BRS REFERENCE MANUAL

Issued March 3, 1967
Revised August 28, 1967
Revised November, 1967

BY

Ann Hardy
D.L. Schmidt
A.P. Scott
V.E. Van Vlear

TABLE OF CONTENTS

1.0 FORKS, SCHEDULING AND PROGRAM INTERACTION

1.1 Activation, Dismissal And Termination Of Forks

1- 2	9	Start fork
1- 3	31	Wait for specified fork to cause a panic
1- 4	45	Dismiss on quantum overflow
1- 5	81	Dismiss for specified amount of time
1- 6	10	Terminate a fork
1- 7	90	Declare a fork for termination on escape

1.2 Program Interrupts

1- 8	78	Arm/disarm software interrupts
1- 9	79	Cause a program interrupt
1-10	49	Read interrupts armed
1-11	135	Cause a program interrupt after specified period of time

2.0 INPUT/OUTPUT

2.1 Open Files

2- 2	15	Read input file name
2- 3	16	Open input file
2- 4	18	Read output file name
2- 5	19	Open output file
2- 6	48	Look up input/output file name
2- 7	60	Look up input/output file name and insert in file directory if necessary
2- 8	151	Open a commands from file

2.2 Close Files

2- 9	20	Close a file
2-10	17	Close all files
2-11	147	Close all except commands from file

2.3 Input/Output Operations

2-12	CIO	Character input/output
2-13	WIO	Word input/output
2-14	BIO	Block input/output
2-15	CIT	Character input and test
2-16	95	Write a dump file

2.4 Miscellaneous File Operations

2-17	110	Read device and unit
2-18	113	Compute file size of a disk file
2-19	41	Return from input/output subroutine
2-20	142	Get file date and access count

3.0 TELETYPE INPUT/OUTPUT OPERATIONS

3- 2	TCI	Teletype character input
3- 3	TCO	Teletype character output
3- 4	13	Test input buffer for empty
3- 5	11	Clear input buffer
3- 6	14	Dismiss until output buffer empty
3- 7	29	Clear output buffer
3- 8	12	Declare echo table or set 8-level input
3- 9	40	Read echo table
3-10	85	Set 8-level output
3-11	86	Clear 8-level output
3-12	134	Set line feed or carriage return ignore
3-13	74	Set 1/2-duplex mode
3-14	75	Set full-duplex mode
3-15	76	Skip on 1/2-duplex mode

4.0 MEMORY OPERATIONS

4-2	4	Release a page of memory
4-3	121	Release a page of memory
4-4	43	Read pseudo-relabeling
4-5	44	Set pseudo-relabeling
4-6	116	Read program relabeling
4-7	117	Set program relabeling
4-8	80	Make page read only
4-9	70	Count free user pages

5.0 STRING PROCESS

5.1 String Input/Output

5- 2	33	Read string
5- 3	34	Output message
5- 4	35	Output string

5.2 Hash Table Search

5- 5	5	Look up string in hash table
5- 6	6	Insert string in hash table

5.3 String Manipulation

5- 7	STP	Store string pointer
5- 8	LDP	Load string pointer
5- 9	SKSE	Skip if string equal
5-10	SKSG	Skip if string greater

5.4 Character Manipulation

5-11	GCI	Get character from beginning of string and increment beginning pointer
5-12	WCI	Put character on end of string and increment end pointer
5-13	GCD	Get character from end of string and decrement end pointer
5-14	WCD	Put character on beginning of string and decrement beginning pointer
5-15	WCH	Write character to memory by table

6.0 NUMBER OPERATION

6.1 Number Input/Output

6- 2	36	Output number to specified radix
6- 3	38	Input number to specified radix
6- 4	52	Formatted floating point input
6- 5	53	Formatted floating point output
6- 6	SIC	String to internal conversion
6- 7	ISC	Internal to string conversion

6.2 Arithmetic Operations

6- 8	50	Conversion from floating point to fixed point
6- 9	51	Conversion from fixed point to floating point
6-10	21	Floating point negate
6-11	FAD	Floating point addition
6-12	FSB	Floating point subtract
6-13	FMP	Floating point multiplication
6-14	FDV	Floating point division

8.0 SYSTEM AND SUBSYSTEM RESTRICTED BRS'S

8.1 Escape Control

8- 2	46	Turn escape off
8- 3	47	Turn escape on
8- 4	26	Skip if escape waiting

8.2 Fork Management

8- 5	72	Exec dismiss
8- 6	22	Prevent termination
8- 7	23	Allow termination
8- 8	109	Dismiss
8- 9	111	Terminate class 3 BRS fork
8-10	139	Give fork system status

8.3 Input/Output

8-11	1	Open a file or device
8-12	2	Close a file
8-13	8	Close all files
8-14	148	Close all but a specified file
8-15	66	Delete disk file data
8-16	67	Delete a specified disk block
8-17	87	Read disk file index block
8-18	124	Read disk
8-19	125	Write disk
8-20	114	Turn off run-away magnetic tape
8-21	118	Allocate magnetic tape unit
8-22	119	De-allocate magnetic tape unit
8-23	CTRL	Input/output control
8-24	130	Test a break point switch

8.4 Teletype Control

8-25	126	Test for carrier presence
8-26	129	Turn channel on or off
8-27	112	Turn off teletype station
8-28	152	Ignore off interrupts

8.5 Memory Control

8-29	120	Assign PMT entry
8-30	3	Make PMT pointer indirect

8.6 EXEC Command BRS's

8-31	96	Recover
8-32	127	Look at memory

8.7 Miscellaneous

8-33	88	Read execution time
8-34	37	General string look up
8-35	128	Set disk bit map
8-36	131	Crash system
8-37	136	Set exec switches
8-38	141	Get exec subroutines
8-39	143	Start gathering statistics
8-40	61	Start clock statistics
8-41	144	Get a buffer
8-42	145	Return a buffer
8-43	97	Reset subsystem counter
8-44	98	Increment subsystem counter
8-45	99	Read subsystem counter

9.0 COMPLETE LIST OF ALL BRS'S AND SYSOP'S

1.0 FORKS, SCHEDULING AND PROGRAM INTERACTION

NUMBER: 9

NAME: FKST

FUNCTION: Start Fork

STATUS: User

CALLING SEQUENCE:
```
LDA   T
BRS   9
```

T = Address of a "Panic Table" (See Appendix A, Glossary). Bits 0 through 5 of register A have the following significance:

0 = Give fork system status if current fork has system status.
1 = Set fork relabeling from panic table. Otherwise use current relabeling.
2 = Propogate escape assignment to fork (See BRS 90).
3 = Make fork fixed memory. It is not allowed any more memory than it started with.
4 = Make fork local memory. New memory will be assigned to it independent of the controlling fork.
5 = Give fork subsystem status if current fork has subsystem status.

DESCRIPTION: BRS 9 is used to create dependent entries in the PAC table. The panic table indicated by register A must not be the same for two forks of the same job or overlap a page boundary; if it is, BRS 9 is illegal. BRS 9 creates a new fork as a fork of the fork creating it, which is called the controlling fork. The fork is lower in the hierarchy of forks than the controlling fork. The controlling fork may itself be a fork of some still higher fork. For more detailed information see Document No. 30.10.31.

When BRS 9 is executed by a fork with user or subsystem status, the controlling fork is dismissed until the lower fork terminates. A user may not have more than eight forks in his fork structure. This includes the EXEC fork and one fork for each EXEC BRS that is active. Only one EXEC BRS can be active at a time.

REGISTERS AFFECTED: None

NUMBER: 31

NAME: FKWT

FUNCTION: Wait For Fork To Cause A Panic

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   31
```

P = Panic Table Address

DESCRIPTION: Causes the controlling fork to be dismissed until the lower fork causes a panic (see Appendix A, Glossary). When it does, the controlling fork is reactivated at the instruction following this BRS, and the panic table contains the status of the fork on its dismissal. The status is also put into the X register. The panic table address is put into the A register.

REGISTERS AFFECTED: X, A

NUMBER: 45

NAME: SQO

FUNCTION: Dismiss On Quantum Overflow

STATUS: User

CALLING SEQUENCE: BRS 45

DESCRIPTION: This BRS causes the user to be dismissed as though he had overflowed his quantum. It guarantees that the next time he is started he will have a complete short time quantum.

REGISTERS AFFECTED: None

NUMBER: 81

NAME: WREAL

FUNCTION: Dismiss For Specified Amount Of Time

STATUS: User

CALLING SEQUENCE:
```
LDA   T
BRS   81
```

T = Dismissal time in milliseconds.

DESCRIPTION: The fork is dismissed for the number of milliseconds specified in A. The fork is never activated sooner than the delay requested; but it will generally not be activated at exactly the time requested. See also BRS 135.

REGISTERS AFFECTED: A

NUMBER: 10

NAME: PPAN

FUNCTION: Programmed Panic. Terminates A Fork.

STATUS: User

CALLING SEQUENCE: BRS 10

BRS 10 terminates the fork that issues it and returns control to the higher fork. It is like typing ESCAPE on the teletype.

DESCRIPTION: Terminates a fork. This condition can be distinguished from a panic caused by the escape key only by the fact that in the former case the program counter in the panic table points to a word containing BRS 10. This BRS would normally be used to terminate a fork when it is finished.

REGISTERS AFFECTED: None

NUMBER: 90

NAME: DFR

FUNCTION: Declare A Fork For "Escape"

STATUS: User

CALLING SEQUENCE: BRS 90

DESCRIPTION: In case the user types ESCAPE, this is the highest fork to terminate. If this fork has armed Interrupt 1, that interrupt will be taken instead of terminating the fork.

REGISTERS AFFECTED: None

NUMBER: 78

NAME: SAIR

FUNCTION: Arm/Disarm Software Interrupts

STATUS: User

CALLING SEQUENCE:
```
LDA   M
BRS   78
```

M is the complete new interrupt mask.

DESCRIPTION: The new interrupt mask is substituted for the old one. A user may arm Interrupts 1-10. A system status fork may interrupt 11 also. Interrupt 1 is in bit 4 of the mask word. The Interrupts are as follows:

1 Interrupt if Program Panic (BRS 10 or ESCAPE)

2 Interrupt if Memory Panic

3 Interrupt if Lower Fork terminates

4 Interrupt if any I/O condition occurs which sets a flag bit (0, 7 or 8 in file number word)

11 Interrupt if DSU error

5 through 10 Interrupts on condition set by user

Location 200 octal plus the interrupt number must be set to point to a routine to process the interrupt. When the interrupt occurs an SBRM* is executed to the location pointed to. If it is desired to return to the point in the program interrupted, the user must BRR to the location where the return was saved.

Example:

```
   SET          INTERRUPT ROUTINE      RETURN
LDA=ESCAPE     ESCAPE ZRO ESCRTN    BRR ESCRTN
STA 201B                 .
 .                       .
 .                       .
```

REGISTERS AFFECTED: None

NUMBER: 79

NAME: SIIR

FUNCTION: Cause Interrupt

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   79
```

N = Interrupt number. N has the range of 5 to 10.

DESCRIPTION: The search through the fork structure begins with the fork just above the one which issued the BRS 79 and proceeds to the higher level forks in sequence. The interrupt will be caused in the first fork found which has the interrupt armed. If no fork has the interrupt armed, it is treated like a NOP. This would normally be used to cause Interrupts 5 through 10 to interrupt. All forks below the one with the interrupt armed will be terminated.

REGISTERS AFFECTED: None

NUMBER: 49

NAME: SRIR

FUNCTION: Read Interrupts Armed

STATUS: User

CALLING SEQUENCE: BRS 49

DESCRIPTION: Reads the interrupt mask into the A register. Bit 4 corresponds to Interrupt number 1, 5 to number 2 and etc. There are 11 programmable interrupts. See also BRS 78.

REGISTERS AFFECTED: A

NUMBER: 135

NAME: TIMINT

FUNCTION: Interrupts A Fork After A Specified Period Of Time

STATUS: User

CALLING SEQUENCE:
```
LDA   M
LDB   T
LDX   N
BRS   135
NORMAL RETURN
```

M is the new interrupt mask.
T is the time in milliseconds after which the fork will be interrupted.
N is the interrupt number.

DESCRIPTION: The fork issuing this BRS will be interrupted after the delay if the interrupt specified by N is armed at that time. If a fork gives this BRS again with the same N before the time has passed, the new time will be set. A fork may have a maximum of three timing interrupts pending simultaneously. All forks below the one receiving the interrupt will be terminated. See also BRS 81.

