission is always synchronous.

#### Bit Rate

The speed, usually expressed in bits per second (BPS), with which bits are transmitted over a communication channel.

#### Block

A group of characters or bits sent as an integral unit. Usually an error-checking procedure is applied over a block, for control and recovery purposes.

### Block Check Character (BCC)

The Block Check Character is a redundant character added to the end of a transmission block for the purpose of error detection.

### Block Control Byte (BCB)

The Block Control Byte consists of eight bits, of which four bits are used to number blocks of data transmitted between the remote and central systems.

### Break

Hangs-up the switched line.

### Channel

The communication path used to transmit signals between two or more points. Often referred to as a link, circuit, line, or path.

### Character

A set of elements arranged in orderly groups to represent digits, symbols, or letters. Represented in two forms: (1) for use by computers, business machines, communications facilities, for example, usually in groups of binary bits; and (2) for use by man in conveying an understandable form of decimal digits, alphabetic characters, punctuation, or other special symbols. Characters may be represented using groups of bits, commonly five, six, seven, or eight bits.

### Circuit

The configuration of equipment used in transmitting data from one location to another. A circuit may involve more than one type of facility.

### Circuit Assurance

The function of verifying the existence or the operational state of the communication channel between stations.

### Common Carrier

A company that provides communication service for public hire.

### Communication Link

The connection of two or more stations by the same communication channel. This link includes the communication control capability of the stations connected in the link.

## Communication System

The combination of all the links, link interface equipment, application and systems software, including control procedures, that are required to effect the transmission of coded information between stations in the system.

## Connection

The established path between two or more terminal installations. A permanent connection is established by leasing facilities from a common carrier.

## Control Character, Data Communications

A functional character intended to control or facilitate transmission of information over communication networks. The major communication control characters provided are ACK, BEL, CAN, CR, DC1, DC2, DC3, DC4, DEL, DLE, ENQ, EOT, ESC, ETX, FIB, FF, FS, FSL, GS, GSL, HT, LF, NAK, NUL, POL, RS, SI, SO, SHO, STX, SYN, US, and VT.

## Control Procedure, Data Communications

The means used to provide for the orderly communication of information between the stations on a data communications link.

## Data Communications System

One or more data links, each of which may be operating in the same or a different mode.

## Data Communications

The transfer of encoded information by means of electrical transmission systems.

## Data Link

This is an ensemble of terminal installations and the interconnecting network, together with controlling procedures operating in a particular mode, that permits information to be exchanged between locations.

## Data Link Escape (DLE)

Data Link Escape is a control character that is used to change the meaning of a limited number of contiguously following characters, which then become supplementary data transmission control functions. Only graphics and transmission control characters can be used in DLE sequences.

The Data Link Escape character combinations are used primarily for binary synchronous operation. The Data Link Escape character is always the first character of the two character combinations. The DLE combination characters are listed below:

ACK0	The ACK0 character is used only for binary synchronous operation. Refer to ACK0 in this Glossary for a detailed description of the ACK0 character.
DLE STX	Used to start transparent mode in BSC operation.
DLE DLE DLE EOT DLE ENQ DLE SYN	Used as control characters during transparent mode of operation. The mnemonic for DLE EOT is DEOT.
DLE ETB DLE ETX DLE ITB	These characters are used to end transparent mode of operation.

#### Data Set

A circuit termination device used to provide an interface between a data communication circuit and a data terminal. A modulation and/or demodulation function is typically performed in a data set.

#### Data Terminal

That part of a station concerned with the functions of generating data and/or recording or displaying of data, together with the control equipment or system software necessary to control these functions.

#### Dedicated Channel

A communication channel provided for the exclusive use of a specific subscriber on a contract basis. (See Private Line.)

#### Delimiters

Control signals used to define the extent of a particular sequence of characters.

#### Demodulation

The process of retrieving an original signal from a modulated carrier wave. This technique is used in data sets to make communication signals compatible with business machine signals (contrast with definition of Modulation).

#### DEOT

The mnemonic for the DLE EOT communication control sequence, used to signal that a disconnect of a switched circuit must be initiated.

#### Dial-Up

The use of a dial or pushbutton telephone to establish a station-to-station telephone contact.



### Disconnect (DEOT)

A Disconnect is a communication control sequence consisting of DLE followed by EOT, used to signal that a disconnect of a switched circuit must be initiated.

### End of Text (ETX)

The end-of-text control character is used to indicate the end of the last block of text.

### End of Transmission (EOT)

The end-of-transmission control character is used to indicate the conclusion of a communication sequence. Receipt of an EOT will set the terminal in a control state, waiting for a polling, selection, or contention sequence. EOT may be transmitted by a master station to abort a transmission sequence. To ensure that terminals are in a control state, EOT precedes a communication control sequence.

### End of Transmission Block (ETB)

The end-of-transmission block control character is optionally used when messages are of sufficient length to warrant their being broken into smaller transmission blocks. ETB indicates the end of a block of data, either in a heading or in the message text. The heading or text is resumed after transmission of a block number and SOH or STX.

### Enquiry (ENQ)

The enquiry control character is used as a request for a response of station status or for a retransmission of control characters.

### Error

Any received character or sequence of characters that does not conform to those transmitted.

### Error Control

A system that detects the presence of errors. In some systems, refinements are added that correct some of the errors, either by operations on the received data or by retransmission of the data from the source.

### Error Recovery Procedure

Data communications control procedures used to restore normal operation to a data link after unusual (abnormal) events have occurred.

### Exchange

A defined area, served by a communications carrier, within which the carrier furnishes service.

## Facility

The type of communication medium used to provide communication circuits (e.g., cable, radio, open-wire).

## Format

The predetermined arrangement of characters, field, lines, page numbers, punctuation marks, etc., used to transfer data from one location to another. Refers to input, output and files.

## Function Control Sequence (FCS)

The function control sequence is 16 bits in length and is used to control the flow of individual message streams.

## Handshaking

This is the exchange of control signals between data sets when the connection is established.

## Header

A sequence of characters that may precede the text of a message to provide the information necessary to route the message to its ultimate destination(s). A message header may also contain communications relating information other than routine instructions.

## Hertz (Hz)

A measure of frequency or bandwidth. The same as cycles per second.

## Houston Automatic Spooling Program (HASP)

A set of programs used for message communications between an IBM central computer system and one or more remote computer systems.

## IBM HASP

The HASP software used with the IBM central computer in a HASP system.

## Identification

A sequence of characters used during the establishment of a connection and subsequently, if needed, to identify a station.

## Identifier

A sequence of one or more characters transmitted by a station in order to identify itself.

### Information Block

A sequence of characters of fixed or variable length which is a subdivision of an information message formed for the purpose of meeting transmission requirements.

### Information Message

A sequence of characters conveying the text. It may also convey supplementary information forming a heading.

### Line Adapter

A line adapter consists of a logic card used to interface a data set or communications line into a Multi-Line or Single-Line I/O Control.

