



**PRODUCT SPECIFICATION**

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	7/16/80	<i>J. Hall</i>	Original Issue -- Mark 9.0 Release.
B	9/05/80	<i>J. Hall</i>	Major Revision--- Mark 10.0 Release.

"THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO BURROUGHS CORPORATION AND IS NOT TO BE DISCLOSED TO ANYONE OUTSIDE OF BURROUGHS CORPORATION WITHOUT THE PRIOR WRITTEN RELEASE FROM THE PATENT DIVISION OF BURROUGHS CORPORATION"

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

## TABLE OF CONTENTS

BASE-LIMIT MEMORY LAYOUT . . . . .	1-1
INSTRUCTION SET	2-1
ALPHABETICAL LISTING . . . . .	2-1
LISTING BY NUMERIC OP-CODE	2-5
ARITHMETIC REPLACEMENT S-OPS . . . . .	2-5
LOGICAL REPLACEMENT AND IF STATEMENT S-OPS	2-7
BRANCH S-OPS . . . . .	2-7
TYPE AND SIGN CONVERSION S-OPS	2-7
SUBSCRIPT VALUE COMPUTATION S-OPS . . . . .	2-7
DO LOOP MAINTENANCE	2-8
CHARACTER TYPE S-OPS . . . . .	2-8
SUBROUTINE LINKAGE S-OPS	2-8
SPECIAL FUNCTION S-OPS . . . . .	2-8
PRIVILEGED USER S-OPS	2-9
TRIGONOMETRIC AND OTHER FUNCTIONS . . . . .	2-9
FORMATS	3-1
REGISTERS . . . . .	3-1
ERROR CONCTION INFORMATION	3-1
VALUES . . . . .	3-2
LOCAL DATA BLOCK	3-2
SUBROUTINE LINKAGE MECHANISM . . . . .	3-3
LAYOUT TABLE	3-3
TRANSFER VECTOR . . . . .	3-4
ASSIGNED GOTO AND FORMAT TABLE	3-5
STANDARD INDEX . . . . .	3-7
ADDRESSES	3-7
STANDARD SOURCE . . . . .	3-10
STANDARD DESTINATION	3-10
STANDARD CHARACTER SOURCE . . . . .	3-11
STANDARD CHARACTER DESTINATION	3-12
RUN-TIME DIMENSION TABLE . . . . .	3-12
ARITHMETIC REPLACEMENT S-OPS	4-1
LOGICAL REPLACEMENT AND IF STATEMENT S-OPS . . . . .	5-1
BRANCH S-OPS	6-1
TYPE AND SIGN CONVERSION S-OPS . . . . .	7-1
SUBSCRIPT VALUE COMPUTATION S-OPS	8-1
DO-LOOP MAINTENANCE . . . . .	9-1
CHARACTER TYPE S-OPS	10-1
SUBROUTINE LINKAGE S-OPS . . . . .	11-1
SPECIAL FUNCTION S-OPS	12-1
PRIVILEGED USER S-OPS . . . . .	13-1
TRIGONOMETRIC AND OTHER FUNCTIONS	14-1

BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
P.S. 2222 2749 (B)

### BASE-LIMIT MEMORY LAYOUT

#### STATIC MEMORY CONTAINS :

- 1) INTRINSIC COMMON BLOCKS, CONSISTING OF:
  - A. OVERFLOW/DIVIDE-BY-ZERO MASK,
  - B. STATEMENT NUMBER,
  - C. DEBUG INTERFACE TABLE ADDRESS,
  - D. ENVIRONMENT NUCLEUS BLOCK ADDRESS,
  - E. SEGMENT DICTIONARY ADDRESS,
  - F. DATA DICTIONARY ADDRESS,
  - G. CODE SEGMENT NUMBER,
  - H. I/O BUFFER,
  - I. OTHER I/O POINTERS AND VARIABLES KNOWN TO THE INTERPRETER,
  - J. OTHER COMMON INTRINSIC VARIABLES.
  
- 2) LOCAL DATA BLOCKS, CONSISTING OF:
  - A. RETURN ADDRESS,
  - B. DESCRIPTORS OF DUMMY ARGUMENTS,
  - C. OTHER LOCAL DATA, EXCEPT PAGED ARRAYS.
  
- 3) OTHER COMMON BLOCKS AS DECLARED, EXCEPT PAGED BLOCKS.

DYNAMIC MEMORY CONTAINS PAGED LOCAL ARRAYS AND/OR EQUIVALENCE GROUPS AND PAGED COMMON BLOCKS. THE PLACEMENT OF THESE IN DYNAMIC MEMORY IS DETERMINED BY SIZE, WITH DATA PAGES BEING BYTE ADDRESSABLE, A DATA ITEM MAY BEGIN AT ANY BYTE ADDRESS. A NUMERIC ITEM WILL OCCUPY 4 OR 8 BYTES OF MEMORY. THE SIZE OF MOST DATA PAGES IS 1024 (2\*\*10) BYTES, EXCEPT FOR THE LAST PAGE OF A DATA BLOCK. A DATA BLOCK CONSISTS OF CONSECUTIVE DATA PAGES, AS DESCRIBED BY THE DATA DICTIONARY.

A CODE SEGMENT CONTAINS A TRANSFER VECTOR OF VARIABLE LENGTH FOLLOWED BY EXECUTABLE CODE CONSISTING OF S-OPS. SEE BELOW FOR FORMAT OF A TRANSFER VECTOR.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

## INSTRUCTION SET

### ALPHABETICAL LISTING

ABS	68	(ABSOLUTIZE)
ACOS	9C	(ARCCOSINE)
ADDR	6A	(BASE RELATIVE ADDRESS)
AGO	67	(ASSIGNED GOTO)
AIF	65	(ARITHMETIC IF)
AINT	9E	(FLOOR)
ALOG	9F	(NATURAL LOG)
ALCG10	AJ	(LOG TO BASE 10)
AMOD	94	(REMAINDER)
ASIN	9B	(ARCSINE)
ATAN	9D	(ARCTANGENT)
BAT	8B	(BUILD ARRAY TABLE)
BAAT	8C	(BUILD ASSUMED SIZE ARRAY TABLE)
ENRY	81	(GET BINARY INPUT)
BUMP	A3	(BUMP)
CALL	70	(SUBROUTINE CALL)
CAT	7B	(CHARACTER CONCATENATION)
CATD	79	(CHARACTER CONCATENATION WITH DESCRIPTOR)
CGO	66	(COMPUTED GOTO)
COMM	7C	(COMMUNICATE)
COS	96	(COSINE)
COSH	99	(HYPERBOLIC COSINE)
CREL	5E	(CHARACTER RELATION)
CRIF	5F	(CHARACTER RELATIONAL IF)
CS	8D	(COMPUTE SUBSCRIPT VALUE)
CSB	8E	(CS AND CHECK BOUNDS)
CSV	8F	(CS WITH ARRAY TABLE, ALSO CHECKS BOUNDS)
CADD	50	(DOUBLE ADD)
DBL	57	(DOUBLE PRECISION)
DCAL	71	(DYNAMIC SUBROUTINE CALL)
DDIV	53	(DOUBLE DIVIDE)
DESC	6B	(MAKE DESCRIPTOR)
DMOVE	02	(MOVE DOUBLE WORD)
DMUL	52	(DOUBLE MULTIPLY)
DO-UP	90	(DO LOOP UPDATE)
DREL	5C	(DOUBLE RELATION)
DRIF	5D	(DOUBLE RELATIONAL IF)
DS	7F	(DISCONTINUE JOB)
DSUB	51	(DOUBLE SUBTRACT)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