REGISTERS AFFECTED: None

2.0 INPUT/OUTPUT

NUMBER: 15

NAME: GFN

FUNCTION: Reads Input File Name From A Command File And Looks Up The File Name In The User's File Directory

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   15
EXCEPTION RETURN
NORMAL RETURN
```

N = Command File Number = 0 For Teletype Input

DESCRIPTION: The routine ignores leading spaces, leading multi-blanks, and leading carriage return characters. The exception return is taken if the input file name string cannot be located in the file directory.

Exception Return:	X:	Pointer to the input file name string pointers.
	A & B:	Input file name string pointers.
Normal Return:	A:	Pointer to the string pointers of the desired file in the file directory hash table.
	B:	The value word of the hash table entry.
	X:	Destroyed.

NOTE: The information contained in the registers cannot be used directly by the user since the addresses are in the T.S. Block; this BRS is normally followed by the BRS 16.

If the input file name string begins with a left paren, or with the full quote, the file name will be located in another user's file directory or in the public file directory, respectively.

REGISTERS AFFECTED: All

NUMBER: 68 (Exec D7 Monitor D10)

FUNCTION: Returns File Name String pointer and File Parameter for other Exec File BRS's

STATUS: User

CALLING SEQUENCE:
```
LDP   P
LDX   A
BRS   68
EXCEPTION RETURN
NORMAL RETURN
```

P and P+1	Null String pointers
A	Index

DESCRIPTION: Given a number A in X between 0 and Q where Q corresponds to the last file in the file directory, the BRS returns in B the ending string pointer of a file name if A is a valid index. If A is an invalid index, A and B are set to zero. A parameter is returned in the A register for a valid index that allows the BRS 16 or BRS 19 to be called and open the file. If the number A is greater than Q (corresponding to the last valid file), the exception return is taken; otherwise the normal return is taken.

REGISTERS AFFECTED: A, B

SAMPLE PROGRAM: (Types all file names)

```
START   CLX
        LDP   P
        BRS   68
        BRS   10
        SKE   =0
        BRU   *+3
S1      EAX   1,2
        BRU   START+1
        STX   SV
        LDX   =1
        LDA   P
        BRS   35
        LDX   SV
        TCO   =155B
        BRU   S1
```

(See also BRS 69)

NUMBER: 69

FUNCTION: Delete File

STATUS: User

CALLING SEQUENCE:

```
CLA
BRS   15
BRU   ERROR
BRS   69
BRU   ERROR (Error return)
(normal return)
```

DESCRIPTION: If preceded by an appropriate File Name BRS (BRS 15, 18, 48, 60, 68), the BRS 69 will delete the file, release the storage and cause the file directory to be updated on the disk. Error return if the file is Read Only or cannot be opened.

REGISTERS AFFECTED: All

SAMPLE PROGRAM: (Deletes all the files in a file directory)

```
START  CLX
       STX   SV
       LDP   P      (null string pointers)
       BRS   68
       BRS   10
       BRS   69
       BRU   ERROR
       LDX   SV
       EAX   1,2
       BRU   START+1
```

NOTE:

NEW FILE CONTROL BRS's (In EXEC 7)

BRS 62 Replaces 15 and 16. Requires same arguments as BRS 15.

BRS 63 Replaces 18 and 19. Requires same argument as BRS 18. In addition, requires file type in X (same as BRS 19).

BRS 64 Replaces 48 and 16. Requires same arguments as BRS 48.

BRS 65 Replaces 60 and 19. Requires same arguments as BRS 60. In addition, requires file type in X (same as BRS 19). This BRS also may be used as follows:

In X Register

Bits 0-11 File type (1-4)
Bits 12-23 Command Input File

In this case the BRS will type OLD FILE or NEW FILE and wait for a Carriage Return or Line Feed confirmation.

NUMBER: 16

NAME: GIFNB

FUNCTION: Open Input File

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   16
EXCEPTION RETURN
NORMAL RETURN
```

N = File Directory Pointer Address

DESCRIPTION: Opens an input file. The BRS requires in A the value returned in A by a BRS 15, 48, or 60. The exception return is taken if the pointer in A is not pointing to a proper location or if the file cannot be opened for any reason, such as a physical device that cannot be an input file.

```
Exception Return:   All registers destroyed
Normal Return: A:   File Number
               B:   File Type (0-4)
               X:   File Size
```

REGISTERS AFFECTED: All

NUMBER: 18

NAME: GOFNA

FUNCTION: Reads File Name From A Command File And Looks Up The File Name In The User's File Directory. The Command File Must Be An Input File.

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   18
EXCEPTION RETURN
NORMAL RETURN
```

N = Command File Number = 0 for Teletype Input.
If Bit 1 = 1 in the A Register, the BRS assumes a file name is correct and does not type "OLD FILE" or "NEW FILE".

DESCRIPTION: This BRS ignores leading spaces, leading multi-blanks characters, and leading Carriage Returns. There are two types of file names which BRS 18 looks up: those which are surrounded by single quotes or slashes, and those which are not.

Single quote and slash files: If the string begins with a single quote or slash, it must be terminated by the same character which must then be followed by a confirming Carriage Return. The exception exit is taken if these requirements are not met. If the string is found in the file directory, the message OLD FILE is typed, otherwise, the message NEW FILE is typed. If the next character in the input string is a Line Feed, Carriage Return, or period, the normal return will be taken; otherwise, the exception return is taken. In the case of a new file, the file name is inserted conditionally into the file directory.

Other files: The string is looked up in the file directory and if it is not found, the exception return is taken. The NEW FILE/OLD FILE message is never typed. (New output file names must always be surrounded by either slashes or single quotes.)

If the file is read-only, the exception return is taken.

```
Exception Return:    All destroyed.
Normal Return: A:    Location of the file in the directory
                     hash table.
               B:    Confirming character in case of a quote
                     or slash file; otherwise, the file
                     directory.
               X:    Destroyed.
```

REGISTERS AFFECTED: All

NUMBER: 19

NAME: GOFNB

FUNCTION: Opens An Output File

STATUS: User

CALLING SEQUENCE:
```
LDA   N1
LDB   N2 (for tape files only)
LDX   N3
BRS   19
EXCEPTION RETURN
NORMAL RETURN
```

N1 = Information supplied in A by BRS 18, BRS 48 or BRS 60. (Location in file directory.)
N2 = Information supplied in X by BRS 16. (File size for tape files only.)
N3 = File type. (See Appendix A, Glossary.)

DESCRIPTION: Opens an output file. On the normal return, A = the file number. The exception return is taken if:

1) The word in A is not a valid pointer.
2) The file cannot be opened. (Such as a physical device that cannot be used for output.)
3) The file directory is already full.

In these cases an appropriate message is typed.

REGISTERS AFFECTED:
```
Exception Return: All destroyed.
Normal Return:    A = file number
                  B and X are destroyed.
```

NUMBER: 48

NAME: GSFN

FUNCTION: Look Up Input/Output File Name

STATUS: User

CALLINE SEQUENCE:
```
LDP   N
BRS   48
EXCEPTION RETURN
NORMAL RETURN
```

N = String pointers for the file name.

DESCRIPTION: The file name is looked up in the file directory. If it is not there, the exception return is taken.

Exception Return:	A & B	- No change.
	X	- The number of matches.
Normal Return:	A & B	- Location in file directory. Can be used by BRS 16 or BRS 19.
	X	- Destroyed.

REGISTERS AFFECTED: All

NUMBER: 60

NAME: GSFI

FUNCTION: Looks Up A File Name In The File Directory And Inserts It If It Is Not There

STATUS: User

CALLING SEQUENCE:
```
LDP   N
BRS   60
EXCEPTION RETURN
NORMAL RETURN
```

N = String pointers for the file name.

DESCRIPTION: The file name is looked up in the file directory. If it is not there, the name is inserted. The exception return is taken if the file directory is full.

```
Exception Return: A & B= No change
                     X= -1
Normal Return:    A & B= Location in file directory. (Can
                         be used by BRS 16 and BRS 19.)
                     X= Destroyed.
```

REGISTERS AFFECTED: All

NUMBER: 151

NAME: OPCMF

FUNCTION: Open A Commands-From File

STATUS: User

CALLING SEQUENCE:
```
LDA   A
BRS   151
```

A = File number of Commands-From file.

DESCRIPTION: Closes the Commands-From file if one is open and sets the Commands-From file to the file specified in A.

REGISTERS AFFECTED: None

NUMBER: 20

NAME: CFILE

FUNCTION: Close A File

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   20
```

N = File Number

DESCRIPTION: The "close file" BRS is used to indicate to the system all processing is completed on this file. If the file number indicates Mag Tape, the file will be terminated and if output, the End of File will be written; but in either case, the tape will be positioned at the start of the next file and the tape is de-allocated. All registers are destroyed.

REGISTERS AFFECTED: All

NUMBER: 17

NAME: UABORT

FUNCTION: Close All Files

STATUS: User

CALLING SEQUENCE: BRS 17

DESCRIPTION: If magnetic tape has been used, the last record will be terminated and if output, the End of File will be written; in either case the tape will be positioned at the start of the next file. The tape is then closed and the unit is deallocated. All registers are destroyed.

REGISTERS AFFECTED: All

NUMBER: 147

NAME: BS147

FUNCTION: Closes All Except Commands-From File

STATUS: User

CALLING SEQUENCE: BRS 147

DESCRIPTION: Closes all users' files except the Commands-From file.

REGISTERS AFFECTED: None

NUMBER: 161

NAME: CIO

FUNCTION: Character Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   C (Output Only)
CIO   N
```

C = 8 bit data character right justified
N = Address of word containing a file number

DESCRIPTION: CIO is used to input or output a single character from, or to, a file from the A register. On input an End of File condition will set bits 0 and 7 in the file number and return a 137_8 character. If Interrupt 4 is armed (see BRS 78), it will occur. The End of File condition occurs on the next input operation after the last character of the file. If an error occurs, bits 0 and 6 will be set in N and Interrupt 4 will occur if it is armed.

WIO and BIO should not be mixed with CIO to read or write a given file.

REGISTERS AFFECTED: A

NUMBER: 160

NAME: WIO

FUNCTION: Word Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   D (Output Only)
WIO   N
```

D = Data word to be written
N = Address of word containing a file number

DESCRIPTION: WIO is used to input or output a word of data to or from the A register. On input an End of File condition returns a word of three 137_8 characters and sets bits 0 and 7 in the file number word. If Interrupt 4 is armed, it will occur. If an End of File condition occurs with a partially filled out word, the word is completed with 137_8 characters. If an error occurs, bits 0 and 6 are set in N. If Interrupt 4 is armed it will occur.

CIO and WIO should not be mixed to read or write a given file.

REGISTERS AFFECTED: A.

NUMBER: 176

NAME: BIO

FUNCTION: Blocked Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   W
LDX   I
BIO   N
EXCEPTION RETURN
NORMAL RETURN
```

I = Starting memory address
W = Number of words to be read or written
N = Address of word containing a file number

DESCRIPTION: BIO is used to input a block of words to memory or output a block of words from memory. The A register will contain the first memory location not read into or out of at the end of the operation. If the operation is completed successfully, control will be transferred to the normal return; otherwise, control will be transferred to the exception return.

On input an End of File condition will set bits 0 and 7 in the file number. An error will set bits 0 and 6. Interrupt 4 will occur if armed when any of these bits are set.

Exception conditions are:

1. End of File
2. Bad Data

REGISTERS AFFECTED: A,X

NUMBER: 134

NAME: CIT

FUNCTION: Character Input And Test

STATUS: User

CALLING SEQUENCE:
```
LDA   N
CIT   F
EXCEPTION RETURN
NORMAL RETURN
```

N = Character to be tested
F = File Number (see CIO) (Input Only)

DESCRIPTION: The character in the A register is compared against the next character in the input file. If it compares, the normal return is taken and the character is removed from the input buffer. If it does not compare, the character is left in the input buffer and is returned in A.

Exception Return:	A	- The next character in the input buffer.
	B & X	- No change.
Normal Return:	A	- The character supplied remains in A (the character is removed from the input buffer).

REGISTERS AFFECTED: A

NUMBER: 95

NAME: ECDUMP

FUNCTION: Dump

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   95
```

N = File Number

DESCRIPTION: This BRS writes the entire current state of the machine (user's program only) on the specified file, which is made type 4. The status of the pseudo-relabeling registers and all information necessary to restart the user from his current situation are written on the dump file so that it can be restored by a recovery procedure.