### Line Switching

A switching technique that temporarily connects two lines together so that two stations can directly exchange information.

### Link, Data Communication

The logical association of two or more stations interconnected by the same communication channel. A data link includes the communication control capability of the connected stations.

### Message

A sequence of characters arranged in a form suitable for the purpose of conveying information from an originator to one or more destinations (or addresses). It contains the information to be conveyed (called the text) and may, in addition, contain communication information to aid in the routing of handling of the message (called the header).

### Message Format

Rules for the placement of such portions of a message as message header, address, text, and end of message.

### Message Numbering

The identification of each message within a communications system by the assignment of a sequential number.

### Modem

See definition of Data Set.

## Modulation

The process by which some characteristic of one wave is varied in accordance with another wave. This technique is used in data sets to make business machine signals compatible with communication facilities (contrast with definition of Demodulation).

## Multi-Leaving

A computer-to-computer communication technique developed for use with the HASP system. With multi-leaving, both the central and remote computers in a HASP system have the capability to respond to a message stream with another message stream. Refer to the HASP documents listed in the subsection titled Related Documents in Section 1 of this manual, for additional details about multi-leaving.

## Multi-Line Control (MLC)

The Multi-Line Control is an I/O control that provides the functional control between the computer system and the line adapters. An integral part of a multi-line data communications subsystem.

## NAK

The Negative Acknowledgement control character is used as a negative response to a transmission (indicating character parity failure for any character of a block or, in a message, a failure of the BCC, or block number sequence check failure).

## Network

The ensemble of equipment through which connections are made between terminal installations. This ensemble of equipment operates in real time and does not introduce, store, or forward delays. A switched telephone network is the network of telephone lines normally used for dialed telephone calls. A private network is a network of communication channels reserved for the use of one customer.

## Off Line

Used to describe terminal equipment which is not connected to a transmission line. Can also describe other devices not in direct communication with the central processing unit.

## On Line

Used to describe terminal equipment which is connected to a transmission line. Can also describe other devices in a direct communication with the central processing unit.

## Operating System (OS)

The operating system used with IBM 360/370 Series computers.

### Point-to-Point Connection

A configuration in which a connection is established between two, and only two, terminal installations. The connection may include switching facilities.

### Private Line

A channel or circuit furnished a subscriber for his exclusive use (see definition of Dedicated channel).

### Queue

A linear list for which all insertions are usually made at one end of the list and all deletions and other accesses are made at the other end.

### Record Control Byte (RCB)

The Record Control Byte is used to identify each record type within a transmission block.

### Remote Job Entry (RJE)

A computer communications system in which a central computer executes programs that were sent to it by other computers that appear as terminals to the central computer.

### Signal-Conversion Equipment

That part of the terminal installation belonging to the data channel, comprising at least one modulator or one demodulator. This equipment provides modulation according to the signals to be transmitted, and/or demodulation of the signals received.

### Single Line Control (SLC)

The single line control is an I/O control that provides the functional control between a computer system and a line adapter. The SLC is an integral part of a single line data communications subsystem.

### Start of Header (SOH)

The start-of-header control character is required only when a message header is to be sent with a transmission. When used, SOH is the first of a sequence of characters used for the message header.

### Start of Text (STX)

The start-of-text control character precedes a sequence of characters which form the text of the transmission.

## Station

The aggregate of the terminal equipment (and system software) and communication control equipment (and system software) attached to a particular line adapter for an SLC or MLC. Used for the input or output of information for the communications system of which it is a part.

## String Control Byte (SCB)

The String Control Byte is eight bits in length and contains information about the data character strings.

## Switching

Operations involved in interconnecting circuits in order to establish a temporary communication channel between two or more stations.

## Teleprocessing Buffer

A buffer used to contain messages (including the STX and ETX) to be transmitted or received over data communication lines.

## Transparent Mode

An operational mode in which all coded combinations of eight-bit characters are allowed as message characters within a message text.

## Turnaround Time

This is the time required by data sets when switching between transmit and receive modes.

## Unattended Operation

The automatic features of a station's operation that permit the transmission and reception of messages without a computer operator being in attendance.

## Wideband Channel

A channel that is wider in bandwidth than a voice-grade channel.

## APPENDIX B

### HASP TRANSMISSION BLOCK FORMAT

#### TYPICAL HASP TRANSMISSION BLOCK FORMAT

Messages transmitted between the remote and central systems when running HASP are variable in length and contain control characters and data. The entire message received, including control characters, is stored in memory and is referred to as the HASP transmission block.

The format of a typical HASP transmission block for transparent mode is shown below. As noted, several control characters are replaced by other control characters when the transmission mode is non-transparent. Several of the control characters are defined in detail on the following pages of this appendix, and the remaining control characters are defined in the Glossary.

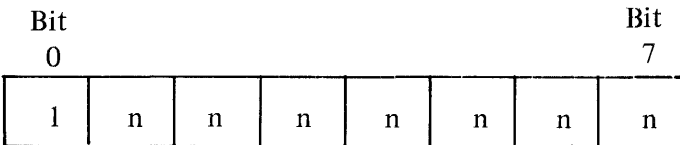
DLE	Binary synchronous communications leader. When operating in non-transparent mode, SOH is used instead of DLE.
STX	Start of Text for binary synchronous transmission.
BCB	Block Control Byte.
FCS	Function Control Sequence.
FCS	Function Control Sequence.
RCB	Record Control Byte for record 1.
SRCB	Subrecord Control Byte for record 1.
SCB	String Control Byte for record 1.
data	Character String for record 1.
SCB	String Control Byte for record 1.
data	Character string for record 1.
SCB	Terminating SCB for record 1.
RCB	Record Control Byte for record 2.
SRCB	SRCB for record 2.
SCB	SCB for record 2.
data	Character string for record 2.
SCB	Ending SCB for record 2.

RCB	Ending Record Control Block for record 2.
DLE	Binary synchronous communications leader. When operating in non-transparent mode, SYN is used instead of DLE.
ETB	Binary synchronous End of Transmission Block.

## DETAILS OF SELECTED CONTROL CHARACTERS

### Block Control Byte

The Block Control Byte (BCB) is eight bits in length and contains the following information about the transmission block status and sequence count:



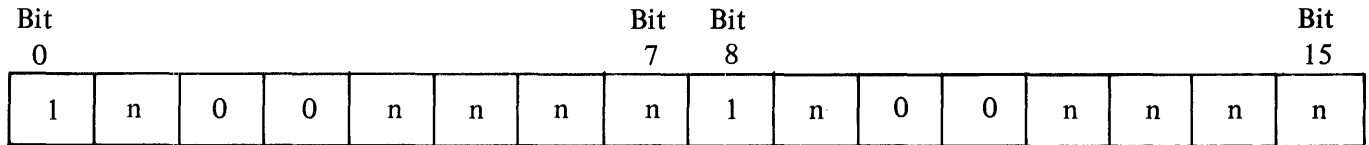
n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1-3		These three bits contain the following control information:
	000	Normal block.
	001	Bypass sequence count validation.
	010	Reset the expected block sequence count to the value contained in bits 4 through 7.
	011	Reserved.
	100	Reserved.
	101	Available for use.
	110	Available for use.
	111	Reserved.
4-7	nnnn	Modulo 16 block sequence count.