EXP	A2	(EXPONENTIAL)
FANC	87	(FETCH AND CLEAR ERROR CONDITION)
FADD	0C	(REAL ADD)
FAMMM	37	(REAL ADD - MEM, MEM, MEM)
FAMMR	36	(REAL ADD - MEM, MEM, REG)
FAMRM	35	(REAL ADD - MEM, REG, MEM)
FAMRR	34	(REAL ADD - MEM, REG, REG)
FARMM	33	(REAL ADD - REG, MEM, MEM)
FARMR	32	(REAL ADD - REG, MEM, REG)
FARRM	31	(REAL ADD - REG, REG, MEM)
FARRR	30	(REAL ADD - REG, REG, REG)
FDIV	0F	(REAL DIVIDE)
FDMMM	4F	(REAL DIVIDE - MEM, MEM, MEM)
FDMMR	4E	(REAL DIVIDE - MEM, MEM, REG)
FDMRM	4D	(REAL DIVIDE - MEM, REG, MEM)
FDMRR	4C	(REAL DIVIDE - MEM, REG, REG)
FDRMM	4B	(REAL DIVIDE - REG, MEM, MEM)
FDRMR	4A	(REAL DIVIDE - REG, MEM, REG)
FDRRM	49	(REAL DIVIDE - REG, REG, MEM)
FDRRR	48	(REAL DIVIDE - REG, REG, REG)
FLOAT	54	(CONVERT FROM INTEGER TO FLOATING)
FMMMM	47	(REAL MULTIPLY - MEM, MEM, MEM)
FMMMR	46	(REAL MULTIPLY - MEM, MEM, REG)
FMMRM	45	(REAL MULTIPLY - MEM, REG, MEM)
FMMRR	44	(REAL MULTIPLY - MEM, REG, REG)
FMRMM	43	(REAL MULTIPLY - REG, MEM, MEM)
FMRMR	42	(REAL MULTIPLY - REG, MEM, REG)
FMRRM	41	(REAL MULTIPLY - REG, REG, MEM)
FMRRR	40	(REAL MULTIPLY - REG, REG, REG)
FMUL	0E	(REAL MULTIPLY)
FREL	5A	(REAL RELATION)
FRIF	5B	(REAL RELATIONAL IF)
FSPMM	3F	(REAL SUBTRACT - MEM, MEM, MEM)
FSPMR	3E	(REAL SUBTRACT - MEM, MEM, REG)
FSPRM	3D	(REAL SUBTRACT - MEM, REG, MEM)
FSPRR	3C	(REAL SUBTRACT - MEM, REG, REG)
FSRMM	3B	(REAL SUBTRACT - REG, MEM, MEM)
FSRMR	3A	(REAL SUBTRACT - REG, MEM, REG)
FSRRM	39	(REAL SUBTRACT - REG, REG, MEM)
FSRRR	38	(REAL SUBTRACT - REG, REG, REG)
FSUB	0D	(REAL SUBTRACT)
GOTO	64	(UNCONDITIONAL BRANCH)
HMON	92	(HARDWARE MONITOR)
IADD	08	(INTEGER ADD)
IAMMM	17	(INTEGER ADD - MEM, MEM, MEM)
IAMMR	16	(INTEGER ADD - MEM, MEM, REG)
IAMRM	15	(INTEGER ADD - MEM, REG, MEM)
IAMRR	14	(INTEGER ADD - MEM, REG, REG)
IARMM	13	(INTEGER ADD - REG, MEM, MEM)

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

IARMR	12	(INTEGER ADD - REG, MEM, REG)
IARRM	11	(INTEGER ADD - REG, REG, MEM)
IARRR	10	(INTEGER ADD - REG, REG, REG)
IDIV	0B	(INTEGER DIVIDE)
IDMMM	2F	(INTEGER DIVIDE - MEM, MEM, MEM)
IDMMR	2E	(INTEGER DIVIDE - MEM, MEM, REG)
IDMRM	2D	(INTEGER DIVIDE - MEM, REG, MEM)
IDMRR	2C	(INTEGER DIVIDE - MEM, REG, REG)
IDRMM	2B	(INTEGER DIVIDE - REG, MEM, MEM)
IDRMR	2A	(INTEGER DIVIDE - REG, MEM, REG)
IDRRM	29	(INTEGER DIVIDE - REG, REG, MEM)
IDRRR	28	(INTEGER DIVIDE - REG, REG, REG)
IFIX	55	(CONVERT FROM FLOATING TO INTEGER)
IMMMM	27	(INTEGER MULTIPLY - MEM, MEM, MEM)
IMMMR	26	(INTEGER MULTIPLY - MEM, MEM, REG)
IMMRM	25	(INTEGER MULTIPLY - MEM, REG, MEM)
IMMRR	24	(INTEGER MULTIPLY - MEM, REG, REG)
IMRMM	23	(INTEGER MULTIPLY - REG, MEM, MEM)
IMRMR	22	(INTEGER MULTIPLY - REG, MEM, REG)
IMRRM	21	(INTEGER MULTIPLY - REG, REG, MEM)
IMRRR	20	(INTEGER MULTIPLY - REG, REG, REG)
IMUL	0A	(INTEGER MULTIPLY)
INSERT	89	(INSERT BITS)
IREL	58	(INTEGER RELATION)
IRIF	59	(INTEGER RELATIONAL IF)
ISMMM	1F	(INTEGER SUBTRACT - MEM, MEM, MEM)
ISMMR	1E	(INTEGER SUBTRACT - MEM, MEM, REG)
ISMRM	1D	(INTEGER SUBTRACT - MEM, REG, MEM)
ISMRR	1C	(INTEGER SUBTRACT - MEM, REG, REG)
ISRMM	1B	(INTEGER SUBTRACT - REG, MEM, MEM)
ISRMR	1A	(INTEGER SUBTRACT - REG, MEM, REG)
ISRRM	19	(INTEGER SUBTRACT - REG, REG, MEM)
ISRRR	18	(INTEGER SUBTRACT - REG, REG, REG)
ISUB	09	(INTEGER SUBTRACT)
LB	6E	(LOCAL BASE)
LCR	7D	(LOAD COMMUNICATE REPLY)
LEN	6C	(CHARACTER LENGTH)
LIF1	63	(LOGICAL IF, 1-OPERAND)
LIF2	61	(LOGICAL IF, 2-OPERANDS)
LNOT	62	(LOGICAL NOT)
LOAD	06	(LOAD REGISTER)
LOG	60	(LOGICAL RELATION)
MMOVE	07	(MOVE MEMORY)
MOVE	01	(MOVE SINGLE WORD)
MVC	7A	(MOVE CHARACTER)
NEG	69	(CHANGE SIGN)
NEXT	84	(EXAMINE NEXT CHARACTER)
PASS	6F	(PASS DESCRIPTOR)
REAL	83	(GET REAL VALUE)

EURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
P.S. 2222 2749 (B)

RMV	04	(MOVE REGISTER)
RTN	72	(SUBROUTINE RETURN)
SAVE	60	(SAVE REGISTERS)
SFCL	73	(STATEMENT FUNCTION CALL)
SFRTN	74	(STATEMENT FUNCTION RETURN)
SIGN	82	(SEARCH FOR OPTIONAL SIGN)
SIN	95	(SINE)
SINH	98	(HYPERBOLIC SINE)
SNGL	56	(CONVERT FROM DOUBLE TO SINGLE PRECISION)
SPAN	75	(SCRAMBLE AND PROVIDE ARGUMENTS)
SQRT	A1	(SQUARE ROOT)
SSTL	77	(SUBSTRING MOVE)
SSTR	76	(SUBSTRING DESCRIPTOR)
STC	78	(STORE CHARACTERS)
STMA	91	(FORTRAN STATEMENT NUMBER)
STORE	05	(STORE REGISTER)
TAN	97	(TANGENT)
TANH	9A	(HYPERBOLIC TANGENT)
TIME	7E	(PROCESSOR TIME)
VD	8A	(VALIDATE DESCRIPTOR)
WFF	85	(WRITE E-FORMAT)
WFF	86	(WRITE F-FORMAT)
WID	80	(WRITE INTEGER DIGITS)
WIF	93	(WRITE I-FORMAT)
XTRACT	88	(EXTRACT BITS)



BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

LISTING BY NUMERIC OP-CODE (WITHIN S-OP GROUPINGS)