REGISTERS AFFECTED: All

NUMBER: 110

NAME: RDU

FUNCTION: Read Device And Unit

STATUS: User

CALLING SEQUENCE:
```
LDA    =FILE No.
BRS    110
NORMAL RETURN
```

DESCRIPTION: Output X = Device number
A = Unit number

REGISTERS AFFECTED: A, X

NUMBER: 113

NAME: DFCD

FUNCTION: Compute File Size Of A Disk File

STATUS: User

CALLING SEQUENCE:
```
LDA   =File Number
BRS   113
NORMAL RETURN
```

DESCRIPTION: Adds the number of data words in the file to the number in the X register. Returns the result in X.

REGISTERS AFFECTED: X

NUMBER: 41

NAME: IORET

FUNCTION: Return From Input/Output Subroutine

STATUS: User

CALLING SEQUENCE: BRS 41

DESCRIPTION: This is used by the author of an Input/Output subroutine to return to the calling program.

REGISTERS AFFECTED: A

NUMBER: 142

NAME: GTFDT

FUNCTION: Gets File Date And Access Count

STATUS: User

CALLING SEQUENCE:
```
LDA   N1
LDB   N2
BRS   142
```

N1 = Information supplied in A by BRS 15 or BRS 48
N2 = Information supplied in B by BRS 15 or BRS 48

DESCRIPTION: Returns access count in B and a 9 bit creation date in A. The high order 4 bits are month from 0 - 11 and low order 5 bits are date from 0 - 30.

REGISTERS AFFECTED: A, B

3.0 TELETYPE INPUT/OUTPUT OPERATIONS

NUMBER: 174

NAME: TCI

FUNCTION: Teletype Character Input

STATUS: User

CALLING SEQUENCE: TCI M

M = Memory location

DESCRIPTION: This SYSPOP reads the character from the teletype input buffer and places it into the location M right justified. The remainder of location M is cleared. The character is also placed in the A register right justified.

REGISTERS AFFECTED: A

NUMBER: 175

NAME: TCO

FUNCTION: Teletype Character Output

STATUS: User

CALLING SEQUENCE: TCO M

M = Memory address

DESCRIPTION: This SYSPOP outputs the character from the rightmost eight bits of location M to the controlling teletype. In addition to the ordinary ASCII characters, all teletype output operations will accept 135_8 as a multiple blank character. The next character will be taken as a blank count, and the indicated number of blanks will be typed.

REGISTERS AFFECTED: None

NUMBER: 13

NAME: SKI

FUNCTION: Test Input Buffer For Empty

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   13
EXCEPTION RETURN
NORMAL RETURN
```

T = Teletype number (-1 is used to indicate the controlling teletype)

DESCRIPTION: This BRS tests for the presence of input characters in the buffer. If the buffer is empty, control is transferred to the "normal return". If there are any characters in the input buffer, control is transferred to the "exception return".

REGISTERS AFFECTED: None

NUMBER: 11

NAME: CIB

FUNCTION: Clear The Teletype Input Buffer

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   11
```

T = Teletype number (-1 is used to indicate the controlling teletype)

DESCRIPTION: Sets the buffer pointers to indicate there are no characters in the teletype input buffer.

REGISTERS AFFECTED: None

NUMBER: 14

NAME: DOB

FUNCTION: Dismiss Until The Teletype Output Buffer Is Empty

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   14
```

T = Teletype number (-1 is used to indicate the controlling teletype)

DESCRIPTION: Dismiss this fork until the teletype output buffer indicated is empty. It is dismissed until the last interrupt is received.

REGISTERS AFFECTED: None

NUMBER: 29

NAME: COB

FUNCTION: Clear The Output Buffer

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   29
```

T = Teletype number (-1 indicates the controlling teletype)

DESCRIPTION: Sets the buffer pointers to indicate there are no characters in the teletype output buffer.

REGISTERS AFFECTED: None

NUMBER: 12

NAME: CET

FUNCTION: Declare Echo Table Or Set 8-Level Mode

STATUS: User

CALLING SEQUENCE:
```
LDX   T
DLA   R
BRS   12
```

T = Teletype number (-1 is used to indicate the controlling teletype)

R = 0,1,2, or 3 to indicate the proper echo table. R may also contain any eight-bit character if the sign bit is on. In this case, each eight bit character read from the teletype is transmitted unchanged to the user's program. No echoes are generated while in this special eight-level mode. The eight bit character is the character on which eight-level mode will terminate. If the sign bit is ON and bit 15 is ON, there will be no terminating character. The program must stop the reading itself. If it fails to do this, the user may hang up his phone. Neither the ESCAPE nor high-speed ESCAPE will terminate the program. Otherwise, this is like regular eight-level mode.

DESCRIPTION: BRS 12 sets the echo table for the teletype indicated by Register X. Echo tables are as follows:

0 = Echo each character just as it was received and break on all characters.

1 = Same echo as 0 but all characters except letters, digits and spaces are break characters.

2 = Same echo as 0, but the only break characters are control characters (including carriage return and line feed). Note: the EXEC sets this for the user.

3 = No echo for any character and break on all characters.

REGISTERS AFFECTED: None

NUMBER: 40

NAME: RDET

FUNCTION: Read Echo Table

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   40
```

T = Teletype number

DESCRIPTION: Reads the echo table number (0,1,2,3) into the A register.

If the teletype is not in eight-level input mode, reads the echo table number (0,1,2,3) into the A register. If the teletype is in eight-level mode, the sign bit of A is set, the address field contains the terminal character.

REGISTERS AFFECTED: A

NUMBER: 85

NAME: SET8P

FUNCTION: Set Special Teletype Output

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   85
```

T = Teletype number (-1 is used to indicate controlling teletype)

DESCRIPTION: Sets teletype to eight-level output mode. The teletype specified must be the controlling teletype. Eight-level is transmitted to the teletype exactly as it is received from the user program. A fork sending eight-level code should do a BRS 14 before terminating.

REGISTERS AFFECTED: None

NUMBER: 86

NAME: CLR8P

FUNCTION: Clear Special Teletype Output

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   86
```

T = Teletype number (-1 is used to indicate controlling teletype)

DESCRIPTION: Puts the teletype output back into normal mode. The teletype specified must be the controlling teletype.

REGISTERS AFFECTED: None

NUMBER: 134

NAME: CRSW

FUNCTION: To Allow The User To Ignore Line Feed Or Carriage Return When It Follows A Carriage Return Or Line Feed

STATUS: User

CALLING SEQUENCE:
```
LDX =-1
LDA  =0 (ignore) =-1 (do not ignore)
BRS  134
NORMAL RETURN
```

DESCRIPTION: The contents of the A register will give the following results. If A is negative, all line feeds and carriage returns received from the teletype will be sent to the program and echoed. If A is positive, a line feed after a carriage return received from the teletype will be ignored (not sent to the program and not echoed) and a carriage return after a line feed will be ignored (not sent to the program and not echoed). In all cases the first line feed or carriage return received will be sent to the program and echoed plus echo its compliment.

REGISTERS AFFECTED: None

NUMBER: 74

NAME: SNDEKO

FUNCTION: Puts Users Teletype Into 1/2 Duplex Mode

STATUS: User

CALLING SEQUENCE: BRS 74

DESCRIPTION: Puts users teletype into 1/2 duplex mode.

REGISTERS AFFECTED: None

NUMBER: 75

NAME: RNDEKO

FUNCTION: Puts Users Teletype Into Full Duplex Mode

STATUS: User

CALLING SEQUENCE: BRS 75

DESCRIPTION: Puts users teletype into full duplex mode. This is the normal mode.

REGISTERS AFFECTED: None

NUMBER: 76

NAME: TNOEKO

FUNCTION: Tests For 1/2 Duplex Mode

STATUS: User

CALLING SEQUENCE: BRS 76
RETURN1: Full duplex mode
RETURN2: 1/2 duplex mode

DESCRIPTION: Skips if channel is in 1/2 duplex mode.

REGISTERS AFFECTED: None

4.0 MEMORY OPERATIONS

NUMBER: 4

NAME: MPT

FUNCTION: Release A Page Of Memory

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   4
```

N = Contains any address in the page to be released

DESCRIPTION: The PMT entry for the block is cleared and in any other fork which has this PMT byte in its relabeling, the byte is cleared to 0.

REGISTERS AFFECTED: None

NUMBER: 121

NAME: DPMTE

FUNCTION: Release Specified PMT Entry

STATUS: User

CALLING SEQUENCE:
```
LDA   R
BRS   121
```

R = Relabeling byte of the page to be released

DESCRIPTION: Releases the specified page from the PMT. It is exactly like a BRS 4 except that it takes a byte number instead of an address.

Instruction Trap:

1) Byte not in PMT.
2) A user fork tried to release a system page.

REGISTERS AFFECTED: None

NUMBER: 43

NAME: RDRL

FUNCTION: Read Pseudo-Relabeling

STATUS: User

CALLING SEQUENCE: BRS 43

DESCRIPTION: Reads the current pseudo-relabeling registers into registers A and B.

REGISTERS AFFECTED: A, B

NUMBER: 44

NAME: STRL

FUNCTION: Set Pseudo-Relabeling

STATUS: User

CALLING SEQUENCE:
```
LDA   R1
LDB   R2
BRS   44
```

R1 & R2 = Relabeling registers

DESCRIPTION: This BRS takes the contents of registers A and B and stores them into the current pseudo-relabeling registers. It also causes the real relabeling to be reset to correspond to the new pseudo-relabeling.

This BRS will result in an instruction trap for any of the following reasons:

1) Swapping in the new pages was not completed. (Usually because of a RAD failure.)
2) The user tried to relabel over a system page.
3) The user tried to relabel over a page he did not have. (This is not the way to obtain more memory.)

REGISTERS AFFECTED: None

NUMBER: 116

NAME: RURL

FUNCTION: Read Program Relabeling

STATUS: User

CALLING SEQUENCE: BRS 116

DESCRIPTION: Puts the program relabeling into A and B. This is what the system executive uses as program relabeling. It is kept in the TS block.

REGISTERS AFFECTED: A, B

NUMBER: 117

NAME: SURL

FUNCTION: Set Program Relabeling

STATUS: User

CALLING SEQUENCE:
```
LDA   RL1
LDB   RL2
BRS   117
```

RL1 and RL2 are the new values for the program relabeling

DESCRIPTION: Sets the program relabeling in the TS block as specified. User programs should use BRS 44 to set relabeling for a fork.

Instruction Trap:

1) A specified relabeling byte was not assigned.
2) A user fork tried to relabel a system byte.

REGISTERS AFFECTED: None

This is the program relabeling typed by the STATUS command. It should correspond to the relabeling of the first non-subsystem fork below the EXECUTIVE. If the fork is running under DDT, DDT will update this relabeling. Otherwise, it is the responsibility of the user. It is particularly important that this relabeling be set correctly before issuing a DUMP command.

NUMBER: 80

NAME: MBRO

FUNCTION: Make Page Read Only

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   80
```

P = PMT/SMT number
If bit 0 of A = 1, make page read only.
If bit 0 of A = 0, make page read-write.

DESCRIPTION: Sets the read-write status of the entry according to the value of A. An SMT entry can only be changed by a system status fork. The former status of the entry is returned in A.

Instruction trap:

1) Specified entry is not in use.
2) The RAD failed.

REGISTERS AFFECTED: A

NUMBER: 70

NAME: FRPMT

FUNCTION: Counts Number Of Free User Pages

STATUS: User

CALLING SEQUENCE: BRS 70

DESCRIPTION: Returns the number of free user pages in the A register. This is the number of pages that are available to one user.

REGISTERS AFFECTED: A

5.0 STRING PROCESS

NUMBER: 33

NAME: GETSTR

FUNCTION: Read String

STATUS: User

CALLING SEQUENCE:
```
LDA   A
LDB   T
LDX   F
BRS   33
```

A = Address of string pointer
T = Terminal character
F = File number
Bit 0 of A on = The string is taken as null with the second pointer equal to the first.

DESCRIPTION: This BRS reads characters from the file and appends them to the string until the terminal character is reached. The terminal character is not appended to the string. It returns the updated string pointers in the A and B registers and updates the end string pointer in memory.

REGISTERS AFFECTED: A, B

NUMBER: 34

NAME: OUTMSG

FUNCTION: Output Message

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   W
LDB   C
BRS   34
```

F = File number
W = Beginning word address
C = Character count or -1

DESCRIPTION: This BRS outputs C consecutive characters starting with the first character of the specified word. If B = -1, characters are output until a / is encountered; the character $ is interpreted as a carriage return and line feed.