## FUNCTION CONTROL SEQUENCE

The Function Control Sequence (FCS) is 16 bits in length and is used to control the flow of individual message streams. A description of the 16 bits of the FCS follows:

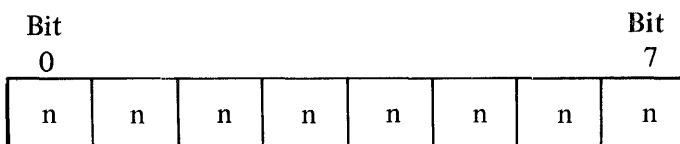


n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1	0	Normal state.
	1	Suspend all stream transmission (WAIT.A.BIT).
2-3	00	Reserved.
4-7	0	Suspend transmission.
	1	Continue transmission.
<p>These four bits (individually) contain function stream identifiers for input message streams. If a bit is ON (1), transmission of the associated message stream is to continue. If a bit is OFF (0), the transmission of the associated message stream is to be suspended.</p>		
8	1	This bit always has a value of 1.
9	0	Suspend remote console stream.
	1	Continue remote console stream.
10-11	00	Reserved.
12-15	0	Suspend transmission.
	1	Continue transmission.
<p>These four bits (individually) contain function stream identifiers for input message streams. If a bit is ON (1), transmission of the associated message stream is to continue. If a bit is OFF (0), the transmission of the associated message stream is to be suspended.</p>		

## RECORD CONTROL BYTE

The Record Control Byte (RCB) is eight bits in length and is used to identify each record type within a transmission block. A description of the eight bits of the RCB follows:



n = binary 0 or 1.

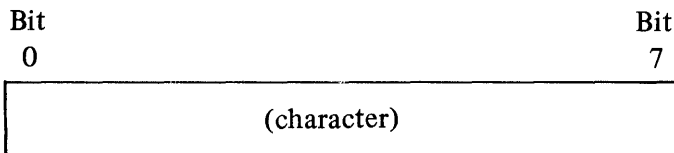
<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0-7	00000000	Indicates that this RCB is functioning as an End of Transmission Block (ETB).
0	1	Indicates that this is a normal RCB.
1-3		If bits 4 through 7 equal 0000, bits 1, 2, and 3 indicate the following control information:
	000	Reserved.
	001	Request to initiate a function transmission.
	010	Permission to initiate a function transmission.
	011	Reserved.
	100	Reserved.
	101	Available for use.
	110	Available for use.
	111	General control record (the type of control record is indicated in the SRCB).
		If bits 4 through 7 are not equal to 0000, bits 1, 2, and 3 are used to identify job streams of multiple, identical functions.
4-7		Bits 4 through 7 are used to identify the type of data record that follows the RCB.
	0000	Control record.
	0001	Operator message display.
	0010	Operator command.
	0011	Normal input record.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
4-7 (cont)	0100	Print record.
	0101	Punch record.
	0110	Data set record.
	0111	Terminal message routing request.
	1000-1100	Reserved.
	1101-1111	Available for use.

### SUBRECORD CONTROL BYTE

The Subrecord Control Byte (SRCB) is eight bits in length and is used in conjunction with a Record Control Byte. The SRCB contains information about the record, depending upon the type of record described by the preceding RCB. The format of the SRCB for five types of records follows:

CASE 1: General format for a control record.

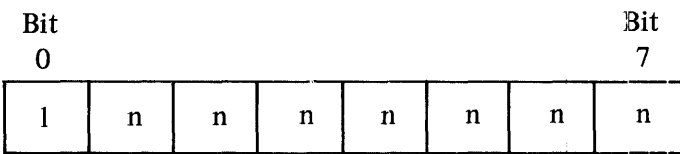


character = A,B,C,D,E,F,G,H,S,T,U,V,W,X,Y,Z

<u>Bit Number</u>	<u>Character</u>	<u>Description or Function</u>
0-7	A	Initial sign-on.
	B	Final sign-off.
	C	Print initialization record.
	D	Punch initialization record.
	E	Input initialization record.
	F	Data set transmission initialization.

<u>Bit Number</u>	<u>Character</u>	<u>Description or Function</u>
0-7 (cont)	G	System configuration status.
	H	Diagnostic control record.
	I-R	Reserved.
	S-Z	Available for use.

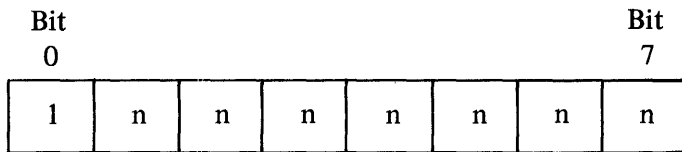
CASE 2: Carriage control information for print records.



n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1	0	Carriage control.
	1	Reserved.
2-3	00	Write before advancing the number of lines indicated by the value in bits 6 and 7.
	01	Write before advancing to the channel indicated by the value in bits 4 through 7.
	10	Space the number of lines indicated by the value in bits 6 and 7.
	11	Skip to channel indicated by the value in bits 4 through 7.
2-7	000000	Suppress spacing.

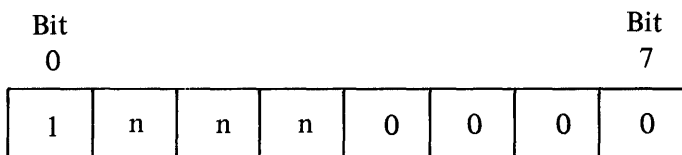
CASE 3: Information for punch records.



n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1-2	00	SCB count for unit 1.
	01	SCB count for unit 2.
	10	SCB count for unit 4.
	11	Reserved.
3	0	Normal EBCDIC card image.
	1	Column binary card image.
4-5	00	Reserved.
6-7	00	Punch stacker select information.
	01	Punch stacker select information.
	10	Punch stacker select information.
	11	Punch stacker select information.

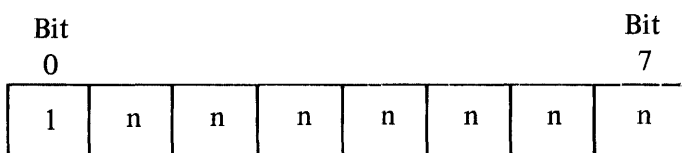
CASE 4: Information for input records.



n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1-2	00	SCB count for unit 1.
	01	SCB count for unit 2.
	10	SCB count for unit 4.
	11	Reserved.
4-7	0000	Reserved.

CASE 5: Destination of a display message.