ARITHMETIC REPLACEMENT S-OPS

00		(INVALID)
01	MOVE	(MOVE SINGLE WORD)
02	DMOVE	(MOVE DOUBLE WORD)
03		(INVALID)
04	RMOVE	(MOVE REGISTER)
05	STORE	(STORE REGISTER)
06	LOAD	(LOAD REGISTER)
07	PMOVE	(MOVE MEMORY)
08	IADD	(INTEGER ADD - SS,SS,SD)
09	ISUB	(INTEGER SUBTRACT - SS,SS,SD)
0A	IMUL	(INTEGER MULTIPLY - SS,SS,SD)
0B	IDIV	(INTEGER DIVIDE - SS,SS,SD)
0C	FADD	(FLOATING ADD - SS,SS,SD)
0D	FSUB	(FLOATING SUBTRACT - SS,SS,SD)
0E	FMUL	(FLOATING MULTIPLY - SS,SS,SD)
0F	FDIV	(FLOATING DIVIDE - SS,SS,SD)
10	IARRR	(INTEGER ADD - REG,REG,REG)
11	IARRM	(INTEGER ADD - REG,REG,MEM)
12	IARMR	(INTEGER ADD - REG,MEM,REG)
13	IARMM	(INTEGER ADD - REG,MEM,MEM)
14	IAMRR	(INTEGER ADD - MEM,REG,REG)
15	IAMRM	(INTEGER ADD - MEM,REG,MEM)
16	IAMMR	(INTEGER ADD - MEM,MEM,REG)
17	IAMMM	(INTEGER ADD - MEM,MEM,MEM)
18	ISRRR	(INTEGER SUBTRACT - REG,REG,REG)
19	ISRRM	(INTEGER SUBTRACT - REG,REG,MEM)
1A	ISRRM	(INTEGER SUBTRACT - REG,MEM,REG)
1B	ISRRM	(INTEGER SUBTRACT - REG,MEM,MEM)
1C	ISPRR	(INTEGER SUBTRACT - MEM,REG,REG)
1D	ISPRM	(INTEGER SUBTRACT - MEM,REG,MEM)
1E	ISPMR	(INTEGER SUBTRACT - MEM,MEM,REG)
1F	ISPMM	(INTEGER SUBTRACT - MEM,MEM,MEM)
20	IMRRR	(INTEGER MULTIPLY - REG,REG,REG)
21	IMRRM	(INTEGER MULTIPLY - REG,REG,MEM)
22	IMRRM	(INTEGER MULTIPLY - REG,MEM,REG)
23	IMRRM	(INTEGER MULTIPLY - REG,MEM,MEM)
24	IMPRR	(INTEGER MULTIPLY - MEM,REG,REG)
25	IMPRM	(INTEGER MULTIPLY - MEM,REG,MEM)
26	IMPMR	(INTEGER MULTIPLY - MEM,MEM,REG)
27	IMPMM	(INTEGER MULTIPLY - MEM,MEM,MEM)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

28	IDRRR	(INTEGER DIVIDE - REG,REG,REG)
29	IDRRM	(INTEGER DIVIDE - REG,REG,MEM)
2A	IDRRR	(INTEGER DIVIDE - REG,MEM,REG)
2B	IDRRM	(INTEGER DIVIDE - REG,MEM,MEM)
2C	IDMRR	(INTEGER DIVIDE - MEM,REG,REG)
2D	IDRRM	(INTEGER DIVIDE - MEM,REG,MEM)
2E	IDMRR	(INTEGER DIVIDE - MEM,MEM,REG)
2F	IDRRM	(INTEGER DIVIDE - MEM,MEM,MEM)
30	FARRR	(FLOATING ADD - REG,REG,REG)
31	FARRM	(FLOATING ADD - REG,REG,MEM)
32	FARRR	(FLOATING ADD - REG,MEM,REG)
33	FARRM	(FLOATING ADD - REG,MEM,MEM)
34	FAPRR	(FLOATING ADD - MEM,REG,REG)
35	FAPRR	(FLOATING ADD - MEM,REG,MEM)
36	FAPRR	(FLOATING ADD - MEM,MEM,REG)
37	FAPRR	(FLOATING ADD - MEM,MEM,MEM)
38	FSRRR	(FLOATING SUBTRACT - REG,REG,REG)
39	FSRRM	(FLOATING SUBTRACT - REG,REG,MEM)
3A	FSRRR	(FLOATING SUBTRACT - REG,MEM,REG)
3B	FSRRM	(FLOATING SUBTRACT - REG,MEM,MEM)
3C	FSYRR	(FLOATING SUBTRACT - MEM,REG,REG)
3D	FSRRM	(FLOATING SUBTRACT - MEM,REG,MEM)
3E	FSYRR	(FLOATING SUBTRACT - MEM,MEM,REG)
3F	FSYRR	(FLOATING SUBTRACT - MEM,MEM,MEM)
40	FMRRR	(FLOATING MULTIPLY - REG,REG,REG)
41	FMRRM	(FLOATING MULTIPLY - REG,REG,MEM)
42	FMRRR	(FLOATING MULTIPLY - REG,MEM,REG)
43	FMRRM	(FLOATING MULTIPLY - REG,MEM,MEM)
44	FMRRR	(FLOATING MULTIPLY - MEM,REG,REG)
45	FMRRM	(FLOATING MULTIPLY - MEM,REG,MEM)
46	FMRRR	(FLOATING MULTIPLY - MEM,MEM,REG)
47	FMRRM	(FLOATING MULTIPLY - MEM,MEM,MEM)
48	FDRRR	(FLOATING DIVIDE - REG,REG,REG)
49	FDRRM	(FLOATING DIVIDE - REG,REG,MEM)
4A	FDRRR	(FLOATING DIVIDE - REG,MEM,REG)
4B	FDRRM	(FLOATING DIVIDE - REG,MEM,MEM)
4C	FDMRR	(FLOATING DIVIDE - MEM,REG,REG)
4D	FDRRM	(FLOATING DIVIDE - MEM,REG,MEM)
4E	FDMRR	(FLOATING DIVIDE - MEM,MEM,REG)
4F	FDMRR	(FLOATING DIVIDE - MEM,MEM,MEM)
50	DADD	(DOUBLE PRECISION ADD)
51	DSUB	(DOUBLE PRECISION SUBTRACT)
52	DMUL	(DOUBLE PRECISION MULTIPLY)
53	DDIV	(DOUBLE PRECISION DIVIDE)
A3	BUMP	(BUMP)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

LOGICAL REPLACEMENT AND IF STATEMENT S-OPS

58	IREL	(INTEGER RELATION)
59	IRIF	(INTEGER RELATIONAL IF)
5A	FREL	(FLOATING RELATION)
5B	FRIF	(FLOATING RELATIONAL IF)
5C	DREL	(DOUBLE RELATION)
5D	DRIF	(DOUBLE RELATIONAL IF)
60	LOG	(LOGICAL RELATION)
61	LIF2	(LOGICAL IF - 2 OPERANDS)
62	LNOT	(LOGICAL NOT)
63	LIF1	(LOGICAL IF - 1 OPERAND)

BRANCH S-OPS

64	GOTO	(BRANCH UNCONDITIONAL)
65	AIF	(ARITHMETIC IF)
66	CGO	(COMPUTED GOTO)
67	AGO	(ASSIGNED GO TO)

TYPE AND SIGN CONVERSION S-OPS

54	FLCAT	(CONVERT FROM INTEGER TO FLOATING)
55	IFIX	(CONVERT FROM FLOATING TO INTEGER)
56	SNGL	(CONVERT FROM DOUBLE TO SINGLE PRECISION)
57	DBL	(CONVERT FROM SINGLE TO DOUBLE PRECISION)
68	ABS	(ABSOLUTE VALUE)
69	NEG	(CHANGE SIGN)

SUBSCRIPT VALUE COMPUTATION S-OPS

8A	VD	(VALIDATE DESCRIPTOR)
8B	BAT	(BUILD ARRAY TABLE)
8C	BAAT	(BUILD ASSUMED SIZE ARRAY TABLE)
8D	CS	(COMPUTE SUBSCRIPT VALUE)
8E	CSE	(CS WITH BOUNDS CHECKING)
8F	CSV	(CS WITH ARRAY TABLE, ALSO CHECKS BOUNDS)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

CO LOOP MAINTENANCE

90 DO.UP (UPDATE CO LOOP)

CHARACTER TYPE S-OPS

5E	CREL	(CHARACTER RELATIONAL)
5F	CRIF	(CHARACTER RELATIONAL IF)
6C	LEN	(CHARACTER LENGTH)
76	SSTR	(SUBSTRING DESCRIPTOR)
77	SSTL	(SUBSTRING MOVE)
78	STC	(STORE CHARACTERS)
79	CATD	(CONCATENATION WITH DESCRIPTOR)
7A	MVC	(MOVE CHARACTER)
7B	CAT	(CONCATENATION)

SUBROUTINE LINKAGE S-OPS

6D	SAVE	(SAVE/RESTORE REGISTERS)
6E	LB	(LOCAL BASE)
6F	PASS	(PASS DESCRIPTOR)
70	CALL	(SUBROUTINE CALL)
71	DCAL	(DYNAMIC SUBROUTINE CALL)
72	RTN	(SUBROUTINE RETURN)
73	SFCL	(STATEMENT FUNCTION CALL)
74	SFRTN	(STATEMENT FUNCTION RETURN)
75	SPAM	(SCRAMBLE AND PROVIDE ARGUMENTS)

SPECIAL FUNCTION S-OPS

91	STMN	(FORTRAN STATEMENT NUMBER)
92	HMON	(HARDWARE MONITOR) *

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18CC/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### PRIVILEGED USER S-QPS