REGISTERS AFFECTED: None

NUMBER: 35

NAME: OUTSTR

FUNCTION: Output String

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   P
LDB   P+1
BRS   35
```

F = File number
P, P+1 = A string pointer pair

DESCRIPTION: Outputs the string indicated by the string pointers in registers A and B to the specified file.

REGISTERS AFFECTED: None

NUMBER: 5

NAME: SSCH

FUNCTION: Look Up String In Hash Table

STATUS: User

CALLING SEQUENCE:
```
LDA   P
LDB   P+1
LDX   T
BRS   5
EXCEPTION RETURN
NORMAL RETURN
```

P and P+1 = String pointers for a string to be looked up
T = Address of a three word table of the form:

```
ZRO Hash Table Beginning Address
ZRO Hash Table End Address
ZRO 0
```

DESCRIPTION: BRS 5 searches the hash table for a string to match the string indicated by A and B registers. If successful it returns in register B the address of the hash table string pointers, and in register A, the string "value" and executes the "normal" return. Otherwise, it executes the "exception" return with registers A, B and X unchanged and the address of the next free hash table entry in word 3 of the table which is pointed to by register X. (Word 3 will be -1 if the table is full.) The "value" is the hash image for this string.

See BRS 6.

REGISTERS AFFECTED: A, B

NUMBER: 6

NAME: SSIN

FUNCTION: Insert String In Hash Table

STATUS: User

CALLING SEQUENCE: A, B, and X must have the output from BRS 5
BRS 6

DESCRIPTION: BRS 6 inserts the string pointer into the hash table at the point determined by the last BRS 5 which did not find a match. If the hash table is full (word 3 of the table pointed to by X is -1) an "Illegal Instruction" trap results. BRS 6 is intended for use in conjunction with BRS 5. It should be used only after BRS 5 has failed to find a match. Furthermore, string pointers should not be placed in the hash table in any manner other than with BRS 6 (otherwise, the scanning algorithm used in BRS 5 may cause undesired results).

BRS 6 does not physically move the string to which registers A and B point. On return, register B contains the address of the first word of the new hash table entry and register A contains the "value" word of the entry.

REGISTERS AFFECTED: A, B

NUMBER: 167

NAME: STP

FUNCTION: Store Pointers

STATUS: User

CALLING SEQUENCE: STP A

A = Address of a string pointer pair

DESCRIPTION: This SYSPOP is generally used in conjunction with LDP. It stores the contents of the A and B registers into the string pointers indicated in the calling sequence.

REGISTERS AFFECTED: None

NUMBER: 166

NAME: LDP

FUNCTION: Load Pointers

STATUS: User

CALLING SEQUENCE: LDP A

A = Address of a string pointer pair

DESCRIPTION: This SYSPOP loads the string pointers indicated in the calling sequence into the A and B registers.

REGISTERS AFFECTED: None

NUMBER: 163

NAME: SKSE

FUNCTION: Skip On String Equal

STATUS: User

CALLING SEQUENCE:
```
LDA   B
LDB   E
SKSE  A
EXCEPTION RETURN
NORMAL RETURN
```

A = Address of a string pointer pair
B = Beginning string pointer
E = End string pointer

DESCRIPTION: If the string addressed by the pointers in the A and B registers is identical with the string addressed by A of the calling sequence, control will be transferred to the normal return. Otherwise, control will be transferred to the exception return. If the strings are of different lengths or have different contents, control will be transferred to the exception return.

REGISTERS AFFECTED: None

NUMBER: 162

NAME: SKSG

FUNCTION: Skip On String Greater

STATUS: User

CALLING SEQUENCE:
```
LDA   B
LDB   E
SKSG  A
EXCEPTION RETURN
NORMAL RETURN
```

B = Beginning string pointer
E = End string pointer
A = Address of a string pointer pair

DESCRIPTION: This SYSPOP compares the string indicated by A and B registers with the string indicated by A of the calling sequence, character by character and terminates with the first unequal character. The numerical internal code representation of characters is used to determine inequality. If the strings are equal for the entire length of the shorter one, the longer one is indicated as greater. If the contents of the string addressed by the A and B registers is greater than the contents of the string addressed by A, control will be transferred to the normal return. Otherwise, control is transferred to the exception return.

REGISTERS AFFECTED: None

NUMBER: 165

NAME: GCI

FUNCTION: Get Character And Increment

STATUS: User

CALLING SEQUENCE:
```
GCI   A
EXCEPTION RETURN
NORMAL RETURN
```

A = Address of a string pointer pair

DESCRIPTION: This SYSPOP reads into the A register the first character from the string indicated by the beginning string pointer given in the calling sequence. If the string is null or empty, nothing is done and control is transferred to the exception return. If the string is not null its first character is loaded into the A register right-justified, and the beginning string pointer is incremented by one so that the beginning string pointer now points to the string with the first character deleted. Control is transferred to the normal return. Unless a copy of the original pointer is saved, the contents of the string are effectively destroyed.

REGISTERS AFFECTED: A

NUMBER: 157

NAME: WCI

FUNCTION: Write Character And Increment

STATUS: User

CALLING SEQUENCE: WCI P

P = Address of string pointer pair

DESCRIPTION: WCI writes the character in the A register on the end of the string addressed by the end string pointer. The end string pointer is incremented by one.

REGISTERS AFFECTED: B

NUMBER: 137

NAME: GCD

FUNCTION: Get Character And Decrement

STATUS: User

CALLING SEQUENCE:
```
GCD   P
EXCEPTION RETURN
NORMAL RETURN
```

P = Address of a string pointer pair.

DESCRIPTION: A GCD is, in every way, similar to GCI except that the character is taken from the end of the specified string.

The last character on the string is loaded in the A register, and end string pointer is decremented so that it points to the previous character in the string. Control is transferred to the exception return if the end pointer is not greater than the beginning pointer before it is decremented.

REGISTERS AFFECTED: B

NUMBER: 135

NAME: WCD

FUNCTION: Write Character And Decrement

STATUS: User

CALLING SEQUENCE: WCD P

P = Address of a string pointer pair

DESCRIPTION: This SYSPOP writes the character in the A register on the beginning of the string and decrements the beginning string pointer.

REGISTERS AFFECTED: B

NUMBER: 164

NAME: WCH

FUNCTION: Write Character

STATUS: User

CALLING SEQUENCE:
```
LDA   C
WCH   T
```

C = A character right-justified in the A register.
T = The address of a three word table. The table is as follows:

```
Word 0 = A character address
Word 1 = A character address
Word 2 = A transfer address
```

DESCRIPTION: This SYSPOP tries to write a character into the area defined by the character addresses in the table. Provided that the second address in the table is greater than the first address, WCH will write the character in the A register into the character position indicated by the first character address plus one and will increment the first character address in the table. If the first character address is equal to or greater than the second character address in the table the character is not written and control is transferred to the third word of the table with A and X registers undisturbed and the address of the WCH in the B register. The address in the third word of the table can be an exit to a routine which allocates more memory or garbage collects the remaining characters.

REGISTERS AFFECTED: B

6.0 NUMBER OPERATION

NUMBER: 36

NAME: OUTNUM

FUNCTION: Output Number

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   N
LDB   R
BRS   36
```

F = File number
N = Number to be output
R = Radix

DESCRIPTION: Outputs a number in the radix R. The number will be output as an unsigned 24 bit integer. If the radix is less than 2, an instruction trap will be given.

REGISTERS AFFECTED: None

NUMBER: 38

NAME: GETNUM

FUNCTION: Read Number

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDB   R
BRS   38
```

F = File number
R = Radix

DESCRIPTION: Inputs an integer to any radix. The number may be preceded by a plus or minus sign. On exit the number will be in the A register. The conversion is terminated by any non-numeric character which will be in the B register on exit. The number is computed by multiplying the number obtained at each stage by the radix and adding the new digit.

REGISTERS AFFECTED: A, B

NUMBER: 52

NAME: FFI

FUNCTION: Formatted Input

STATUS: User

CALLING SEQUENCE:
```
LDX   FORMAT
BRS   52
BRU   X
```

DESCRIPTION: This routine reads characters from a file specified in the format word, FORMAT. FORMAT also specifies the format of the input. Free form input from the teletype results when FORMAT = 0. A skip return is generated if and only if (1) the input is free form, and (2) the input is floating point. The internal translation of the input file is stored in A, B.

REGISTERS AFFECTED: A, B, X

NUMBER: 53

NAME: FFO

FUNCTION: Formatted Output

STATUS: User

CALLING SEQUENCE:
```
LDX   FORMAT
BRS   53
```

DESCRIPTION: The integer in A or the double word floating point value in A, B is output to the file according to the file number and format specified in FORMAT.

REGISTERS AFFECTED: None

NUMBER: 141

NAME: SIC

FUNCTION: String To Internal Conversion

STATUS: User

CALLING SEQUENCE:
```
LDX   FORMAT
SIC   POINTER
BRU   INTEGER
BRU   FLOATING
```

DESCRIPTION: See String Processing System documents. FORMAT describes the type of conversion to be done.

The contents of POINTER point to the character immediately preceding the character string. POINTER+1 contains the character address of the last character of the string.

INTEGER and FLOATING are routines that handle the converted input. Error flags, if applicable, are in the index register. A double word value corresponding to the string is in A,B upon return.

REGISTERS AFFECTED: A, B, X

NUMBER: 140

NAME: ISC

FUNCTION: Converts Internal Numbers To Formatted Output Strings

STATUS: User

CALLING SEQUENCE:
```
LDP    M
LDX    FORMAT
ISC    POINTER
```

DESCRIPTION: See String Processing Documents. FORMAT describes the type of conversion to be done. The contents of POINTER point to the character immediately preceding the character string. POINTER+1 contains the character address of the character immediately preceding the position where the first character of output is to go. M,M+1 contain the floating point word to be converted. POINTER+1 is incremented once for each character added to the string.

REGISTERS AFFECTED: A, B, X

NUMBER: 50

NAME: FFIX

FUNCTION: Conversion From Floating Point To Fixed Point

STATUS: User

CALLING SEQUENCE: BRS 50

DESCRIPTION: Fixes the double word floating point value in (A,B). The integer part is left in A. The fractional part is left adjusted in B.

REGISTERS AFFECTED: A, B

NUMBER: 51

NAME: FFLT

FUNCTION: Conversion From Fixed Point To Floating Point

STATUS: User

CALLING SEQUENCE: BRS 51

DESCRIPTION: The integer in A is converted to a normalized floating point value in A,B.

REGISTERS AFFECTED: A, B

NUMBER: 21

NAME: FNA

FUNCTION: Floating Negate

STATUS: User

CALLING SEQUENCE: BRS 21

DESCRIPTION: The double word floating point value in the A and B registers is negated.

REGISTERS AFFECTED: A, B

NUMBER: 156

NAME: FAD

FUNCTION: Floating Point Addition

STATUS: User

CALLING SEQUENCE: FAD M

DESCRIPTION: (A,B)+(M,M+1)
A floating addition is performed to the contents of memory location M and M+1 and the A and B registers. The results are left in the A and B registers.

REGISTERS AFFECTED: A, B

NUMBER: 155

NAME: FSB

FUNCTION: Floating Point Subtraction

STATUS: User

CALLING SEQUENCE: FSB M

DESCRIPTION: (A,B) - (M,M+1)
The contents of memory locations M and M+1 are subtracted (floating subtraction) from the contents of the A and B registers. The results are left in the A and B registers.

REGISTERS AFFECTED: A, B

NUMBER: 154

NAME: FMP

FUNCTION: Floating Point Multiplication

STATUS: User

CALLING SEQUENCE: FMP M

DESCRIPTION: (A,B)*(M,M+1)
The contents of memory locations M and M+1 are multiplied (floating multiplication) by the A and B registers and the results left in the A and B registers.

REGISTERS AFFECTED: A, B

NUMBER: 153

NAME: FDV

FUNCTION: Floating Point Divide

STATUS: User

CALLING SEQUENCE: FDV M

DESCRIPTION: (A,B)/(M,M+1)
The contents of the A and B registers are divided (floating divide) by the contents of memory locations M and M+1 with the quotient left in the A and B registers.

REGISTERS AFFECTED: A, B

7.0 MISCELLANEOUS

NUMBER: 42

NAME: RREAL

FUNCTION: Read Real Time Clock

STATUS: User

CALLING SEQUENCE: BRS 42

DESCRIPTION: Sets the contents of the A register equal to the value of the real time clock. Time is given as a 24 bit binary number representing 60ths of a second. The clock is set to zero when the system is started and it is incremented by one at every 1/60th second. A binary form of the start-up time is put in B. The first four bits of B are the month number. The rest of the bits are the minute of the month. From A and B the user can calculate the month, date and time.

REGISTERS AFFECTED: A, B

NUMBER: 91

NAME: EXRTIM

FUNCTION: Read Date And Time Into A String

STATUS: User

CALLING SEQUENCE:
```
LDA   S
LDB   S+1
BRS   91
```

S = Beginning string pointer
S+1 = Ending string pointer

DESCRIPTION: The current date and time are appended to the string provided in A and B registers and the resulting string pointers are returned in the A and B registers. The characters appended to the string have the form:

MM/dd hh:mm

MM = Month
dd = Day
hh = Hours counted from 0 to 24
mm = Minutes

REGISTERS AFFECTED: B

NUMBER: 39

NAME: RCPW

FUNCTION: Reads Control Parameter Word And AUNN

STATUS: User

CALLING SEQUENCE: BRS 39

DESCRIPTION: Returns the job's control parameter word in the A register and the AUNN in the B register.

REGISTERS AFFECTED: A, B

NUMBER: 170

NAME: SBRM

FUNCTION: Executes A BRM Indirect

STATUS: User

CALLING SEQUENCE: SBRM A

DESCRIPTION:

LOC.	INSTR.	ADDRESS
A	ZRO	B

The location of the SBRM instruction is stored in B and control is transferred to location A+1.

REGISTERS AFFECTED: None

NUMBER: 71

NAME: SKXEC

FUNCTION: Skip If System Status Set

STATUS: User

CALLING SEQUENCE: BRS 71

DESCRIPTION: The B register is set to the value of the use code which the user has set for the job. These values are:

Value of B	Use Code
1	Subsystem User
0	User
-1	Both
-2	System

The BRS skips if the B register is negative.

REGISTERS AFFECTED: B

8.0 SYSTEM AND SUBSYSTEM RESTRICTED BRS'S

NUMBER: 46

NAME: NROUT

FUNCTION: Turn Escape Off

STATUS: System

CALLING SEQUENCE: BRS 46

DESCRIPTION: This BRS will set up to remember an escape interrupt, but not allow the program to be interrupted. It will stack the first escape occurring and ignore any subsequent ones.

A program running with escape turned off cannot be terminated by a higher fork.

See also BRS 26 and BRS 47.

REGISTERS AFFECTED: None

NUMBER: 47

NAME: SROUT

FUNCTION: Turn Escape On

STATUS: System

CALLING SEQUENCE: BRS 47

DESCRIPTION: This BRS reverses BRS 46; that is, reactivates the escape interrupt. If an escape interrupt was stacked (remembered) while in an Off condition, the interrupt will occur.

REGISTERS AFFECTED: None

NUMBER: 26

NAME: SKROUT

FUNCTION: Skip If Escape Waiting

STATUS: System

CALLING SEQUENCE:
```
BRS   26
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Checks for a stacked escape for this program and if there is one, transfers control to the "normal return" or to the "exception return" if there is not an escape stacked. Significant only after BRS 46 is used.