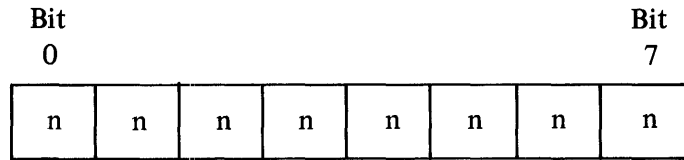


n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0	1	This bit always has a value of 1.
1-7	0000001 to 1100011	Remote system number from 1 to 99 inclusive.
1-7	1100100 to 1111111	Remote group number from 100 to 127 inclusive. Specified in HASPGEN.
1-7	0000000	Broadcast to all remote stations.

### STRING CONTROL BYTE

The String Control Byte (SCB) is eight bits in length and contains information about the data character strings.



n = binary 0 or 1.

<u>Bit Number</u>	<u>Binary Value</u>	<u>Description or Function</u>
0-7	00000000	Indicates end of record.
0-7	10000000	The record is continued in the next transmission block.
0	1	This is a data character string SCB.
1	1	Non-duplicate character string.
1-2	00	Duplicate character is blank. Bits 3 through 7 contain the count of the number of blank characters.
1-2	01	Duplicate character is non-blank and follows the SCB. Bits 3 through 7 contain the duplication count.
2-7	nnnnnn	These six bits contain a value that indicates the character string length.

**APPENDIX C**  
**TIPS AND TECHNIQUES**

**GENERAL**

The material in this appendix is intended to give the user additional information about the B 1800/B 1700 HASP Remote System and includes a subsection that explains the factors that affect throughput to and from the central system. Improved system performance can result from reading and applying the information in this appendix.

Since the B 1800/B 1700 Remote System configuration can vary, no attempt was made to list requirements based on any specific system configuration.

**B 1800/B 1700 REMOTE SYSTEM THROUGHPUT**

System throughput is determined by many factors, some of which can be determined by the user. This subsection describes the factors that affect throughput and shows how some of them are used in formulas to calculate throughput.

**Factors Affecting Throughput**

**ADAPTER OPTIONS**

The B 1800/B 1700 binary synchronous adapter must be strapped for either LEASED or SWITCHED line operation by the field engineer. When strapped for SWITCHED line operation, a 96 millisecond delay occurs between write and read operations. When strapped for LEASED line operation, no delay occurs.

The jumper chip to be strapped is either at chip location I2 on the binary synchronous adapter (number A-2210 8666), or location H2 on the binary synchronous adapter (number A-2216 0717).

The two methods of strapping for LEASED line operation are as follows:

- a. For two-wire operation, the strapping connections must be as follows:

<u>Connection</u>	<u>Description</u>
Pin A to Pin N	Connect for two-wire operation.
Pin F to Pin J	Introduces a four millisecond delay in the write operation before the adapter transmits the first sync code.

- b. For four-wire operation, the strapping connections must be as follows:

<u>Connection</u>	<u>Description</u>
Pin D to Pin L	Holds the Request to Send signal on at all times.



<u>Connection</u>	<u>Description</u>
Pin F to Pin J	Introduces a four millisecond delay in the write operation before the adapter transmits the first sync code.

The method for strapping for SWITCHED line operation is as follows:

<u>Connection</u>	<u>Description</u>
Pin A to Pin P	Switched line operation.
Pin F to Pin J	Introduces a four millisecond delay in the write operation before the adapter transmits the first sync code.

For data sets that have a squelch delay of less than 96 milliseconds, the binary synchronous adapter can be strapped for no delay when strapped SWITCHED by making the following connection:

<u>Connection</u>	<u>Description</u>
Pin D to Pin R	No delay occurs between the write and read operations.

## TYPE OF CONTROL

When the Single Line Control is used, it must be connected to a higher-numbered channel than anything else, with the following exception: if the reader—sorter control is present, it must be on a higher-numbered channel than the Single Line Control.

When the Multi-Line Control is used, it must be connected to any available port. The transfer rate of data from the Multi-Line Control to memory is faster than when the Single Line Control is used, with the result that throughput normally increases.

## DATA SET CHARACTERISTICS

Data sets vary in the number of bits per second at which they can operate, and in their turnaround time. Turnaround time can exceed the time required to send the message. Zero turnaround time is the optimum condition.

## BLOCK SIZE

The block size is an important determinant in achieving optimum throughput. Block size is determined by the number of data characters in each record and the number of records that are to make up a block. The optimum block size can be calculated from the formulas presented in the subsection titled CALCULATING THROUGHPUT.

## DATA COMPRESSION

The COMP.TYPE=n and COMPRESSION.MIN=n run-time parameters are used to specify the packing method to be used by the HASP and HASP/SPOOL programs. The packing methods and the effect they have on performance of the B 1800/B 1700 HASP program are described on the following pages.

The COMP.TYPE=n run-time parameter is described as follows for n = 0, 1, 2, and 3.

#### COMP.TYPE=0

COMP.TYPE=0 requires less processor time to pack a card into the teleprocessing buffer than when n = 1, 2, or 3. The text portion of the record is increased from 80 bytes to 82 bytes to accommodate two one-byte String Control Bytes (SCB).

Since the original card image has grown larger rather than smaller, the number of card images that can be packed into one teleprocessing buffer is reduced. The result is that transmission of data to the central system requires increased line time and reduced processor time.

#### COMP.TYPE=1

COMP.TYPE=1 causes a backward scan from position 80 to position 1 of the card image record. The scan stops when a non-blank character is found. The trailing spaces are truncated and are not sent to the central computer.

The use of this compression type can reduce processor overhead and increase the number of card images that can be packed into a teleprocessing buffer.

#### COMP.TYPE=2

COMP.TYPE=2 causes a backward scan from position 80 to position 1 of the card image record. The scan stops when a non-blank character is found. The trailing spaces are truncated and are not sent to the central computer. The remaining portion of the card image is scanned for the occurrence of two or more consecutive blank characters.

When two or more consecutive blank characters are found, the total number of consecutive blank characters is counted. If the total number of consecutive blank characters is:

- a. Less than 32, a one-character SCB replaces the blank characters.
- b. Greater than or equal to 32 but less than 62, two one-character SCB's replace the blank characters.
- c. Greater than or equal to 62, three one-character SCB's replace the blank character.

The use of this compression type results in increased processor time, but a greater number of card images can be packed into a teleprocessing buffer, thereby reducing total line time.

#### COMP.TYPE=3

COMP.TYPE=3 is logically the same as COMP.TYPE=2, with the following exception: In addition to compressing, consecutive blank characters, consecutive duplicate characters are also compressed. The minimum number of duplicate characters that are compressed is specified by the run-time parameter COMPRESSION.MIN=n.

The use of COMP.TYPE=3 requires more processor time than the other compression types, but is the most efficient method to increase the number of card images that can be packed into a teleprocessing buffer. Since increasing the processor time allows less time to perform other functions (such as unpacking buffers received), this compression type may not be the best method for optimum throughput.