6A	ADDR	(BASE RELATIVE ADDRESS)
6B	DESC	(MAKE DESCRIPTOR)
78	STC	(STORE CHARACTERS)
7C	COMM	(COMMUNICATE)
7D	LCR	(LOAD COMMUNICATE REPLY)
7E	TIME	(PROCESSOR TIME)
7F	DS	(DISCONTINUE JOB)
80	WIC	(WRITE INTEGER DIGITS)
81	ENRY	(GET INTEGER INPUT)
82	SIGN	(GET OPTIONAL SIGN)
83	REAL	(GET FLOATING INPUT)
84	NEXT	(EXAMINE NEXT INPUT)
85	WEF	(WRITE E-FORMAT)
86	WFF	(WRITE F-FORMAT)
87	FANC	(FETCH AND CLEAR ERROR CONCITION)
88	XTRACT	(EXTRACT BITS)
89	INSERT	(INSERT BITS)
93	WIF	(WRITE I-FORMAT)

QPS 2942 - 2A22 MAY ALSO BE USED AS PRIVILEGED OPERATORS WITH THE  
 FORMAT: PRIV <DESIRED FUNCTION> (STD\_SOURCE...  
 STD\_DESTINATION). THESE OPERATORS ARE DESCRIBED UNDER  
 TRIGONOMETRIC AND OTHER FUNCTIONS.

### TRIGONOMETRIC AND OTHER FUNCTIONS

NOTE: ALL OF THESE OPERATORS MAY ALSO BE USED AS PRIVILEGED  
 OPERATORS.

94	AMOD	(REMAINDER)
95	SIN	(SINE)
96	COS	(COSINE)
97	TAN	(TANGENT)
98	SINH	(HYPERBOLIC SINE)
99	COSH	(HYPERBOLIC COSINE)
9A	TANH	(HYPERBOLIC TANGENT)
9B	ASIN	(ARCSINE)
9C	ACOS	(ARCCOSINE)
9D	ATAN	(ARCTANGENT)
9E	AINT	(FLOOR)
9F	ALOG	(NATURAL LOG)

BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
F.S. 2222 2749 (B)

A0	ALOG10	(LOG TO BASE 10)
A1	SQRT	(SQUARE ROOT)
A2	EXP	(EXPONENTIAL)

2442 - 2FF2 (INVALID)

\* NOT IMPLEMENTED

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (8)

## FORMATS

## REGISTERS

### SCRATCH.PAD ASSIGNMENTS

	A		B		
15		TEMP.HI		TEMP.LO	
14		ITEMP		SCRATCH.3	
13		SCRATCH.5		SCRATCH.6	
12		CODE.SEG.NUMBER		SCRATCH.4	
11		SCRATCH		IX-REG 11	
10		DATA.DICT.BASE		IX-REG 10	
9		BR.REG		IX-REG 9	
8		LOCAL.DATA.BLOCK		IX-REG 8	
7		TRANSFER.VECTOR / CSB		IX-REG 7	
6		NEXT.INST.PTR		IX-REG 6	
5		ACC.5 (HI)		IX-REG 5	(ACC.5 LO)
4		ACC.4 (HI)		IX-REG 4	(ACC.4 LO)
3		ACC.3 (HI)		IX-REG 3	(ACC.3 LO)
2		ACC.2 (HI)		IX-REG 2	(ACC.2 LO)
1		ACC.1 (HI)		IX-REG 1	(ACC.1 LO)
0		SCRATCH.1		SCRATCH.2	

## ERROR CONDITION INFORMATION

LOCATED BETWEEN BASE-LIMIT AREA (AT A BASE-RELATIVE ADDRESS), THIS INFORMATION CONSISTS OF SIX BITS.

AT BYTE ADDRESS 27, FOLLOWING A FILLER OF BIT(2), ARE THE SIX BITS AS FOLLOWS:

- FIRST BIT IS SET IF AN OVERFLOW CONDITIONS OCCURS.
- SECOND BIT IS SET IF AN EXPONENT UNDERFLOW CONDITION OCCURS.
- THIRD BIT IS SET IF A DIVIDE-BY-ZERO CONDITION OCCURS.
- FOURTH BIT IS SET IF OVERFLOW IS PERMITTED.
- FIFTH BIT IS SET IF UNDERFLOW IS PERMITTED.
- SIXTH BIT IS SET IF DIVIDE BY ZERO IS PERMITTED.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### VALUES

INTEGER		
TWO'S COMPLEMENT INTEGER		BIT(32)
LOGICAL		
FALSE = 0, TRUE = NOT 0		BIT(32)
REAL		
SIGN		BIT(1)
EXPONENT (EXCESS 64; RADIX 16)		BIT(7)
MANTISSA		BIT(24)
DOUBLE PRECISION		
SIGN		BIT(1)
EXPONENT (EXCESS 64; RADIX 16)		BIT(7)
MANTISSA		BIT(56)
CHARACTER		
BYTE		BIT(8)
EXTERNAL_CODE_ADDRESS		
FILLER		BIT(4)
SEGMENT_NUMBER		BIT(10)
BIT DISPLACEMENT INTO SEGMENT		BIT(18)

### LOCAL DATA BLOCK

32 BITS	48 BITS	48 BITS	32 BITS EACH
-----			
RETURN ADDRESS	ARG1	ARG2	...   LOCAL VARIABLES   ...
-----			



EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (8)

### SUBROUTINE LINKAGE MECHANISM

A 48-BIT ARGUMENT DESCRIPTOR IS BUILT USING THE "PASS" S-OP FOR EACH ACTUAL ARGUMENT PASSED.

EXECUTION OF THE "CALL" S-OP IN THE CALLING SUBROUTINE WILL UPDATE THE RETURN-ADDRESS FIELD OF THE LOCAL DATA BLOCK ASSOCIATED WITH THE CALLED SUBROUTINE AND CONTROL IS PASSED TO THE CALLED SUBROUTINE.

EXECUTION OF THE "RTN" S-OP IN THE CALLED SUBROUTINE WILL USE THE RETURN ADDRESS INFORMATION IN ITS LOCAL DATA BLOCK TO GET BACK TO THE CALLING SUBROUTINE. THE RETURN-ADDRESS FIELD IS CLEARED BEFORE CONTROL IS REVERTED BACK TO THE CALLING SUBROUTINE.

### LAYOUT TABLE

THE LAYOUT TABLE CONTAINS NAMES AND TYPES OF VARIABLES.

FOR EACH SUBROUTINE AND COMMON BLOCK (SEEN IN A BLOCK DATA SUBPROGRAM ONLY), THERE WILL BE AN ENTRY OF THE FORMAT:

NAME OF SUBPROGRAM	CHARACTER (6)
#TV_ENTRIES	EIT ( 8)
#SYMBOL_ENTRIES	BIT (20)
ADDRESS OF LOCAL DATA BLOCK	EIT (20)
TV*S(#TV_ENTRIES)	CHARACTER (6)
SYMBOLS(#SYMBOL_ENTRIES) WITH THE FORMAT:	

NAME	CHARACTER (6)
LENGTH	BIT ( 8)
ARRAY_VARIANT	BIT ( 1)
TYPE	EIT ( 3)
INDIRECT_VARIANT	BIT ( 1)
BLOCK-RELATIVE ADDRESS	EIT (19)

WHERE LENGTH WILL HAVE A VALUE OF ZERO FOR A NON-CHARACTER SYMBOL AND TYPE MAY HAVE ONE OF THE FOLLOWING VALUES:

1 = INTEGER  
 2 = REAL  
 3 = DOUBLE  
 4 = COMPLEX

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

5 = LOGICAL  
 6 = CHARACTER  
 7 = LABEL

SYMBOLS(#SYMBOL\_ENTRIES) MAY ALSO HAVE THE FOLLOWING  
 FORMAT IF THE PRECEEDING ENTRY DESCRIBES AN ACTUAL  
 ARRAY (ARRAY\_VARIANT ON AND INDIRECT\_VARIANT OFF):

NUMBER OF DIMENSIONS	BIT ( 3)
NUMBER OF ELEMENTS	BIT (17)
LOWER BOUND OF 1ST DIM.	BIT ( 8)
UPPER BOUND OF 1ST DIM.	BIT (12)
LOWER BOUND OF 2ND DIM.	BIT ( 8)
UPPER BOUND OF 2ND DIM.	BIT (12)
LOWER BOUND OF 3RD DIM.	BIT ( 8)
UPPER BOUND OF 3RD DIM.	BIT (12)

THE FIELD OF NUMBER OF DIMENSIONS WILL BE EQUAL  
 TO ZERO IF THERE ARE MORE THAN 3 DIMENSIONS OR  
 IF THE VALUES OF LOWER AND/OR UPPER BOUNDS EXCEED  
 THE SIZES OF THEIR RESPECTIVE FIELDS.