REGISTERS AFFECTED: None

NUMBER: 72

NAME: EXDMS

FUNCTION: System Fork Dismissal

STATUS: System

CALLING SEQUENCE:
```
LDB   D
LDX   N
BRS   72
```

N = The number of the queue that the fork is to be put on
D = Dismiss condition

DESCRIPTION: Dismisses a system fork and puts it on the specified queue. Returns to call +1 when reactivated.

0 = Teletype queue
1 = Input/output queue
2 = Short time quantum queue
3 = Long time quantum queue

REGISTERS AFFECTED: None

NUMBER: 22

NAME: NTERM

FUNCTION: Prevents Fork From Terminating On Quantum Overflow

STATUS: System

CALLING SEQUENCE: BRS 22

DESCRIPTION: Prevents the monitor from terminating the calling fork on quantum overflow.

REGISTERS AFFECTED: None

NUMBER: 23

NAME: ALTERM

FUNCTION: Allows Fork To Terminate On Quantum Overflow

STATUS: System

CALLING SEQUENCE: BRS 23

DESCRIPTION: Allows the monitor to terminate the calling fork on quantum overflow. It is used to reset BRS 22.

REGISTERS AFFECTED: None

NUMBER: 109

NAME: DMS

FUNCTION: Dismiss

STATUS: User

CALLING SEQUENCE: BRS 109

DESCRIPTION: The fork is dismissed. It can only be activated again by a program interrupt which has been armed by this fork or the termination of a lower fork.

REGISTERS AFFECTED: None

NUMBER: 111

NAME: BRSRET

FUNCTION: Return From Class 3 BRS

STATUS: System

CALLING SEQUENCE: BRS 111

DESCRIPTION: This BRS is used only by the author of class 3 BRS's. It is the only normal termination of a class 3 BRS. It corresponds to a BRS 10 for other forks.

Instruction Trap:

BRS issued by a fork which was not a class 3 BRS.

REGISTERS AFFECTED: None

NUMBER: 139

NAME: MFSYS

FUNCTION: Gives The Fork System Status

STATUS: Operator Or System

CALLING SEQUENCE:
```
LDA   A
BRS   139
```

A = A constant checked by the BRS

DESCRIPTION: The fork issuing this BRS will run with system status.

REGISTERS AFFECTED: None

NUMBER: 1

NAME: MONOPN

FUNCTION: Open A File Of A Specific Device

STATUS: System

CALLING SEQUENCE:
```
LDA   ±I
LDX   D
BRS   1
EXCEPTION RETURN
NORMAL RETURN
```

File number will be in register A, the index block in register X, and the address of the data area in the buffer in register B on Normal Return.
I = The relative address (Disk Address MOD 4) of the file's Index Block for disk files, or unit number for magnetic tape, otherwise anything.
\- = Make the file read only.
\+ = Make the file read/write.
D = Device number

Available device numbers are as follows:

1. Paper tape input
2. Paper tape output
3. Card input
4. Magnetic tape input
5. Magnetic tape output
8. Sequential disk input
9. Sequential disk output
11. Printer

DESCRIPTION: The "open file" BRS is used to condition a file for input or output processing. If the file is successfully opened, control is transferred to the normal return; otherwise control is transferred to the exception return. If the exception return is taken, the A register contains an indication of the reason:

1) Device or file in use or not available.
2) Too many files open.
3) Bit map not set.
4) No disk space left.

A file may be opened for input any number of times for the purpose of multiple user access or multiple processing by a single user. A file that is opened for output cannot be opened again until it is closed. See also BRS's 2, 20, 82, 17, 148, 147, 8.

REGISTERS AFFECTED: A, X

NUMBER: 2

NAME: MONCLS

FUNCTION: Close A File

STATUS: System

CALLING SEQUENCE:
```
LDA   N
BRS   2
NORMAL RETURN
```

N = File number (obtained when file was opened)

DESCRIPTION: The "close file" BRS is used to indicate to the system all processing is completed on this file. All necessary termination processing will be completed and control will be transferred to the normal return. See also BRS's 1, 8, 17, 20, 147, 148, and 82.

REGISTERS AFFECTED: None

NUMBER: 8

NAME: IOH

FUNCTION: Close All Files

STATUS: System

CALLING SEQUENCE: BRS 8
NORMAL RETURN

DESCRIPTION: The "close all files" BRS is used to indicate to the system all processing is completed on all files. The system will complete all necessary termination processing on all files and transfer control to the normal return. BRS 8 is always executed when control returns to the EXECUTIVE. This BRS will not close magnetic tape files correctly. See also BRS 1, 2, 82, and 17.

REGISTERS AFFECTED: None

NUMBER: 148

NAME: IOH2

FUNCTION: Closes All But A Specified File

STATUS: System

CALLING SEQUENCE:
```
LDA   N
BRS   148
```

N = File number of specified file

DESCRIPTION: Closes all of the user's files except the one specified in A.

REGISTERS AFFECTED: None

NUMBER: 66

NAME: DFDL

FUNCTION: Delete Disk File Data

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   N
BRS   66
NORMAL RETURN
```

N = File number

DESCRIPTION: This BRS will return to available storage all disk blocks which are assigned to the indicated file and clear the index block of disk addresses. The file must be open as an output file.

REGISTERS AFFECTED: None

NUMBER: 67

NAME: DFER

FUNCTION: Delete A Specified Block Of The Disk

STATUS: System

CALLING SEQUENCE:
```
LDA   D
BRS   67
NORMAL RETURN
```

D = Address of the disk block

DESCRIPTION: This BRS will return the disk block indicated by the address in register A to available storage and transfers control to the normal return. This BRS should be used to delete Index Blocks. The BRS does not clear the Index Block address from the Customer File Directory.

REGISTERS AFFECTED: None

NUMBER: 87

NAME: DFRX

FUNCTION: Read Disk File Index Block

STATUS: System

CALLING SEQUENCE:
```
LDA   D
LDX   W
BRS   87
NORMAL RETURN
```

D = Disk address of the index block (MOD 4)
W = Core address into which the block is to be read

DESCRIPTION: Reads the specified block into the given core location and transfers control to the normal return. The block read is the size of the currently defined index block. The size of an index block varies with the assembly.

REGISTERS AFFECTED: None

NUMBER: 124

NAME: ARD

FUNCTION: Read Disk

STATUS: System

CALLING SEQUENCE:
```
LDA   =Core Address
LDB   =Disk Address
LDX   =Number of Words
BRS   124
NORMAL RETURN
```

DESCRIPTION: Reads from the disk as specified. Errors result in an instruction trap, or programmed interrupt 11 if it is armed. Two forks that are to run simultaneously should not both use this BRS. The number of words must be a multiple of 16 and greater than 0. The BRS will not read over page boundaries.

REGISTERS AFFECTED: None

NUMBER: 125

NAME: AWD

FUNCTION: Write Disk

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   =Core Address
LDB   =Disk Address
LDX   =Number of Words
BRS   125
```

DESCRIPTION: Like BRS 124. The number of words must be a multiple of 64 and greater than 0.

REGISTERS AFFECTED: None

NUMBER: 114

NAME: MTDI

FUNCTION: Turn Off Run-Away Magnetic Tape

STATUS: EXEC

CALLING SEQUENCE: BRS 114
NORMAL RETURN

DESCRIPTION: Issues commands to try to stop the tape.

REGISTERS AFFECTED: None

NUMBER: 118

NAME: TGET

FUNCTION: Allocate Magnetic Tape Unit

STATUS: EXEC

CALLING SEQUENCE:
```
LDA    =Tape Number
BRS    118
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Assigns tape requested to the user. If tape is already busy with someone else the exception return is executed.

REGISTERS AFFECTED: None

NUMBER: 119

NAME: TREL

FUNCTION: De-Allocate Magnetic Tape Unit

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   =Tape Number
BRS   119
NORMAL RETURN
```

DESCRIPTION: Releases the tape specified. Releases regardless of who had it.