## COMPRESSION.MIN=n

This run-time parameter is used in conjunction with COMP.TYPE=2 or COMP.TYPE=3, and specifies the minimum number of consecutive duplicate characters that must be detected to cause compression. For example, if n has a value of 20, no compression occurs when strings of less than 20 consecutive duplicate characters are detected. The optimum value for n is 4.

When COMPRESSION.MIN=n and COMP.TYPE=3 are specified, the HASP and HASP/SPOOL programs replace the consecutive duplicate characters with a one-byte SCB and the one-byte character that is duplicated. For every occurrence of consecutive duplicate characters equal to or greater than specified by COMPRESSION.MIN=n and less than 32, two bytes are substituted for the duplicates. The constant 31 is the upper limit that the SCB can reference.

## CHARACTER TRANSFER RATE

The line speed has a direct effect on throughput. As line speed increases, the time required to transfer a character decreases. This relationship is shown here:

<u>Line Speed In Bits Per Second</u>	<u>Character Transfer Rate</u>
2,000	4.00 ms/character
2,400	3.33 ms/character
4,800	1.67 ms/character
7,200	1.11 ms/character
9,600	.83 ms/character
19,200	.42 ms/character
50,000	.16 ms/character

### Calculating Throughput

Throughput from the remote system to the central system can be calculated by using the formulas introduced in this subsection.

## CHARACTER TRANSFER RATE

The character transfer rate, expressed in milliseconds per character, is as follows for the commonly used line speeds shown:

<u>Line Speed In Bits Per Second</u>	<u>Character Transfer Rate</u>
2,000	4.00 ms/character
2,400	3.33 ms/character
4,800	1.67 ms/character

<u>Line Speed In Bits Per Second</u>	<u>Character Transfer Rate</u>
7,200	1.11 ms/character
9,600	.83 ms/character
19,200	.42 ms/character
50,000	.16 ms/character

### BLOCK THROUGHPUT RATE

The block throughput rate in milliseconds per block can be calculated with the following formula:

$$\text{Block Throughput Rate} = R * ((N * C) + 11) + T + (R * 6)$$

where R = the character transfer rate in milliseconds.

N = the number of records per block.

C = the number of data characters per record, including the RCB, SRCB, SCB, ending SRCB, and ending RCB.

T = turnaround delay, which is the sum of the transmit and receive in milliseconds.

### RECORD THROUGHPUT RATE

The record throughput rate in records per minute can be calculated with the following formula:

$$\text{Record Throughput Rate} = \frac{N * 60,000}{\text{Block Throughput Rate}}$$

### CONCLUSION

If the calculated record throughput rate is less than the maximum throughput rate of the peripheral device, the throughput rate is said to be line dependent.

If the calculated record throughput rate is greater than the maximum throughput rate of the peripheral device, the throughput rate is said to be device dependent.

### PAD CHARACTER STRAPPING

A pad character is normally added to the end of a message so that the receiving device can check for the reception of the pad character, thereby indicating that all of the message has been received.

The reason for adding a pad character is that in some cases a device can transmit a message and terminate after the Cyclic Redundancy Check (CRC) character. A device can also drop the Request to Send signal one bit time before the last character or CRC character is completely transmitted, resulting in a faulty CRC character.

To detect this problem, IBM specifies that all stations are to:

- a. Transmit a pad character consisting of all ones (@FF@) following the last character or CRC character, and
- b. Receive at least the first four bits of the pad character before the Request to Send signal is dropped.

Since pad characters are not always used or desired, the adapter can be strapped to inhibit sensing and sending them.

To inhibit sensing and sending pad characters, the following connection must be made on the binary synchronous adapter:

<u>Connection</u>	<u>Description</u>
Pin R to Pin S	Inhibits sending and sensing pad characters.

## TRANSMITTING CARD IMAGES

The B 1800/B 1700 HASP program has the ability to simultaneously transmit four card image files to the central computer. The card images can be read from tape, disk, and/or the card reader by the B 1800/B 1700 HASP program. The record size for card image files can vary from 1 to 4096 bytes. If the record size is greater than 80 bytes, the information beyond position 80 is truncated. If the record size is less than 80 bytes, blank fill characters are appended to the end of the record before compression is performed.

The .SD, .SR, and .ST local commands are available to the B 1800/B 1700 HASP operator to start transmitting card image files to the central computer. A reader is either a DISK, TAPE, or CARD input device.

When the .SD, .SR, or .ST local command is given to the B 1800/B 1700 HASP program, the following are performed:

- a. If no reader number is specified, any available reader is used. If no reader is available, the command is saved until a reader becomes available.
- b. If the corresponding file designated by the reader number is open, the local command is disallowed.
- c. The device type of the file designated by the reader number is changed to:
  1. DISK if the local command is .SDn.
  2. TAPE if the local command is .STn.
  3. CARD READER if the local command is .SRn.
- d. If an alternate file name is specified, the external label of the file designated by the reader number is changed. If the alternate file name format is invalid, the local command is disallowed.

- e. If the file is not present the local command is disallowed.
- f. If the file designated by the reader number is open the local command is disallowed.

#### **RUN-TIME PARAMETER MAX.RDR**

The number of readers assigned to the B 1800/B 1700 HASP program is determined by the run-time parameter MAX.RDR. If multiple readers are assigned to the B 1800/B 1700 HASP program, the following rules apply:

- a. The maximum number of readers open at one time cannot exceed the number assigned by MAX.RDR.
- b. If a reader is open, any attempt to start the same reader is not allowed.
- c. When more than one reader is open at the same time, a read is made from one of the reader's input files and the card image is packed into one of the two output buffer areas in dynamic memory. The next read and succeeding reads are made from the input files of the other readers, and the card images are packed into the same output buffer area as the card image from the first read.

The teleprocessing buffer transmitted to the central computer can contain multiple reader records. The records from each reader are distinguished by a unique Record Control Byte (RCB) that is a part of each card image record within a teleprocessing buffer. The central computer creates separate job streams from card image records that have the same Record Control Bytes.

- d. If multiple readers are open at the same time and one reader goes not ready, the B 1800/B 1700 HASP program continues to read from all other active readers.

For a correlation of the B 1800/B 1700 HASP file numbers, internal file names, reader numbers, and RCB assignments, refer to table 4-1.

## APPENDIX D

### HASP CONTROL LOOP

#### GENERAL

The B 1800/B 1700 HASP program is controlled by a complex WAIT statement. The WAIT statement provides an interface with the Data Comm line, console printer, card reader, line printer, and card punch. Accordingly, the control loop of HASP consists of a WAIT on I/O complete on any one of the above devices. When HASP is awakened, it recognizes the event that completed and takes appropriate action.

#### SIMPLE CONTROL LOOP

The following SDL statements are an example of a simple control loop that is similar to the actual control loop.