TV'S IS AN ARRAY OF TRANSFER VECTOR ITEMS HAVING THE NUMBER OF  
 ELEMENTS AS SPECIFIED BY #TV\_ENTRIES; THIS INFORMATION WILL BE  
 USED BY RUN-TIME INTERACTIVE DEBUGGER (RAID).

SYMBOLS IS AN ARRAY OF SYMBOLS HAVING THE NUMBER OF ELEMENTS AS  
 SPECIFIED BY #SYMBOL\_ENTRIES, AND IS USED BY THE DUMP ANALYZER.

### TRANSFER VECTOR

LENGTH OF TRANSFER VECTOR	BIT(24)
SEGMENT_DISPLACEMENT_CODE_ADDRESS(ES) (NUMBER OF CODE ADDRESSES VARIES, 0 INCLUDED)	
FILLER (WITH VALUE OF 202)	BIT( 4)
CODE_SEGMENT_NUMBER	BIT(10)
BIT DISPLACEMENT INTO SEGMENT	BIT(18)
BLOCK ADDRESS (MAY BE LOCAL DATA BLOCK AND/OR COMMON BLOCK ADDRESS)	
FILLER (WITH VALUE OF 28002)	BIT(12)
BASE-RELATIVE BLOCK ADDRESS (BYTE OFFSET)	BIT(20)

BUROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

THE TRANSFER VECTOR IS ZERO-RELATIVE, BEGINNING AT THE RIGHT WHERE THE FIRST S-OP STARTS. ENTRY ZERO OF THE TRANSFER VECTOR WILL ALWAYS BE THE LOCAL DATA BLOCK ADDRESS. OTHER ENTRIES MAY BE EITHER CODE ADDRESSES OR COMMON BLOCK ADDRESSES, WHICH MAY BE INTERMIXED IN THE TRANSFER VECTOR.

24 BITS	32 BITS EACH	32 BITS EACH	32 BITS
I TV I LENGTH	I COMMON I BLK ADDR	I ... I ADDRESS	I ... I LOCAL DATA I S-OPS ... I BLOCK ADDR
	TV(N)	TV(I) TV(1)	TV(0)

#### ASSIGNED GOTO AND FORMAT TABLE

ASSIGNED GOTO AND FORMAT TABLE IS AN ARRAY OF N ELEMENTS WHERE (N-1) IS THE NUMBER OF UNIQUE LABELS ASSIGNED IN A PROGRAM UNIT.

#### EXECUTABLE STATEMENT LABEL ENTRY FORMAT:

FILLER BIT( 8)  
 CODE ADDRESS (SEE DESCRIPTION IN ADDRESS SECTION) BIT(24)

#### FORMAT STATEMENT LABEL ENTRY FORMAT:

BYTE OFFSET INTO LOCAL DATA BLOCK BIT(32)

A SAMPLE TABLE, AS CONSTRUCTED BY THE COMPILER, WOULD LOOK AS FOLLOWS (SEE SAMPLE FORTRAN CODE BELOW) :

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

```

<---- COLUMN 1 = 32 BITS -----> <-- COLUMN 2 = 32 BITS ----->
-----
|                               41 < ROW 1 UNUSED >                               |
101100000000000000000000000000000001 CODE ADDRESS FOR LABEL 5 |
1100000000000000000000000000000000001 DATA ADDRESS FOR FORMAT 15 |
1001000000000000000000000000000000001 CODE ADDRESS FOR LABEL 10 |
1100000000000000000000000000000000001 DATA ADDRESS FOR FORMAT 20 |
-----

```

THE VALUE IN THE FIRST ROW, FIRST COLUMN IS THE TOTAL NUMBER OF CODE ADDRESSES AND FORMAT LABEL ADDRESSES IN COLUMN 2.

THE FIRST BIT OF EACH ROW IN COLUMN 1 WILL EQUAL 0 (FOR CODE ADDRESSES) OR 1 (FOR FORMAT LABEL ADDRESSES).

NOTE THAT FOR FORMAT LABEL ADDRESSES, ALL BITS IN COLUMN 1 ARE ZERO EXCEPT THE FIRST. THIS ZERO AREA IS THE AREA POINTED TO BY OP.3 IN THE ASSIGNED GOTO OP, BUT OP.3 DOES NOT EXIST FOR VARIANT 6 OF THE PASS OP.

EACH BIT ON COLUMN 1 (EXCEPT BIT 0) SPECIFIES WHETHER A BRANCH CAN BE MADE TO THE CODE ADDRESS IN COLUMN 2 FOR A PARTICULAR ASSIGNED GOTO STATEMENT. BIT 1 IS FOR THE FIRST ASSIGNED GOTO STATEMENT IN THE PROGRAM UNIT, BIT 2 IS FOR THE SECOND ASSIGNED GOTO STATEMENT IN THE PROGRAM UNIT, ETCETERA.

IN THE SAMPLE TABLE, FOR EXAMPLE, THE FIRST ASSIGNED GOTO CAN ONLY BRANCH TO THE FIRST CODE ADDRESS, AND THE SECOND ASSIGNED GOTO CAN BRANCH TO EITHER CODE ADDRESS. THE FORTRAN CODE TO GENERATE SUCH A TABLE MIGHT LOOK LIKE:

```

          ASSIGN 5 TO N
          ASSIGN 15 TO I
          GO TO N(5)
5         CONTINUE
10        ASSIGN 10 TO N
          GO TO N(10,5)
          ASSIGN 20 TO J
15        FORMAT( ... )
20        FORMAT( ... )

```

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

## STANDARD INDEX

### STD\_INDEX

VARIANT BIT(1)  
 INDEX BIT VARYING

VARIANT	INDEX		
-----	-----		
0	REGISTER_NUMBER	BIT(4)	
1	TAG	BIT(3)	FOLLOWED BY
	ADDRESS	BIT VARYING	

TAG	ADDRESS
---	-----
0	DIRECT_ADDR
1	INDIRECT_ADDR
2	PO_ADDR
3-7	UNDEFINED

## ADDRESSES

DIRECT\_ADDR  
 LOCAL BLOCK VS. NON-PAGED COMMON BLOCK VARIANT BIT(1)  
 IF VARIANT=0, BYTE OFFSET BIT(19)  
 THE 19 BITS OF OFFSET ARE IN 2'S COMPLEMENT.  
 IE., IF LEADING BIT IS ON, THE VALUE IS NEGATIVE.  
 IF VARIANT=1, TV\_INDEX FOLLOWED BY BIT(10)  
 BYTE OFFSET. BIT(19)

INDEXED\_DIRECT\_ADDR  
 DIRECT\_ADDR BIT VARYING  
 STD\_INDEX (CONTENT CONTAINS ELEMENT SUBSCRIPT) BIT VARYING

CHARACTER\_DIRECT\_ADDR  
 DIRECT\_ADDR BIT VARYING  
 LENGTH (ELEMENT LENGTH IN BYTES) BIT(8)

BLRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

INDEXED_CHARACTER_DIRECT_ADDR		
DIRECT_ADDR		BIT VARYING
STD_INDEX	(CONTENT CONTAINS ELEMENT SUBSCRIPT)	BIT VARYING
LENGTH	(ELEMENT LENGTH IN BYTES)	BIT(8)
PO_ADDR		
PAGE NUMBER		BIT(10)
BYTE OFFSET INTO PAGE		BIT(10)
INDEXED_PO_ADDR		
PO_ADDR		BIT(20)
STD_INDEX	(CONTENT CONTAINS ELEMENT SUBSCRIPT)	BIT VARYING
CHARACTER_PO_ADDR		
PO_ADDR		BIT(20)
LENGTH	(ELEMENT LENGTH IN BYTES)	BIT(8)
INDEXED_CHARACTER_PO_ADDR		
PO_ADDR		BIT(20)
STD_INDEX	(CONTENT CONTAINS ELEMENT SUBSCRIPT)	BIT VARYING
LENGTH	(ELEMENT LENGTH IN BYTES)	BIT(8)
INDIRECT_ADDR		
LOCAL BLOCK VS. NON-PAGED COMMON BLOCK VARIANT		BIT(1)
IF VARIANT=0, BYTE OFFSET.		BIT(19)
THE 19 BITS OF OFFSET ARE IN 2'S COMPLEMENT.		
IE., IF THE LEADING BIT IS ON, THE VALUE IS NEGATIVE.		
IF VARIANT=1, TV_INDEX FOLLOWED BY		BIT(10)
BYTE OFFSET		BIT(19)
INDEXED_INDIRECT_ADDR		
INDIRECT_ADDR		BIT VARYING
STD_INDEX	(CONTENT OF ELEMENT SUBSCRIPT APPLIES TO ADDRESS PASSED)	BIT VARYING
CHARACTER_INDIRECT_ADDR		
INDIRECT_ADDR		BIT VARYING
INDEXED_CHARACTER_INDIRECT_ADDR		
INDIRECT_ADDR		BIT VARYING
STD_INDEX	(CONTENT OF ELEMENT SUBSCRIPT APPLIES TO ADDRESS PASSED)	BIT VARYING