REGISTERS AFFECTED: None

NUMBER: 172

NAME: CTRL

FUNCTION: Input/Output Control (only tape is implemented)

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   C
CTRL  N
```

C = Control number
N = File number

DESCRIPTION: CTRL provides the following control functions for tape files:

Control #	Description
1.	Write end of record on output. Record count not used.
2.	Backspace physical block.
3.	Forward space physical block.
4.	Backspace file.
5.	Erase tape (output only) (3 inches).
6.	Rewind.
7.	Write EOF. Output only.
8.	Long erase. Output only.

REGISTERS AFFECTED: None

NUMBER: 130

NAME: BPTEST

FUNCTION: Test A Breakpoint Switch

STATUS: Subsystem

CALLING SEQUENCE:
```
LDX   =Switch Number
BRS   130
SWITCH UP RETURN
SWITCH DOWN RETURN
```

DESCRIPTION: Tests the breakpoint switch (1,2,3,4) indicated in X. If the switch is down, the BRS skips on return.

REGISTERS AFFECTED: None

NUMBER: 126

NAME: CARRY

FUNCTION: Test For Carrier Presence

STATUS: System

CALLING SEQUENCE:
```
LDA   =LINE #
BRS   126
EXCEPTION RETURN - No Carrier
NORMAL RETURN - Carrier Present
```

DESCRIPTION: This BRS gives a skip return, if the carrier signal is present on the line indicated in A. No carrier signal - no skip.

REGISTERS AFFECTED: None

NUMBER: 129

NAME: TTYON

FUNCTION: Turns A Teletype Line On Or Off

STATUS: System

CALLING SEQUENCE:
```
LDA   =Teletype #
LDB   =0 (off) or -1 (on)
BRS   129
NORMAL RETURN
```

DESCRIPTION: Issues the EOM and POT commands which cause the line to be turned off (hung up) or made ready to accept an incoming call.

REGISTERS AFFECTED: None

NUMBER: 112

NAME: TSOFF

FUNCTION: Turn Off Teletype Station

STATUS: EXEC

CALLING SEQUENCE:
```
LDA    Job Number
BRS    112
```

DESCRIPTION: This BRS is known as suicide. The job disappears completely from the system.

REGISTERS AFFECTED: All

NUMBER: 152

NAME: IOFI

FUNCTION: Ignore Off-Interrupts

STATUS: Subsystem

CALLING SEQUENCE:
```
LDB   N
BRS   152
```

N = -1 to turn interrupts off
N = 0 to turn interrupts on

DESCRIPTION: Ignores the off interrupts from the user's channel until it is reset.

REGISTERS AFFECTED: None

NUMBER: 120

NAME: APMTE

FUNCTION: Assign PMT Entry

STATUS: System

CALLING SEQUENCE:
```
LDA   R
BRS   120
```

R = Relabeling byte

DESCRIPTION: Obtains a new page for the relabeling byte specified. This BRS is used only in the recover routine in the EXEC.

Instruction Trap:

1) PMT entry is already assigned.
2) The relabeling byte number was not in the PMT.

REGISTERS AFFECTED: None

NUMBER: 3

NAME: PMTI

FUNCTION: Make PMT Pointer Indirect

STATUS: System

CALLING SEQUENCE:
```
LDA   N
LDX   X
BRS   3
```

N = Indirect PMT byte number. Bit 0=1 for read only.
X = Indirect job number.

DESCRIPTION: Allows a user to specify that one of his PMT bytes should be identical to that of another user. The byte must have been previously defined by the other user. Setting bit 0 of the A register allows the user to access the page with read only protection. SMT bytes may also be specified in which case X is ignored.

The BRS returns the number of the PMT byte in the user's memory in the A register.

REGISTERS AFFECTED: A

NUMBER: 96

NAME: ECRECV

FUNCTION: Recover

STATUS: EXEC Only

CALLING SEQUENCE:
```
LDA   N
BRS   96
```

N = File Number

DESCRIPTION: This BRS reads the dump file written by a BRS 95 and recovers the machine status as it appeared at the time the dump was taken. The message "NOT COMPATIBLE" is typed if the dump file was created on an incompatible system.

REGISTERS AFFECTED: All

NUMBER: 127

NAME: PEBRS

FUNCTION: Reads Or Sets One Word In Core

STATUS: System Or EXEC

CALLING SEQUENCE:
```
LDA   V
LDB   =0 or =-1
LDX   =16 bit core address
BRS   127
RETURN
```

V = New value for the word if it is to be set.
The contents of the location are returned in the A register.
If B is positive, the word is read.
If B is negative, the word is changed and the old value returned in A.

DESCRIPTION: Allows a system program to read or set the contents of any location in the memory.

The original contents of the location are always returned in the A register.

REGISTERS AFFECTED: A

NUMBER: 88

NAME: RTEX

FUNCTION: Read Execution Time

STATUS: Subsystem

CALLING SEQUENCE: BRS 88

DESCRIPTION: Returns the execution time for the job in A.

REGISTERS AFFECTED: A

NUMBER: 37

NAME: GSLOOK

FUNCTION: General String Lookup

STATUS: System

CALLING SEQUENCE:
```
LDA   F
LDB   S
LDX   T
BRS   37
EXCEPTION RETURN
NORMAL RETURN
```

F = Input file number
S = Address of string pointer pair
T = Address of the Hash Table Control Table

DESCRIPTION: The hash table is scanned for a string to match the given one. If an exact match is found the normal return is taken (see exits below). If the given string does not match the initial part of any hash table string, the exception return is taken. If the given string matches the initial part of some hash table string, characters from the input file are appended until the string is long enough either to determine a unique hash table string, with a matching initial part, or for no match to be possible, in which case the exception return is taken. In the case where a unique hash table string has been located, more characters are taken from input until an exact match is obtained, in which case the normal return is taken, or until the last character causes a mis-match. If the last character is alphanumeric, the exception return is taken since it is assumed that only a non-alphanumeric character, such as a space, carriage return, punctuation marks, etc., can be considered a proper terminator. In most cases the last character (which caused the mis-match) is left in the input file.

Exits are as follows: (1) The exception return is taken on the no-match condition with a string pointer in A, B to the string so far collected. X is 40000000B if a unique match was found before the bad character; X is zero on no match at all. (2) The normal return is taken on a match with the address of a hash table string pointer in A and the string "value" in B. X is undisturbed.

The "value" is the hash image for the string.

See also Berkeley Document No. 30.10.20 for more details.

REGISTERS AFFECTED: A,B

NUMBER: 128

NAME: SDBM

FUNCTION: Set Disk Bit Map

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   =Address of X Block Mod 4
BRS   128
EXCEPTION RETURN
NORMAL RETURN
```

Exception Return - A contains address that was in conflict

DESCRIPTION: Turns off bits in the disk bit map for the X block and each data block referenced by the index block. If any conflicts occur (the bit is already off), the address is left in the A register and the exception return is taken. A conflict also increments one of two counters, XBERR or FDERR, for errors in the X block or the file directory respectively.

When the bit map has been set, one more call is made to this BRS with A negative. At that time a switch is set allowing output files to be opened; the new overflow pointer is set from B and the accounting area pointer is set from X.

REGISTERS AFFECTED: A

NUMBER: 131

NAME: CRASH

FUNCTION: To Crash The System

STATUS: EXEC

CALLING SEQUENCE: BRS 131
NO RETURN

DESCRIPTION: Saves the registers in SS01, SS02, SS03. Saves 0 in MCRO. Turns off the clock and disables the interrupts. Moves the TS block into real page 7.

REGISTERS AFFECTED: None

NUMBER: 136

NAME: SETSW

FUNCTION: Sets System EXEC Switches In SYMS

STATUS: System

CALLING SEQUENCE:
```
LDA   V
LDX   N
BRS   136
NORMAL RETURN
```

V = New switch value
N = Switch number

DESCRIPTION: The switch is set to the new value and the old value is returned in A.

REGISTERS AFFECTED: A

NUMBER: 141

NAME: EXBRS

FUNCTION: Gets EXEC Subroutines

STATUS: System

CALLING SEQUENCE: BRS 141

DESCRIPTION: Checks that the issuing fork has system status. If it does, an EXEC BRS is issued. This BRS is used to allow system status forks to access subroutines in the EXEC.

REGISTERS AFFECTED: None

NUMBER: 143

NAME: GATHER

FUNCTION: Starts Gathering Of Statistics

STATUS: System

CALLING SEQUENCE:
```
LDX   X
LDB   B
BRS   143
```

X = Address for statistics in user memory
B = 0 to start statistics, and non-zero to stop statistics after the current dump

DESCRIPTION: Copies all statistics since last call to GATHER into user memory starting at location X. The maximum number of accumulated statistics is less than 2048 words.

Instruction trap:

1) If the statistics will not all fit in one user page.
2) If the user page is read-only.
3) If the page is not in user memory.

REGISTERS AFFECTED: None

NUMBER: 61

NAME: SSC

FUNCTION: Starts Clock Interrupt Statistics

STATUS: System In CPARW

CALLING SEQUENCE:
```
LDA   A
LDB   B
LDX   X
BRS   61
```

A,B are constants checked by the BRS.
X is the frequency in clock ticks with which to sample.

DESCRIPTION: Records the program counter and relabeling at every X clock ticks.

REGISTERS AFFECTED: None

NUMBER: 144

NAME: EXBGET

FUNCTION: Gets A Buffer

STATUS: System

CALLING SEQUENCE:
```
BRS   144
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Leaves the address of the data area in the buffer in the A register and takes the normal return.

Takes the exception return if there are no free buffers.

REGISTERS AFFECTED: A

NUMBER: 145

NAME: EXBPUT

FUNCTION: Returns A Buffer

STATUS: System

CALLING SEQUENCE:
```
LDA   A
BRS   145
```

A = Address of data area in buffer to be returned

DESCRIPTION: Returns the buffer to the monitor.

REGISTERS AFFECTED: None

NUMBER: 97

NAME: RESBRS

FUNCTION: Sets Subsystem Counter To 0

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   97
```

A = Counter number (0-31)

DESCRIPTION: Sets the counter specified by A to 0.

REGISTERS AFFECTED: None

NUMBER: 98

NAME: INCBRS

FUNCTION: Increment Subsystem Counter

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   98
```

A = Counter number (0-31)

DESCRIPTION: Increments the counter specified by A.

REGISTERS AFFECTED: None

NUMBER: 99

NAME: REDBRS

FUNCTION: Reads The Subsystem Counter

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   99
```

A = Counter number (0-31)

DESCRIPTION: Returns the value of the counter specified by A in A.

REGISTERS AFFECTED: A

9.0 INDEX OF BRS'S AND SYSTEM OPERATORS

Note: BRS's marked with an asterisk are EXECUTIVE BRS's. All others are monitor BRS's. BRS's and SYSPOP's in Section 8 are restricted to system or subsystem use.

9.1 BRS's

42	7- 2	Read real time clock
43	4- 4	Read pseudo-relabeling
44	4- 5	Set pseudo-relabeling
45	1- 4	Dismiss on quantum overflow
46	8- 2	Turn escape off
47	8- 3	Turn escape on
*48	2- 6	Look up file name
49	1-10	Read interrupts armed
50	6- 8	Floating to fixed conversion
51	6- 9	Fixed to floating conversion
*52	6- 4	Formatted floating point input
*53	6- 5	Formatted floating point output
54		
55		
56		
57		
58		
59		
*60	2- 7	Look up file name and insert if necessary
61	8-40	Start clock statistics
62		
63		
64		
65		
66	8-15	Delete disk file data
67	8-16	Delete a specified disk block
68		
69		
70	4- 9	Count free user pages
71	7- 6	Skip if system status set
72	8- 5	EXEC dismiss
73		
74	3-13	Set 1/2 duplex mode
75	3-14	Set full duplex mode
76	3-15	Skip on 1/2 duplex mode
77		
78	1- 8	Arm/disarm software interrupts
79	1- 9	Cause a program interrupt
80	4- 8	Make page read only
81	1- 5	Dismiss for specified amount of time
82		
83		
84		
85	3-10	Set 8-level output
86	3-11	Clear 8-level output
87	8-17	Read disk file index block
88	8-33	Read execution time
89		
90	1- 7	Declare a fork for termination on "escape"
*91	7- 3	Read date and time to string
92		
93		

94		
*95	2-16	Write a dump file
*96	8-31	Recover
97	8-43	Reset subsystem counter
98	8-44	Increment subsystem counter
99	8-45	Read subsystem counter
100		
101		
102		
103		
104		
105		
106		
107		
108		
109	8- 8	Dismiss
110	2-17	Read device and unit
111	8- 9	Terminate class 3 BRS fork
112	8-27	Turn off teletype station
113	2-18	Compute size of a disk file
114	8-20	Turn off run away magnetic tape
115		
116	4- 6	Read program relabeling
117	4- 7	Set program relabeling
118	8-21	Allocate magnetic tape unit
119	8-22	De-allocate magnetic tape unit
120	8-29	Assign PMT entry
121	4- 3	Release a page of memory
122		
123		
124	8-18	Read disk
125	8-19	Write disk
126	8-25	Test for carrier presence
127	8-32	Look at memory
128	8-35	Set disk bit map
129	8-26	Turn CTE channel ON or OFF
130	8-24	Test breakpoint switch
131	8-36	Crash system
132		
133		
134	3-12	Set to ignore line feed or carriage return
135	1-11	Cause program interrupt after specified period of time
136	8-37	Set EXEC switches
137		
138		
139	8-10	Give fork system status
140		
141	8-38	Get EXEC subroutines
*142	2-20	Get file data and access count
143	8-39	Start gathering statistics
144	8-41	Get a buffer

145	8-42	Return a buffer
146		
147	2-11	Close all except commands from file
148	8-14	Close all but a specified file
149		
150		
*151	2- 8	Open a commands from file
152	8-28	Ignore off interrupts

9.2 System Operators

134	CIT	2-15	Character input and test
135	WCD	5-14	Write character and decrement
136			
137	GCD	5-13	Get character and decrement
140	ISC	6- 7	Internal to string conversion
141	SIC	6- 6	String to internal conversion
142			
143			
144			
145			
146			
147			
150			
151			
152			
153	FDV	6-14	Floating point divide
154	FMP	6-13	Floating point multiply
155	FSB	6-12	Floating point subtract
156	FAD	6-11	Floating point add
157	WCI	5-12	Write character and increment
160	WIO	2-13	Word input/output
161	CIO	2-12	Character input/output
162	SKSG	5-10	Skip if string greater
163	SKSE	5- 9	Skip if string equal
164	WCH	5-15	Write character to memory by table
165	GCI	5-11	Get character and increment
166	LDP	5- 8	Load string pointer
167	STP	5- 7	Store string pointer
170	SBRM	7- 5	System BRM for indirect linkage
171			
172	CTRL	8-23	Input/output control
173	BRS		Branch to system
174	TCI	3- 2	Teletype character input
175	TCO	3- 3	Teletype character output
176	BIO	2-14	Block input/output
177			

APPENDIX A

GLOSSARY

APPENDIX A, GLOSSARY OF TERMS

BREAKPOINT SWITCH

Refers to the four toggle switches physically located on the computer console.