Example:

```
CASE WAIT (TIME.TENTHS(x),
          DC.IO.COMPLETE,
          SPO.INPUT.PRESENT,
          READ.OK (CARDS),
          WRITE.OK (PRINTER),
          WRITE.OK (PUNCH));
TIME.COMPLETE;
DCH;                                % DC.IO.COMPLETE
PROCESS.SPO.MESSAGE;                % GOT SPO MESSAGE
PACK.AND.SEND.CARDS;                % CARD READ COMPLETE
OUTPUT.QUEUED.RECORD;                % OUTPUT COMPLETE PRINTER
OUTPUT.QUEUED.RECORD;                % OUTPUT COMPLETE PUNCH
END CASE;
```

The purpose of the simple control loop example is to illustrate the flow of control, but not the detailed workings of HASP.

## COMPLEX CONTROL LOOP

The actual control loop for HASP is more complex than this example for the following reasons.

- a. HASP must handle several input and output streams. Each input stream has a READ.OK (*<filename>*) event associated with it.
- b. The complex WAIT statement must have a mechanism (not shown in the example) to turn off a stream. If the card reader is much faster than the Data Comm Line, then there must be some way of ignoring card reader interrupts until the Data Comm Line catches up. This mechanism is provided as a special attribute of the WAIT statement and is referred to as the LOCK mechanism.
- c. It is necessary that the procedure OUTPUT.QUEUED.RECORD have some mechanism to distinguish between printer and punch interrupts. This mechanism is not shown in the previous example.

The following paragraphs describe the flow and major functions of HASP.

### Initialization of a Reader

After HASP execution has begun, .SRn can be entered from the console printer to start reading a card stream. This action activates the event SPO.INPUT.PRESENT. HASP wakes up and branches to the PROCESS.SPO.MSG procedure, which does an ACCEPT to receive the message, and checks the syntax of the message. After determining that the message is a request to start stream, HASP obtains permission to send the stream from the central computer. When permission is granted, the READ.OK (*<card file>*) event is UNLOCKED. The MCPH then initiates reading of cards, which causes the READ.OK (*<card file>*) event to come true.

### Read and Compress Card Images

The complex WAIT statement then branches to the PACK.AND.SEND.CARD procedure. This procedure reads and compresses the card image and moves the result into the Data Comm buffer. Control is then returned to the complex WAIT statement to await more card images. This cycle continues until the Data Comm buffer is filled, at which time the buffer is queued for transmission to the central computer.

### Data Comm Buffer Handling

HASP maintains two Data Comm buffers for input. While one buffer is being transmitted on the Data Comm line, the other buffer is being filled with card images. This overlapping of Data Comm I/O and reading card images increases throughput. When both buffers are filled (unavailable), HASP sends a WAIT.A.BIT command to the central computer, suspending further transmission. At some later time, when one of the buffers becomes available, HASP resets the WAIT.A.BIT command and transmission is resumed. When both Data Comm buffers are full, the card stream is LOCKED and no further interrupts (completion of READ.OK events) are caused. When one of the buffers becomes available, the card streams are UNLOCKED and the card compression cycle continues. Card images from different input streams are packed into the same Data Comm Buffer, and consequently, HASP does not maintain two Data Comm buffers per stream, but rather just two Data Comm buffers for all input streams.

When HASP has received a buffer from the central computer, the procedure named DCH does the appropriate line protocol. HASP then branches to the procedure named UNPACK.AND.PACK.RECORDS, which unpacks each output record and queues it in an internal queue buffer. If a buffer contains ten packed records, then all ten records are unpacked and queued before returning to the control loop.



The queue mechanism just described is bypassed if the output queue buffer is empty, and HASP attempts to write the record directly to the output device. If the write is unsuccessful, the record is written into the internal queue buffer. All records unpacked after this are also inserted into the queue. When any output is queued for a particular stream, that stream is UNLOCKED. This action causes the event WRITE.OK (*output file*) to come true, and the WAIT statement branches to the procedure named OUTPUT. QUEUED.RECORD, which removes the record from the top of the appropriate queue and does a WRITE to the output device. Control is then returned to the complex WAIT statement.

**Queue Buffers**

The queue buffer that holds output records intended for printer or punch devices is maintained by stream number. Records inserted in this queue buffer may be in memory or on disk. Figure D-1 illustrates the location of queue buffers in memory.

**Dynamic Memory**

The dynamic memory assigned by the HASP/MODIFIER program can be changed at run time, which can affect the number of queue buffers in memory. If the dynamic memory is not large enough to hold the four Data Comm buffers, then HASP aborts due to insufficient dynamic memory. If there is sufficient dynamic memory only for the four Data Comm buffers, then all queuing occurs on disk.

Example:

```
EX HASP ME 50000;
```

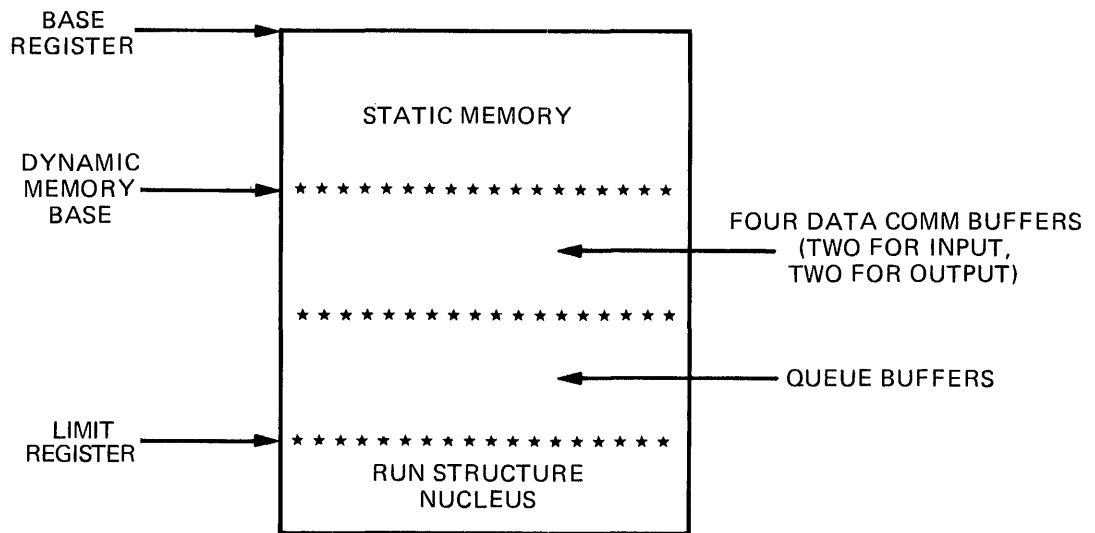


Figure D-1. Queue Buffer Location in Memory

## **Print and Punch Output Procedure**

Since output records for all streams share queue buffers, a coded mechanism has been provided to prevent one stream from using all memory buffers. A threshold (UPPER.Q.LIMIT) has been established above which all output records for a particular stream are written to disk. As an example, if a system has 50 queue buffers, 3 streams, and UPPER.Q.LIMIT=35, then no stream may have more than 35 records in memory. UPPER.Q.LIMIT cannot be set by a user and is always greater than 20. The formula for determining UPPER.Q.LIMIT is dependent upon the number of streams, the number of queue buffers, and the Data Comm buffer size. If the Data Comm buffer size increases, then so does UPPER.Q.LIMIT.