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

CODE_ADDRESS		
FILLER	(RESERVED FOR FURTHER DEVELOPMENT)	BIT(6)
BIT DISPLACEMENT INTO CODE SEGMENT		BIT(18)
EXTERNAL_CODE_ADDRESS (PASSED)		
CODE SEGMENT NUMBER		BIT(14)
BIT DISPLACEMENT INTO CODE SEGMENT		BIT(18)
FILLER		BIT(16)
REGISTER_NUMBER		BIT(4)
MEM_ADDR	(BYTE OFFSET INTO LOCAL DATA BLOCK)	BIT(20)
INDIRECT_DESCRIPTOR		
NON-INDEXED VS. INDEXED VARIANT		BIT(1)
REGISTER_NUMBER		BIT(4)
IF VARIANT=1, STD_INDEX		BIT VARYING
BASIC_DESCRIPTOR		
STATIC VS. DYNAMIC MEMORY VARIANT		BIT(1)
IF VARIANT=0, BASE-RELATIVE ADDRESS		BIT(19)
IF VARIANT=1, PAGE/OFFSET ADDRESS		BIT(19)
LENGTH		BIT(8)
ARRAY_DESC_PASSED		
BASIC_DESCRIPTOR		BIT(28)
NUMBER_ELEMENTS_PASSED		BIT(20)
CHAR_DESC_PASSED		
BASIC_DESCRIPTOR		BIT(28)
FILLER		BIT(20)
PO_ADDR_PASSED		
BASIC_DESCRIPTOR		BIT(28)
FILLER		BIT(20)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### STANDARD SOURCE

#### STD\_SOURCE

VARIANT            BIT(4)  
 SOURCE            BIT VARYING

VARIANT	SOURCE_LOCATION	CONTENTS (SOURCE_LOCATION)
0	INTERPRETER CREATES	INTEGER VALUE = 0
1	INTERPRETER CREATES	INTEGER VALUE = 1
2	INTERPRETER CREATES	INTEGER VALUE = 2
3	INTERPRETER CREATES	INTEGER VALUE = 3
4	IN LINE	4 BIT IN-LINE LITERAL
5	IN LINE	10 BIT IN-LINE LITERAL
6	IN LINE	32 BIT IN-LINE LITERAL
7	IN LINE	64 BIT IN-LINE LITERAL
8	DIRECT_ADDR	32/64 BIT VARIABLE
9	INDIRECT_ADDR	BASIC DESCRIPTOR
A	PO_ADDR	32/64 BIT VARIABLE
B	UNUSED	
C	INDEXED_DIRECT_ADDR	SAME AS 8
D	INDEXED_INDIRECT_ADDR	SAME AS 9
E	INDEXED_PO_ADDR	SAME AS A
F	REGISTER_NUMBER	32/64 BIT VARIABLE

### STANDARD DESTINATION

#### STD\_DESTINATION

VARIANT            BIT(3)  
 DESTINATION        BIT VARYING

VARIANT	DESTINATION
0	DIRECT_ADDR
1	INDIRECT_ADDR
2	PO_ADDR
3	UNUSED
4	INDEXED_DIRECT_ADDR
5	INDEXED_INDIRECT_ADDR



BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

6 INDEXED\_PO\_ADDR  
 7 REGISTER\_NUMBER

STANDARD CHARACTER SOURCE

STD\_CHAR\_SOURCE

VARIANT BIT(4)  
 SOURCE BIT VARYING

<u>VARIANT</u>	<u>SOURCE_ADDRESS</u>	<u>CONTENTS (SOURCE_ADDRESS)</u>
0	INVALID	
1	INVALID	
2	INVALID	
3	INVALID	
4	INVALID	
5	INVALID	
6	INVALID	
7	LENGTH(BIT(8)) // IN_LINE	BYTE_LENGTH IN-LINE LITERAL
8	CHARACTER_DIRECT_ADDR	CHARACTER
9	CHARACTER_INDIRECT_ADDR	CHARACTER_DESCRIPTOR
A	CHARACTER_PO_ADDR	CHARACTER
B	UNUSED	
C	INDEXED_CHARACTER_DIRECT_ADDR	CHARACTER
D	INDEXED_CHARACTER_INDIRECT_ADDR	CHARACTER_DESCRIPTOR
E	INDEXED_CHARACTER_PO_ADDR	CHARACTER
F	INDIRECT_DESCRIPTOR	CHARACTER_DESCRIPTOR

BLRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (8)

**STANDARD CHARACTER DESTINATION**

**STD\_CHAR\_DESTINATION**

VARIANT BIT(3)  
 DESTINATION BIT VARYING

VARIANT	DESTINATION
0	CHARACTER_DIRECT_ADDR
1	CHARACTER_INDIRECT_ADDR
2	CHARACTER_PC_ADDR
3	UNUSED
4	INDEXED_CHARACTER_DIRECT_ADDR
5	INDEXED_CHARACTER_INDIRECT_ADDR
6	INDEXED_CHARACTER_PC_ADDR
7	INDIRECT_DESCRIPTOR

**RUN-TIME DIMENSION TABLE**

AS AN EXAMPLE,

FOR ARRAY A(J,N,2): BAT/BAAT WILL CONSTRUCT:

SCALE BIT(24)	BASE_OFFSET BIT(24)
LENGTH	LOWER BOUND BIT(24)   UPPER BOUND BIT(24)
<J>	
<N*J>	
<N*J*2>	

..... ASSUMED SIZE

LAST UPPER BOUND = MAXIMUM POSSIBLE SUBSCRIPT VALUE.

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### ARITHMETIC REPLACEMENT S-OPS

\*\*\*\*\*  
 \* MOVE \* (MOVE SINGLE WORD)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES
I 2012	I STD_SOURCE	I STD_DESTINATION

STD\_SOURCE ::= REAL I INTEGER

OPERATION:

STD\_DESTINATION := STD\_SOURCE

\*\*\*\*\*  
 \* DMOVE \* (MOVE DOUBLE WORD)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES
I 2022	I STD_SOURCE	I STD_DESTINATION

THE STANDARD SOURCE IS A DOUBLE PRECISION VALUE.

OPERATION:

STD\_DESTINATION := STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B170C FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* RMOVE \* (MOVE REGISTER)  
 \*\*\*\*\*

FORMAT:

8 BITS	4 BITS	4 BITS
2042	REGISTER_NUMBER	REGISTER_NUMBER
	OP1	OP2

OPERATION:

CONTENTS OF OP2 GET CONTENTS OF OP1.

\*\*\*\*\*  
 \* STORE \* (STORE REGISTER)  
 \*\*\*\*\*

FORMAT:

8 BITS	4 BITS	20 BITS
2052	REGISTER_NUMBER	MEM_ADDR

OPERATION:

CONTENTS OF SPECIFIED REGISTER STORED AT SPECIFIED MEM\_ADDR.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

\*\*\*\*\*  
 \* LOAD \* (LCAC REGISTER)  
 \*\*\*\*\*

FORMAT:

8 BITS	20 BITS	4 BITS
2062	MEM_ADDR	REGISTER_NUMBER

OPERATION:

VALUE AT MEM\_ADDR LOADED TO SPECIFIED REGISTER.

\*\*\*\*\*  
 \* MMOVE \* (MOVE MEMORY)  
 \*\*\*\*\*

FORMAT:

8 BITS	20 BITS	20 BITS
2072	MEM_ADDR	MEM_ADDR
	OP1	OP2

OPERATION:

VALUE AT OP1 MOVED TO OP2.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*

\* ADD \* --

\* SUB \* | WITH OPERANDS OF STANDARD SOURCE, STANDARD SOURCE, AND

\* MUL \* | STANDARD DESTINATION.