COMMAND FILE

The particular file from which the commands to the System Executive and Subsystems are input. For teletype input the command file number is zero.

CUSTOMER FILE DIRECTORY

The names of all files for a particular user are recorded in this directory.

DEVICE TABLE

Device	Number
Paper Tape Input	1
Paper Tape Output	2
Magnetic Tape Input	4
Magnetic Tape Output	5
Hollerith Card Output	6
Binary Card Output	7
Disk Input	8
Disk Output	9
High Speed Printer Output	11
Hollerith Card Input	12
Binary Card Input	13

DISK BLOCK

Four consecutive sectors on the disk whose beginning addresses are MOD 4. A block consists of 256 words.

DISK DATA BLOCK

A disk block which contains data in the file.

DISK FILE

A file stored on the Disk. Each file consists of at least an Index Block and if the file contains data, then a sufficient number of disk data blocks to record the data.

EXECUTIVE BRS

An Executive BRS is one which starts a fork to execute.

FILE NUMBER

A file number is assigned by the system to files as they are opened. Also, there are fixed file numbers for certain devices. These are as follows:

0	Teletype Input
1	Teletype Output
2	Nothing

FILE TYPE

There are four standard file types. They are as follows:

1 File written by the System Executive as commanded by the "SAVE" command.

2 General Binary File created by a subsystem, i.e., a FORTRAN Object Program.

3 Symbolic File

4 Dump File

FORK

A fork is all or part of a program. A program may consist of one to eight forks and these forks may be in a hierarchy one to another. Forks are different from subroutines in that all forks making up a program could be theoretically executing simultaneously. At least one fork is associated with each active user in the system.

FORK STATES

-2	Dismissed for input/output.
-1	Running.
0	Dismissed on escape key or programmed panic.
1	Dismissed on illegal instruction panic.
2	Dismissed on memory panic.

INDEX BLOCK

A disk block (256 words) which contains the disk addresses for all data blocks of a file. Words 0 through 121 contain a disk address which is MOD 4 in bits 6 to 23. Bits 0 and 5 of these words are unused. Bit 2 indicates an End of Record data block. Words 124 and 125 are link pointers and 126 is a hash total. Word 127 contains the user number.

INSTRUCTION TRAP

A panic caused by attempting to execute an instruction which cannot be executed in the user mode, such as a halt or device input/output instruction or a BRS which is not available to the user.

MEMORY TRAP

A panic caused by a fork attempting to address memory outside its range or write on memory which is set to read only.

PAC TABLE

Each fork is defined by a program active table. This table contains most of the information required to control selection, execution and interruption of the fork (additional information is stored in the user's TS page).

PAGE

A page can exist on RAD, disk or in core memory but in all cases refers to 2048 words.

PANIC

A panic is a signal to the system to break execution of a fork.

PANIC TABLE

Word
0 = Program Counter
1 = A Register
2 = B Register
3 = X Register
4 = First Relabeling Register
5 = Second Relabeling Register
6 = Status

The status word may be:

-2 Dismissed for Input/Output
-1 Running
0 Dismissed on Escape or BRS 10
1 Dismissed on Illegal Instruction Trap
2 Dismissed on Memory Trap

A Panic Table must not overlap a page boundary.

QUANTUM, LONG TIME

The maximum length of time a fork can run before the scheduler checks for other forks to be run.

QUANTUM, SHORT TIME

The minimum length of time a fork will run before the scheduler checks for other forks to be run which were dismissed for input/output.

RELABELING, PSEUDO

See Relabeling Registers

RELABELING REGISTERS

The relabeling registers are used to indicate a page number which has been assigned to a user for a particular logical page. They are of the form:

First Word	Page 0	Page 1	Page 2	Page 3
Second Word	Page 4	Page 5	Page 6	Page 7

STRING POINTERS

A pair of pointers which contain a character address of the character before the first character of a string and a character address of the last character of the string.

STRING, NULL

A pair of string pointers whose character addresses are the same.

8.0 SYSTEM AND SUBSYSTEM RESTRICTED BRS'S

NUMBER: 46

NAME: NROUT

FUNCTION: Turn Escape Off

STATUS: System

CALLING SEQUENCE: BRS 46

DESCRIPTION: This BRS will set up to remember an escape interrupt, but not allow the program to be interrupted. It will stack the first escape occurring and ignore any subsequent ones.

A program running with escape turned off cannot be terminated by a higher fork.

See also BRS 26 and BRS 47.

REGISTERS AFFECTED: None

NUMBER: 47

NAME: SROUT

FUNCTION: Turn Escape On

STATUS: System

CALLING SEQUENCE: BRS 47

DESCRIPTION: This BRS reverses BRS 46; that is, reactivates the escape interrupt. If an escape interrupt was stacked (remembered) while in an Off condition, the interrupt will occur.

REGISTERS AFFECTED: None

NUMBER: 26

NAME: SKROUT

FUNCTION: Skip If Escape Waiting

STATUS: System

CALLING SEQUENCE:
```
BRS   26
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Checks for a stacked escape for this program and if there is one, transfers control to the "normal return" or to the "exception return" if there is not an escape stacked. Significant only after BRS 46 is used.

REGISTERS AFFECTED: None

NUMBER: 72

NAME: EXDMS

FUNCTION: System Fork Dismissal

STATUS: System

CALLING SEQUENCE:
```
LDB   D
LDX   N
BRS   72
```

N = The number of the queue that the fork is to be put on
D = Dismiss condition

DESCRIPTION: Dismisses a system fork and puts it on the specified queue. Returns to call +1 when reactivated.

0 = Teletype queue
1 = Input/output queue
2 = Short time quantum queue
3 = Long time quantum queue

REGISTERS AFFECTED: None

NUMBER: 22

NAME: NTERM

FUNCTION: Prevents Fork From Terminating On Quantum Overflow

STATUS: System

CALLING SEQUENCE: BRS 22

DESCRIPTION: Prevents the monitor from terminating the calling fork on quantum overflow.

REGISTERS AFFECTED: None

NUMBER: 23

NAME: ALTERM

FUNCTION: Allows Fork To Terminate On Quantum Overflow

STATUS: System

CALLING SEQUENCE: BRS 23

DESCRIPTION: Allows the monitor to terminate the calling fork on quantum overflow. It is used to reset BRS 22.

REGISTERS AFFECTED: None

NUMBER: 109

NAME: DMS

FUNCTION: Dismiss

STATUS: User

CALLING SEQUENCE: BRS 109

DESCRIPTION: The fork is dismissed. It can only be activated again by a program interrupt which has been armed by this fork or the termination of a lower fork.

REGISTERS AFFECTED: None

NUMBER: 111

NAME: BRSRET

FUNCTION: Return From Class 3 BRS

STATUS: System

CALLING SEQUENCE: BRS 111

DESCRIPTION: This BRS is used only by the author of class 3 BRS's. It is the only normal termination of a class 3 BRS. It corresponds to a BRS 10 for other forks.

Instruction Trap:

BRS issued by a fork which was not a class 3 BRS.

REGISTERS AFFECTED: None

NUMBER: 139

NAME: MFSYS

FUNCTION: Gives The Fork System Status

STATUS: Operator Or System

CALLING SEQUENCE:
```
LDA   A
BRS   139
```

A = A constant checked by the BRS

DESCRIPTION: The fork issuing this BRS will run with system status.

REGISTERS AFFECTED: None

NUMBER: 1

NAME: MONOPN

FUNCTION: Open A File Of A Specific Device

STATUS: System

CALLING SEQUENCE:
```
LDA   ±I
LDX   D
BRS   1
EXCEPTION RETURN
NORMAL RETURN
```

File number will be in register A, the index block in register X, and the address of the data area in the buffer in register B on Normal Return.
I = The relative address (Disk Address MOD 4) of the file's Index Block for disk files, or unit number for magnetic tape, otherwise anything.
\- = Make the file read only.
\+ = Make the file read/write.
D = Device number

Available device numbers are as follows:

1. Paper tape input
2. Paper tape output
3. Card input
4. Magnetic tape input
5. Magnetic tape output
8. Sequential disk input
9. Sequential disk output
11. Printer

DESCRIPTION: The "open file" BRS is used to condition a file for input or output processing. If the file is successfully opened, control is transferred to the normal return; otherwise control is transferred to the exception return. If the exception return is taken, the A register contains an indication of the reason:

1) Device or file in use or not available.
2) Too many files open.
3) Bit map not set.
4) No disk space left.

A file may be opened for input any number of times for the purpose of multiple user access or multiple processing by a single user. A file that is opened for output cannot be opened again until it is closed. See also BRS's 2, 20, 82, 17, 148, 147, 8.

REGISTERS AFFECTED: A, X

NUMBER: 2

NAME: MONCLS

FUNCTION: Close A File

STATUS: System

CALLING SEQUENCE:
```
LDA   N
BRS   2
NORMAL RETURN
```

N = File number (obtained when file was opened)

DESCRIPTION: The "close file" BRS is used to indicate to the system all processing is completed on this file. All necessary termination processing will be completed and control will be transferred to the normal return. See also BRS's 1, 8, 17, 20, 147, 148, and 82.

REGISTERS AFFECTED: None

NUMBER: 8

NAME: IOH

FUNCTION: Close All Files

STATUS: System

CALLING SEQUENCE: BRS 8
NORMAL RETURN

DESCRIPTION: The "close all files" BRS is used to indicate to the system all processing is completed on all files. The system will complete all necessary termination processing on all files and transfer control to the normal return. BRS 8 is always executed when control returns to the EXECUTIVE. This BRS will not close magnetic tape files correctly. See also BRS 1, 2, 82, and 17.

REGISTERS AFFECTED: None

NUMBER: 148

NAME: IOH2

FUNCTION: Closes All But A Specified File

STATUS: System

CALLING SEQUENCE:
```
LDA   N
BRS   148
```

N = File number of specified file

DESCRIPTION: Closes all of the user's files except the one specified in A.

REGISTERS AFFECTED: None

NUMBER: 66

NAME: DFDL

FUNCTION: Delete Disk File Data

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   N
BRS   66
NORMAL RETURN
```

N = File number

DESCRIPTION: This BRS will return to available storage all disk blocks which are assigned to the indicated file and clear the index block of disk addresses. The file must be open as an output file.

REGISTERS AFFECTED: None

NUMBER: 67

NAME: DFER

FUNCTION: Delete A Specified Block Of The Disk

STATUS: System

CALLING SEQUENCE:
```
LDA   D
BRS   67
NORMAL RETURN
```

D = Address of the disk block

DESCRIPTION: This BRS will return the disk block indicated by the address in register A to available storage and transfers control to the normal return. This BRS should be used to delete Index Blocks. The BRS does not clear the Index Block address from the Customer File Directory.

REGISTERS AFFECTED: None

NUMBER: 87

NAME: DFRX

FUNCTION: Read Disk File Index Block

STATUS: System

CALLING SEQUENCE:
```
LDA   D
LDX   W
BRS   87
NORMAL RETURN
```

D = Disk address of the index block (MOD 4)
W = Core address into which the block is to be read

DESCRIPTION: Reads the specified block into the given core location and transfers control to the normal return. The block read is the size of the currently defined index block. The size of an index block varies with the assembly.