When the number of output records in a queue buffer for a stream becomes equal to UPPER.Q.LIMIT, the WAIT.A.BIT condition for that stream is set. Thus, in the following transmission the central computer is asked not to send any more buffers for this stream. All subsequent output records for this stream are queued to disk. The maximum size of the disk queue per stream is 240 records. When this limit is reached, HASP continues to run but in a degraded mode. The WAIT.A.BIT condition for any stream is reset when the number of output records for the stream falls below  $UPPER.Q.LIMIT / 2$ , and the central computer then resumes transmission of output files for that particular stream.

## INDEX

- .AUDIT, 4-11, 4-31
- .BOTH, 5-4, 5-5, 5-7 thru 5-9
- .CLOSE, 4-7, 4-11, 4-19, 5-6 thru 5-9
- .EOF, 4-15
- .EOFn, 4-12
- .FINI, 4-11
- .GO, 4-15
- .GOn, 4-12
- .HRn, 4-12
- .NEW-SPOOL, 4-16, 4-20, 4-31
- .PUNCH, 4-21
- .PUNCH=96, 4-18, 4-25
- .PUNCH=nn, 4-14, 4-20, 4-25
- .RO *<option>*, 4-17
- .RS1, 5-7, 5-8
- .RS1 *<filename>*, 5-9
- .S, 4-14, 4-26
- .SD1, 4-31
- .SD1 *<filename>*, 4-16
- .SD, 4-15, 4-23, 4-27, C-6
- .SDn *<filename>*, 4-14
- .SO *<option>*, 4-17
- .SP1, 5-5, 5-7, 5-13
- .SP1 *<filename>*, 5-8, 5-9
- .SR1, 5-5, 5-6, 5-14
- .SR1 *<filename>*, 5-8, 5-9
- .SR, 4-15, 4-16, 4-21, 4-23, 4-27, C-6
- .SRn *<filename>*, 4-14
- .ST1 *<filename>*, 4-16
- .ST, 4-15 thru 4-17, 4-23, 4-27, 4-31, C-6
- .STn *<filename>*, 4-14
- .WHAT, 4-17
  
- \$BSCCPU, 2-2
- \$MLBFSIZ, 2-2
- \$NUMLNES, 2-2
- \$NUMRJE, 2-2
- \$NUMTPBF, 2-2
- \$NUMTPPR, 2-2
- \$NUMTPPU, 2-2
- \$NUMTPRD, 2-2
  
- /\*SIGNOFF, 3-8 thru 3-11
- /\*SIGNON, 3-8 thru 3-11, 3-14
- //LOGOFF, 3-8, 3-9, 3-11
- //LOGON, 3-8, 3-9, 3-11
  
- ACK0, 4-6, 4-19
- adapter options, C-1
  
- ADDRESS=, 3-3, 3-13
- ASCII-graphic, 4-25, 5-12
- assigning RCB'S, 4-32
- AUDIT, 3-3, 4-31
  
- B 1800/B 1700 HASP program, 1-1, 2-2, 4-1
- binary synchronous adapter, 4-2, 4-4, 4-5
- block control byte, 4-19, B-2
- BUFFER, 3-2, 3-13
- buffer size, 3-12
- BUFFER=n, 3-4, 3-12
  
- card image transmission, C-6
- central computer, 4-1, 4-16, 4-17, 4-19, 4-23, 5-1
- central system, 2-1, 3-5, 5-11
- change forms processing, 4-24, 5-11
- channel skips on line printer, 4-23, 5-20
- COMP.TYPE, 3-4, 3-13, C-2
- COMPRESSION.MIN, 3-4, 3-14, C-2
  
- data compression, 4-6
- DC/AUDIT, 3-3, 4-11, 4-31
- DC/AUDIT.FILE, 3-3, 4-11, 4-20
- direct mode, 3-1, 3-4, 3-6, 3-12, 4-1 thru 4-3, 4-6, 4-7, 4-11, 4-14, 4-16, 4-24 thru 4-32
- DMPALL, 3-8, 3-9
- DONT.EX, 3-1, 3-2, 3-4
- dynamic memory, 3-6
  
- function control sequence, B-3
  
- glossary, A-1
  
- handshaking, 4-6
- HASP, 2-2, 3-1 thru 3-6, 3-8, 3-10, 3-12, 3-13, 3-15, 4-1, 4-2, 4-4 thru 4-11, 4-14 thru 4-19, 4-21 thru 4-27, 4-31, 5-2, 5-7, 5-11
- HASP central system commands, 4-10
- HASP communication with the central system, 4-5
- HASP complex control loop, D-1
- HASP console keyboard input commands, 4-10
- HASP console printer output messages, 4-18
- HASP control loop, D-1
- HASP default run-time parameters, 4-4
- HASP dynamic memory, 3-12
- HASP dynamic memory modification, 3-12
- HASP dynamic memory size, 3-13
- HASP file information, 4-26
- HASP file modification, 3-12