\* DIV \* --

\*\*\*\*\*

	ADD	SUBTRACT	MULTIPLY	DIVIDE
DOUBLE PRECISION	DADD 2502	DSUB 2512	DMUL 2522	DDIV 2532
REAL	FADD 20C2	FSUB 20D2	FMUL 20E2	FDIV 20F2
INTEGER	IADD 2082	ISUB 2092	IMUL 20A2	IDIV 20B2

FORMAT:

8 BITS	VARIES		VARIES	VARIES
OPCCDE	STD_SOURCE	STD_SOURCE	STD_DESTINATION	
	OPERAND.1	OPERAND.2	OPERAND.3	

OPERATION:

STD\_DESTINATION := OPERAND.1 [+,-,\*,/] OPERAND.2

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

```

*****
* ADD * --          | REG,REG,REG      MEM,MEM,MEM |
* SUB *   | WITH OPERANDS | REG,REG,MEM    MEM,REG,MEM |
* MUL *   | OF ----->  | REG,MEM,MEM    MEM,MEM,REG |
* DIV * --  WHERE REG IS A | REG,MEM,REG    MEM,REG,REG |
*****          REGISTER_NUMBER
                  AND MEM IS A
                  MEM_ADDR.
    
```

	ADD	SUBTRACT	MULTIPLY	DIVIDE
MEM, MEM, MEM	I IAMMM 2172   FAMMM 2372	I ISMMM 21F2   FSMMM 23F2	I IMMMM 2272   FMMMM 2472	I IDMMM 22F2   FDMMM 24F2
MEM, MEM, REG	I IAMMR 2162   FAMMR 2362	I ISMMR 21E2   FSMMR 23E2	I IMMRR 2262   FMRRR 2462	I IDMMR 22E2   FDMMR 24E2
MEM, REG, MEM	I IAMRM 2152   FAMRM 2352	I ISMRM 21D2   FSMRM 23D2	I IMRRM 2252   FMRRM 2452	IDMRM 22D2   FDMRM 24D2
MEM, REG, REG	I IAMRR 2142   FAMRR 2342	ISMRR 21C2   FSMRR 23C2	IMRRR 2242   FMRRR 2442	IDMRR 22C2   FDMRR 24C2
REG, MEM, MEM	I IARMM 2132   FARMM 2332	ISRMM 21B2   FSRRM 23B2	INRRM 2232   FNRRM 2432	IDRMM 22B2   FDRMM 24B2
REG, MEM, REG	I IARMR 2122   FARMR 2322	ISRRM 21A2   FSRRM 23A2	INRRR 2222   FNRRR 2422	IDRMR 22A2   FDRMR 24A2
REG, REG, MEM	I IARRM 2112   FARRM 2312	ISRRM 2192   FSRRM 2392	INRRM 2212   FNRRM 2412	IDRRM 2292   FDRRM 2492
REG, REG, REG	I IARRR 2102   FARRR 2302	ISRRR 2182   FSRRR 2382	INRRR 2202   FNRRR 2402	IDRRR 2282   FDRRR 2482

FORMAT:

```

      8 BITS
-----
| OPCODE | ADDR1 | ADDR2 | ADDR3 |
-----
          OP.1   OP.2   OP.3
    
```

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18G0/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (E)

## OPERATION:

CONTENTS (ADDR3) := CONTENTS (ADDR1) [+,-,\*,/,] CONTENTS (ADDR2)

\*\*\*\*\*

\* BUMP \* (BUMP)

\*\*\*\*\*

## FORMAT:

8 BITS	VARIES	24 BITS
-----		
I 2432	I STD_DESTINATION	I LITERAL I
-----		

## OPERATION:

STD\_DESTINATION := STD\_DESTINATION + LITERAL

WHERE THE ADDITION PERFORMED IS AN INTEGER ADD.



BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### LOGICAL REPLACEMENT AND IF STATEMENT S-OPS

\*\*\*\*\*

\* IREL \* 2582 (RELATION) (STD.SOURCES ARE OF TYPE INTEGER)  
 \* FREL \* 25A2 " " " " REAL  
 \* DREL \* 25C2 " " " " DOUBLE

\*\*\*\*\*

#### FORMAT:

8 BITS	VARIES		3 BITS	VARIES
OPCODE	STD_SOURCE	STD_SOURCE	MASK	STD_DESTINATION
	OP.1	OP.2	CP.3	OP.4

#### OPERATION:

FOR INTERPRETATION OF MASK, SEE "CRIF" (OP 25D2).

STD\_DESTINATION := TRUE IF OP.1 RELATES TO OP.2 IN ANY OF THE WAYS SPECIFIED BY THE MASK, OTHERWISE FALSE.

TRUE = 2FFFFFFF2 AND FALSE = 200000002.

\*\*\*\*\*

\* IRIF \* 2592 (INTEGER RELATIONAL IF)  
 \* FRIF \* 25B2 (FLOATING POINT RELATIONAL IF)  
 \* DRIF \* 25D2 (DOUBLE PRECISION RELATIONAL IF)

\*\*\*\*\*

#### FORMAT:

8 BITS	VARIES	VARIES	3 BITS	24 BITS
OPCODE	STD_SOURCE	STD_SOURCE	MASK	CODE_ADDRESS
OPCODE	OPERAND.1	OPERAND.2	CP.3	OPERAND.4

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/81700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

OPERATION:

FOR A MASK OF	THEN BRANCH TO CODE ADDRESS IF
1	OPERAND.1 .LE. OPERAND.2
2	OPERAND.1 .GE. OPERAND.2
3	OPERAND.1 .EQ. OPERAND.2
4	OPERAND.1 .NE. OPERAND.2
5	OPERAND.1 .LT. OPERAND.2
6	OPERAND.1 .GT. OPERAND.2

WHERE OPERAND.1 AND OPERAND.2 ARE OF TYPE

INTEGER	IF OP-CODE IS	IRIF ,
REAL	IF OP-CODE IS	FRIF ,
DOUBLE PRECISION	IF OP-CODE IS	DRIF .

\*\*\*\*\*  
 \* LOG \* (LOGICAL RELATION)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES	3 BITS	VARIABLES
-----				
I 2602	I STD_SOURCE	I STD_SOURCE	I LOP	I STD_DESTINATION
-----				
OPCODE	OP.1	OP.2	OP.3	OP.4

OPERATION:

STD\_DESTINATION := OP.1 LOP OP.2  
 WHERE LOP IS:

0	==>	NEQV	(EXOR)
1	==>	AND	
2	==>	OR	
3			(NOT USED)
4	==>	EQV	(NECR)
5	==>	NAND	
6	==>	NOR	
7			(NOT USED)

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

\*\*\*\*\*  
 \* LIF2 \* (LOGICAL IF - 2 OPERANDS)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES		3 BITS	24 BITS
1 2612	1 STD_SOURCE	1 STD_SOURCE	1 LCP	1 CODE_ADDRESS
OPCODE	OP.1	OP.2	OP.3	OP.4

OPERATION:

FOR INTERPRETATION OF THE MASK, SEE "LOG" (OP 2602).

BRANCH TO CODE\_ADDRESS IF THE RESULT OF (OP.1 MASK OP.2) IS TRUE

\*\*\*\*\*  
 \* LNQT \* (LOGICAL NOT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	
1 2622	1 STD_SOURCE	1 STD_DESTINATION
OPCODE	OP.1	OP.2

OPERATION:

STD\_DESTINATION := COMPLEMENT(STD\_SOURCE)

BLRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

\*\*\*\*\*

\* LIF1 \* (LOGICAL IF - 1 OPERAND)

\*\*\*\*\*

FORMAT:

8 BITS	1	VARIES		24 BITS
-----				
I 2632	I V	I STD_SOURCE	I	CODE_ADDRESS I
-----				
OPCODE	OP.1	OP.2		OP.3

OPERATION:

V=0 ==> IF STD\_SOURCE TRUE THEN BRANCH TO CODE\_ADDRESS.  
 V=1 ==> IF STD\_SOURCE FALSE THEN BRANCH TO CODE\_ADDRESS.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (6)

### BRANCH S-OPS

\*\*\*\*\*

\* GOTO \* (UNCONDITIONAL BRANCH)

\*\*\*\*\*

FORMAT:

8 BITS	24 BITS
2642	CODE_ADDRESS
OPCODE	OPERAND.1

OPERATION:

BRANCH TO THE CODE\_ADDRESS (DISPLACEMENT INTO CODE SEGMENT).