REGISTERS AFFECTED: None

NUMBER: 124

NAME: ARD

FUNCTION: Read Disk

STATUS: System

CALLING SEQUENCE:
```
LDA   =Core Address
LDB   =Disk Address
LDX   =Number of Words
BRS   124
NORMAL RETURN
```

DESCRIPTION: Reads from the disk as specified. Errors result in an instruction trap, or programmed interrupt 11 if it is armed. Two forks that are to run simultaneously should not both use this BRS. The number of words must be a multiple of 16 and greater than 0. The BRS will not read over page boundaries.

REGISTERS AFFECTED: None

NUMBER: 125

NAME: AWD

FUNCTION: Write Disk

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   =Core Address
LDB   =Disk Address
LDX   =Number of Words
BRS   125
```

DESCRIPTION: Like BRS 124. The number of words must be a multiple of 64 and greater than 0.

REGISTERS AFFECTED: None

NUMBER: 114

NAME: MTDI

FUNCTION: Turn Off Run-Away Magnetic Tape

STATUS: EXEC

CALLING SEQUENCE: BRS 114
NORMAL RETURN

DESCRIPTION: Issues commands to try to stop the tape.

REGISTERS AFFECTED: None

NUMBER: 118

NAME: TGET

FUNCTION: Allocate Magnetic Tape Unit

STATUS: EXEC

CALLING SEQUENCE:
```
LDA    =Tape Number
BRS    118
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Assigns tape requested to the user. If tape is already busy with someone else the exception return is executed.

REGISTERS AFFECTED: None

NUMBER: 119

NAME: TREL

FUNCTION: De-Allocate Magnetic Tape Unit

STATUS: EXEC

CALLING SEQUENCE:
```
LDA    =Tape Number
BRS    119
NORMAL RETURN
```

DESCRIPTION: Releases the tape specified. Releases regardless of who had it.

REGISTERS AFFECTED: None

NUMBER: 172

NAME: CTRL

FUNCTION: Input/Output Control (only tape is implemented)

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   C
CTRL  N
```

C = Control number
N = File number

DESCRIPTION: CTRL provides the following control functions for tape files:

Control #	Description
1.	Write end of record on output. Record count not used.
2.	Backspace physical block.
3.	Forward space physical block.
4.	Backspace file.
5.	Erase tape (output only) (3 inches).
6.	Rewind.
7.	Write EOF. Output only.
8.	Long erase. Output only.

REGISTERS AFFECTED: None

NUMBER: 130

NAME: BPTEST

FUNCTION: Test A Breakpoint Switch

STATUS: Subsystem

CALLING SEQUENCE:
```
LDX   =Switch Number
BRS   130
SWITCH UP RETURN
SWITCH DOWN RETURN
```

DESCRIPTION: Tests the breakpoint switch (1,2,3,4) indicated in X. If the switch is down, the BRS skips on return.

REGISTERS AFFECTED: None

NUMBER: 126

NAME: CARRY

FUNCTION: Test For Carrier Presence

STATUS: System

CALLING SEQUENCE:
```
LDA   =LINE #
BRS   126
EXCEPTION RETURN - No Carrier
NORMAL RETURN - Carrier Present
```

DESCRIPTION: This BRS gives a skip return, if the carrier signal is present on the line indicated in A. No carrier signal - no skip.

REGISTERS AFFECTED: None

NUMBER: 129

NAME: TTYON

FUNCTION: Turns A Teletype Line On Or Off

STATUS: System

CALLING SEQUENCE:
```
LDA   =Teletype #
LDB   =0 (off) or -1 (on)
BRS   129
NORMAL RETURN
```

DESCRIPTION: Issues the EOM and POT commands which cause the line to be turned off (hung up) or made ready to accept an incoming call.

REGISTERS AFFECTED: None

NUMBER: 112

NAME: TSOFF

FUNCTION: Turn Off Teletype Station

STATUS: EXEC

CALLING SEQUENCE:
```
LDA    Job Number
BRS    112
```

DESCRIPTION: This BRS is known as suicide. The job disappears completely from the system.

REGISTERS AFFECTED: All

NUMBER: 152

NAME: IOFI

FUNCTION: Ignore Off-Interrupts

STATUS: Subsystem

CALLING SEQUENCE:
```
LDB   N
BRS   152
```

N = -1 to turn interrupts off
N = 0 to turn interrupts on

DESCRIPTION: Ignores the off interrupts from the user's channel until it is reset.

REGISTERS AFFECTED: None

NUMBER: 120

NAME: APMTE

FUNCTION: Assign PMT Entry

STATUS: System

CALLING SEQUENCE:
```
LDA   R
BRS   120
```

R = Relabeling byte

DESCRIPTION: Obtains a new page for the relabeling byte specified. This BRS is used only in the recover routine in the EXEC.

Instruction Trap:

1) PMT entry is already assigned.
2) The relabeling byte number was not in the PMT.

REGISTERS AFFECTED: None

NUMBER: 3

NAME: PMTI

FUNCTION: Make PMT Pointer Indirect

STATUS: System

CALLING SEQUENCE:
```
LDA   N
LDX   X
BRS   3
```

N = Indirect PMT byte number. Bit 0=1 for read only.
X = Indirect job number.

DESCRIPTION: Allows a user to specify that one of his PMT bytes should be identical to that of another user. The byte must have been previously defined by the other user. Setting bit 0 of the A register allows the user to access the page with read only protection. SMT bytes may also be specified in which case X is ignored.

The BRS returns the number of the PMT byte in the user's memory in the A register.

REGISTERS AFFECTED: A

NUMBER: 96

NAME: ECRECV

FUNCTION: Recover

STATUS: EXEC Only

CALLING SEQUENCE:
```
LDA   N
BRS   96
```

N = File Number

DESCRIPTION: This BRS reads the dump file written by a BRS 95 and recovers the machine status as it appeared at the time the dump was taken. The message "NOT COMPATIBLE" is typed if the dump file was created on an incompatible system.

REGISTERS AFFECTED: All

NUMBER: 127

NAME: PEBRS

FUNCTION: Reads Or Sets One Word In Core

STATUS: System Or EXEC

CALLING SEQUENCE:
```
LDA   V
LDB   =0 or =-1
LDX   =16 bit core address
BRS   127
RETURN
```

V = New value for the word if it is to be set.
The contents of the location are returned in the A register.
If B is positive, the word is read.
If B is negative, the word is changed and the old value returned in A.

DESCRIPTION: Allows a system program to read or set the contents of any location in the memory.

The original contents of the location are always returned in the A register.

REGISTERS AFFECTED: A

NUMBER: 88

NAME: RTEX

FUNCTION: Read Execution Time

STATUS: Subsystem

CALLING SEQUENCE: BRS 88

DESCRIPTION: Returns the execution time for the job in A.

REGISTERS AFFECTED: A

NUMBER: 37

NAME: GSLOOK

FUNCTION: General String Lookup

STATUS: System

CALLING SEQUENCE:
```
LDA   F
LDB   S
LDX   T
BRS   37
EXCEPTION RETURN
NORMAL RETURN
```

F = Input file number
S = Address of string pointer pair
T = Address of the Hash Table Control Table

DESCRIPTION: The hash table is scanned for a string to match the given one. If an exact match is found the normal return is taken (see exits below). If the given string does not match the initial part of any hash table string, the exception return is taken. If the given string matches the initial part of some hash table string, characters from the input file are appended until the string is long enough either to determine a unique hash table string, with a matching initial part, or for no match to be possible, in which case the exception return is taken. In the case where a unique hash table string has been located, more characters are taken from input until an exact match is obtained, in which case the normal return is taken, or until the last character causes a mis-match. If the last character is alphanumeric, the exception return is taken since it is assumed that only a non-alphanumeric character, such as a space, carriage return, punctuation marks, etc., can be considered a proper terminator. In most cases the last character (which caused the mis-match) is left in the input file.

Exits are as follows: (1) The exception return is taken on the no-match condition with a string pointer in A, B to the string so far collected. X is 40000000B if a unique match was found before the bad character; X is zero on no match at all. (2) The normal return is taken on a match with the address of a hash table string pointer in A and the string "value" in B. X is undisturbed.

The "value" is the hash image for the string.

See also Berkeley Document No. 30.10.20 for more details.

REGISTERS AFFECTED: A,B

NUMBER: 128

NAME: SDBM

FUNCTION: Set Disk Bit Map

STATUS: EXEC

CALLING SEQUENCE:
```
LDA   =Address of X Block Mod 4
BRS   128
EXCEPTION RETURN
NORMAL RETURN
```

Exception Return - A contains address that was in conflict

DESCRIPTION: Turns off bits in the disk bit map for the X block and each data block referenced by the index block. If any conflicts occur (the bit is already off), the address is left in the A register and the exception return is taken. A conflict also increments one of two counters, XBERR or FDERR, for errors in the X block or the file directory respectively.

When the bit map has been set, one more call is made to this BRS with A negative. At that time a switch is set allowing output files to be opened; the new overflow pointer is set from B and the accounting area pointer is set from X.

REGISTERS AFFECTED: A

NUMBER: 131

NAME: CRASH

FUNCTION: To Crash The System

STATUS: EXEC

CALLING SEQUENCE: BRS 131
NO RETURN

DESCRIPTION: Saves the registers in SS01, SS02, SS03. Saves 0 in MCRO. Turns off the clock and disables the interrupts. Moves the TS block into real page 7.

REGISTERS AFFECTED: None

NUMBER: 136

NAME: SETSW

FUNCTION: Sets System EXEC Switches In SYMS

STATUS: System

CALLING SEQUENCE:
```
LDA   V
LDX   N
BRS   136
NORMAL RETURN
```

V = New switch value
N = Switch number

DESCRIPTION: The switch is set to the new value and the old value is returned in A.

REGISTERS AFFECTED: A

NUMBER: 141

NAME: EXBRS

FUNCTION: Gets EXEC Subroutines

STATUS: System

CALLING SEQUENCE: BRS 141

DESCRIPTION: Checks that the issuing fork has system status. If it does, an EXEC BRS is issued. This BRS is used to allow system status forks to access subroutines in the EXEC.

REGISTERS AFFECTED: None

NUMBER: 143

NAME: GATHER

FUNCTION: Starts Gathering Of Statistics

STATUS: System

CALLING SEQUENCE:
```
LDX   X
LDB   B
BRS   143
```

X = Address for statistics in user memory
B = 0 to start statistics, and non-zero to stop statistics after the current dump

DESCRIPTION: Copies all statistics since last call to GATHER into user memory starting at location X. The maximum number of accumulated statistics is less than 2048 words.

Instruction trap:

1) If the statistics will not all fit in one user page.
2) If the user page is read-only.
3) If the page is not in user memory.

REGISTERS AFFECTED: None

NUMBER: 61

NAME: SSC

FUNCTION: Starts Clock Interrupt Statistics

STATUS: System In CPARW

CALLING SEQUENCE:
```
LDA   A
LDB   B
LDX   X
BRS   61
```

A,B are constants checked by the BRS.
X is the frequency in clock ticks with which to sample.

DESCRIPTION: Records the program counter and relabeling at every X clock ticks.

REGISTERS AFFECTED: None

NUMBER: 144

NAME: EXBGET

FUNCTION: Gets A Buffer

STATUS: System

CALLING SEQUENCE:
```
BRS   144
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Leaves the address of the data area in the buffer in the A register and takes the normal return.

Takes the exception return if there are no free buffers.

REGISTERS AFFECTED: A

NUMBER: 145

NAME: EXBPUT

FUNCTION: Returns A Buffer

STATUS: System

CALLING SEQUENCE:
```
LDA   A
BRS   145
```

A = Address of data area in buffer to be returned

DESCRIPTION: Returns the buffer to the monitor.

REGISTERS AFFECTED: None

NUMBER: 97

NAME: RESBRS

FUNCTION: Sets Subsystem Counter To 0

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   97
```

A = Counter number (0-31)

DESCRIPTION: Sets the counter specified by A to 0.

REGISTERS AFFECTED: None

NUMBER: 98

NAME: INCBRS

FUNCTION: Increment Subsystem Counter

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   98
```

A = Counter number (0-31)

DESCRIPTION: Increments the counter specified by A.

REGISTERS AFFECTED: None

NUMBER: 99

NAME: REDBRS

FUNCTION: Reads The Subsystem Counter

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   A
BRS   99
```

A = Counter number (0-31)

DESCRIPTION: Returns the value of the counter specified by A in A.

REGISTERS AFFECTED: A