## INDEX (Cont)

- HASP file number 0, 4-26
- HASP file number 1, 4-27
- HASP file number 2, 4-27
- HASP file number 3, 4-27
- HASP file number 4, 4-27
- HASP file number 5, 4-28
- HASP file number 6, 4-28
- HASP file number 7, 4-28
- HASP file number 8, 4-28
- HASP file number 9, 4-29
- HASP file number 10, 4-29
- HASP file number 11, 4-29
- HASP file number 12, 4-29
- HASP file number 13, 4-29
- HASP file number 14, 4-30
- HASP file number 15, 4-30
- HASP file number 16, 4-30
- HASP file number 17, 4-30
- HASP file number 18, 4-31
- HASP file number 19, 4-31
- HASP file number 20, 4-31
- HASP file number 21, 4-31
- HASP file number 22, 4-32
- HASP file number 23, 4-32
- HASP information reception from the central system, 4-6
- HASP information transmission, 4-6
- HASP local commands, 4-11
- HASP local output messages, 4-18
- HASP modification, 3-12
- HASP operating exceptions, 4-23
- HASP operating instructions and information, 4-4
- HASP operating modes, 4-1
- HASP printer/punch backup capability, 4-24
- HASP program operation, 4-4
- HASP program termination - direct mode, 4-7
- HASP program termination - spool mode, 4-9
- HASP session, 2-2, 3-3, 3-8, 4-4, 4-32, 5-2
- HASP simple control loop, D-1
- HASP summary statistics report, 4-33
- HASP system overview, 2-1
- HASP system requirements, 4-2, 4-4
- HASP transmission block format, B-1
- HASP/CARDS, 4-16, 5-2, 5-5, 5-6, 5-9, 5-14
- HASP/CARDSIN, 5-5, 5-6, 5-9, 5-14
- HASP/MODIFIER, 3-1 thru 3-4, 3-6, 3-8, 3-10, 3-12, 3-15, 4-20
- HASP/MODIFIER console printer output messages, 3-13
- HASP/MODIFIER default run-time parameters, 3-3, 3-8
- HASP/MODIFIER functional description, 3-1
- HASP/MODIFIER operating instructions and information, 3-3
- HASP/MODIFIER program execution, 3-9
- HASP/MODIFIER run-time parameter list, 3-3
- HASP/MODIFIER run-time parameters, 3-9
- HASP/MODIFIER system configuration, 3-2
- HASP/MODIFIER system requirements, 3-2
- HASP/PARMS, 3-8, 3-10, 3-14, 3-15
- HASP/PARMS creation, 3-8
- HASP/SAVE.PARMS, 3-1, 3-2, 3-5, 3-15, 4-19, 4-20, 4-23, 5-10, 5-19
- HASP/SPOOL, 2-2, 3-1, 3-4 thru 3-6, 3-10, 3-12, 3-13, 3-15, 4-2, 4-9, 4-24, 5-1 thru 5-13, 5-20
- HASP/SPOOL console keyboard input commands, 5-8
- HASP/SPOOL console printer messages, 5-8, 5-9
- HASP/SPOOL dynamic memory modification, 3-13
- HASP/SPOOL file information, 5-13
- HASP/SPOOL file number 0, 5-13
- HASP/SPOOL file number 1, 5-14
- HASP/SPOOL file number 2, 5-14
- HASP/SPOOL file number 3, 5-14
- HASP/SPOOL file number 4, 5-15
- HASP/SPOOL file number 5, 5-15
- HASP/SPOOL file number 6, 5-15
- HASP/SPOOL file number 7, 5-16
- HASP/SPOOL file number 8, 5-16
- HASP/SPOOL file number 9, 5-16
- HASP/SPOOL file number 10, 5-17
- HASP/SPOOL file number 11, 5-17
- HASP/SPOOL file number 12, 5-18
- HASP/SPOOL file number 13, 5-18
- HASP/SPOOL file number 14, 5-18
- HASP/SPOOL file number 15, 5-19
- HASP/SPOOL file number 16, 5-19
- HASP/SPOOL file number 17, 5-19
- HASP/SPOOL input spool creation mode, 5-1
- HASP/SPOOL input spool mode, 5-2, 5-4 thru 5-6, 5-8, 5-14
- HASP/SPOOL operating exceptions, 5-20
- HASP/SPOOL operating instructions and information, 5-3
- HASP/SPOOL operating modes, 5-1
- HASP/SPOOL output spool mode, 5-4, 5-5, 5-7, 5-8, 5-15 thru 5-19
- HASP/SPOOL output spool printing and punching mode, 5-1, 5-2
- HASP/SPOOL printer/punch backup capability, 5-20
- HASP/SPOOL program execution, 5-4
- HASP/SPOOL program operation, 5-4
- HASP/SPOOL system requirements - input spool mode, 5-2

## INDEX (Cont)

- HASP/SPOOL system requirements - output spool mode, 5-2
- HASPGEN, 2-2
- HASPGEN parameters, 2-1
- HASPIN, 4-31
- HASPOUT, 4-31
- HEREP, 4-7, 4-9, 4-11, 4-32, 4-33
  
- IBM central system, 1-1, 2-1
- IBM HASP messages, 4-23
- IBM HASP program, 1-1
- IBM object-code files, 3-6, 4-14, 4-25, 5-11, 5-12
- ID=, 3-4, 3-14
  
- LEASED, 3-5
- line discipline, 4-1, 4-2
- LIST, 3-5
- LIST.ONLY, 3-5
  
- MAX.PCH, 4-22
- MAX.PRT, 3-2, 3-5, 3-13, 3-14, 4-22
- MAX.PRT=n, 3-5, 3-12
- MAX.PUN, 3-2, 3-5, 3-13, 3-14
- MAX.PUN=n, 3-5, 3-12
- MAX.RDR, 3-14, C-7
- MAX.RDR=n, 3-6
- maximum printers, 3-12
- maximum punches, 3-12
- MCPII, 2-2, 3-2, 4-2, 4-4, 4-24, 5-1, 5-2, 5-20
- MEM=nnnnnn, 3-6
- MLC, 4-5
- multi-leaving, 2-1, 2-2
- multi-line control, 3-3, 4-2, 4-4, 4-19
  
- NEWSPOOL, 4-17
- non-transparent, 3-6
- NONTRANS, 3-6
- notation convention, 1-1
  
- pad character strapping, C-5
- password, 3-11
- PBD, 4-24, 5-20
- primary input file, 4-15
- PRINTER=, 3-6, 3-14
- punch width, 3-12
  
- PUNCH=80, 4-14
- PUNCH=96, 4-14, 5-11, 5-12
- PUNCH=nn, 3-6, 3-12, 3-14, 5-12
  
- R, 3-10, 3-14
- RCB, 4-21, 5-10, 5-15 thru 5-19
- record control byte, 4-19, 4-32, B-3
- related documents, 1-1
- REMOTE, 3-7
- remote system, 2-2, 4-14
- remote system configuration, 4-2, 4-4, 5-2, 5-3
- remote system throughput, C-1
  
- SCB, 5-10
- secondary input file, 4-15
- sign-off, 3-6, 4-7
- sign-on, 3-4, 3-5, 3-6, 3-8, 3-10
- sign-on card, 3-10
- signoff, 3-10, 4-19
- SIGNOFF=a, 3-7, 3-9, 3-10, 3-11
- signon card, 3-11
- SIGNON=a, 3-7, 3-9, 3-10, 3-11
- single line control, 3-3, 4-2, 4-4, 4-19
- SLC, 4-5
- SOHENQ, 4-6, 4-20
- SPOOL, 3-2, 3-7
- spool file, 4-8, 4-9, 4-21, 5-2, 5-11
- spool file transmission, 4-8
- spool files reception, 4-9
- spool mode, 3-1, 3-6, 3-12, 4-1, 4-2, 4-4, 4-8, 4-9, 4-11, 4-21, 4-24, 4-26, 4-31, 4-32, 5-13
- spooling, 2-1, 2-2, 5-1
- SRCB, 5-10
- string control byte (SCB), 4-20, B-8
- subrecord control byte (SRCB), B-5
- SWITCHO, 3-10
- switched lines, 4-5
  
- throughput calculation, C-4
- tips and techniques, C-1
- transparent mode, 3-6
  
- YES, 3-10, 3-14
  
- ZIP, 3-2