\*\*\*\*\*

\* AIF \* (ARITHMETIC IF)

\*\*\*\*\*

FORMAT:

8 BITS	1	VARIES	24 BITS	24 BITS	24 BITS
2652	VAR	STD_SOURCE	CODE_ADDRESS	CODE_ADDRESS	CODE_ADDRESS
OPCODE	OP.1	OPERAND.2	OPERAND.3	OPERAND.4	OPERAND.5

OPERATION:

IF STD\_SOURCE < 0 THEN BRANCH TO CODE\_ADDRESS IN OPERAND.2  
 IF STD\_SOURCE = 0 THEN BRANCH TO CODE\_ADDRESS IN OPERAND.3  
 IF STD\_SOURCE > 0 THEN BRANCH TO CODE\_ADDRESS IN OPERAND.4

VARIANT = 1 ==> STD\_SOURCE IS DOUBLE PRECISION;  
 VARIANT = 0 ==> STD\_SOURCE IS REAL OR INTEGER.

THE VARIANT ONLY EXISTS SO THAT IF THE FOLLOWING STANDARD  
 SOURCE IS A LITERAL, NIP CAN BE ALIGNED CORRECTLY AT CODE  
 ADDRESS AS THE OP IS STEPPED THROUGH.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* CGO \* (COMPUTED GOTO)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	8 BITS	24 BITS	24 BITS
I 2662	I STD_SOURCE I	COUNT	I CODE_ADDRESS	... CODE_ADDRESS I
OPCODE	OPERAND.1	OPERAND.2	OPERAND.3	... OPERAND.COUNT

STD\_SOURCE ::= CONTAINS AN INDEX INTO THE LIST OF CODE ADDRESSES FOLLOWING.

COUNT ::= THE NUMBER OF CODE\_ADDRESS FOLLOWING.

CODE\_ADDRESS := THE ADDRESS TO BRANCH TO.

OPERATION:

BRANCH TO THE CODE ADDRESS POINTED TO BY THE INDEX IN STD\_SOURCE.

\*\*\*\*\*  
 \* AGO \* (ASSIGNED GOTO)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES
I 2672	I STD_DESTINATION I	STD_SOURCE I	STD_SOURCE I
OPCODE	OP.1	OP.2	OP.3

OP.1 POINTS TO A TABLE WITH N ROWS AND 2 COLUMNS WHERE (N-1) IS THE NUMBER OF UNIQUE LABELS ASSIGNED IN A PROGRAM UNIT.

BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
P.S. 2222 2749 (B)

OPERATION:

IF THE MOST SIGNIFICANT BIT OF OP.2 IS ZERO (SUBBIT(OP.2,0,1) = 0), THEN THE VARIABLE IS NOT ASSIGNED A LABEL AND AN ERROR IS GIVEN.

IF THE VALUE OF SUBBIT(OP.2,1) IS GREATER THAN THE VALUE OF THE FIRST ENTRY IN THE TABLE (POINTED TO BY OP.1) THEN THE ASSIGNED LABEL IS NOT IN THE TABLE AND AN ERROR IS GIVEN.

IF THE OP.3-TH BIT OF THE TABLE ENTRY WHOSE ROW EQUALS OP.2 IN THE FIRST COLUMN OF THE TABLE EQUALS 0 (SUBBIT(OP.1(OP.2,1),OP.3,1) = 0), THEN THE LABEL IS OUT OF SCOPE AND AN ERROR IS GIVEN.

OTHERWISE, IF NONE OF THE ABOVE CONDITIONS IS TRUE, BRANCH TO THE CODE ADDRESS FOUND AT THE TABLE ENTRY WHOSE ROW EQUALS OP.2 IN THE SECOND COLUMN OF THE TABLE (OP.1(OP.2,2)).

REFER TO THE FORMAT SECTION AT THE BEGINNING OF THIS DOCUMENT FOR THE DESCRIPTION OF THE ASSIGNED GOTO TABLE.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### TYPE AND SIGN CONVERSION S-OPS

\*\*\*\*\*

\* FLOAT \* (CONVERT FROM INTEGER TO REAL OR DOUBLE PRECISION)

\*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	1 BIT	VARIABLES
1 2542	1 STD_SOURCE	1 VARIANT	1 STD_DESTINATION 1
OPCODE	OPERAND.1	OPERAND.2	OPERAND.3

OPERATION:

STD\_DESTINATION := FLOAT(STD\_SOURCE)

IF VARIANT = 1 , STD\_DESTINATION IS DOUBLE PRECISION;

IF VARIANT = 0 , STD\_DESTINATION IS REAL.

\*\*\*\*\*

\* IFIX \* (CONVERT FROM REAL OR DOUBLE PRECISION TO INTEGER)

\*\*\*\*\*

FORMAT:

8 BITS	1 BIT	VARIABLES	VARIABLES
1 2552	1 VARIANT	1 STD_SOURCE	1 STD_DESTINATION 1
OPCODE	OPERAND.1	OPERAND.2	OPERAND.3

OPERATION:

STD\_DESTINATION := IFIX(STD\_SOURCE)

IF VARIANT = 1 , STD\_SOURCE IS DOUBLE PRECISION;

IF VARIANT = 0 , STD\_SOURCE IS REAL.



BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (8)

\*\*\*\*\*

\* SNGL \* (CONVERT FROM DOUBLE TO REAL)

\*\*\*\*\*

FORMAT:

	8 BITS	VARIABLES	VARIABLES
I	2562	I STD_SOURCE	I STD_DESTINATION I
	OPCODE	OPERAND.1	OPERAND.2

OPERATION:

STD\_DESTINATION := SNGL(STD\_SOURCE)

\*\*\*\*\*

\* DBL \* (CONVERT FROM REAL TO DOUBLE PRECISION)

\*\*\*\*\*

FORMAT:

	8 BITS	VARIABLES	VARIABLES
I	2572	I STD_SOURCE	I STD_DESTINATION I
	OPCODE	OPERAND.1	OPERAND.2

OPERATION:

STD\_DESTINATION := DBL(STD\_SOURCE)

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* ABS \* (ABSOLUTE VALUE)  
 \*\*\*\*\*

FORMAT:

	8 BITS	2 BITS	VARIABLES	VARIABLES
I	2682	1	VARIANT	STD_SOURCE   STD_DESTINATION
	OPCODE	OPERAND.1	OPERAND.2	OPERAND.3

VARIANT	STD_SOURCE
00	FLOATING POINT NUMBER
01	DOUBLE PRECISION NUMBER
10	INTEGER
11	UNDEFINED

OPERATION :

STD\_DESTINATION := ABSOLUTE VALUE (STD\_SOURCE)

\*\*\*\*\*  
 \* NEG \* (CHANGE SIGN)  
 \*\*\*\*\*

FORMAT:

	8 BITS	2 BITS	VARIABLES	VARIABLES
I	2692	1	VARIANT	STD_SOURCE   STD_DESTINATION
	OPCODE	OPERAND.1	OPERAND.2	OPERAND.3

VARIANT	STD_SOURCE
00	REAL
01	DOUBLE PRECISION
10	INTEGER
11	UNUSED

OPERATION:

STD\_DESTINATION := - (STD\_SOURCE)

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### SUBSCRIPT VALUE COMPUTATION S-OPS

\*\*\*\*\*

\* VD \* (VALIDATE DESCRIPTOR)

\*\*\*\*\*

FORMAT:

8 BITS	20 BITS	VARIES	VARIES
I 28A2	I MEM_ADDR	I STD_SOURCE	I STD_SOURCE I
OPCODE	OP.1	ELEMENT LENGTH DECLARED	ARRAY SIZE DECLARED

MEM\_ADDR IS THE BLOCK-RELATIVE ADDRESS OF THE DESCRIPTOR OF THE ARGUMENT PASSED. THIS DESCRIPTOR DESCRIBES WHERE THE ACTUAL ARGUMENT IS, ITS ELEMENT LENGTH PASSED AND THE ACTUAL SIZE PASSED.

"ELEMENT LENGTH DECLARED" EQL 0 IMPLIES ASSUMED-LENGTH.

"ARRAY SIZE DECLARED" IS THE NUMBER OF ELEMENTS PASSED IF NUMERIC ARRAY, OR THE NUMBER OF BYTES PASSED IF CHARACTER ARRAY, OR ZERO IF CHARACTER VARIABLE. IT WILL POINT TO THE THE SIZE FIELD OF THE RUN-TIME DIMENSION TABLE BUILT BY B4T/B4AT IF ASSUMED SIZE OR ADJUSTABLE SIZE. OTHERWISE IT WILL BE A LITERAL.

OPERATION:

- + VERIFY THAT THE SIZE OF THE ARRAY DECLARED IS LESS THAN OR EQUAL TO THE SIZE OF THE ARRAY PASSED.
- + MODIFY THE DESCRIPTOR AT MEM\_ADDR TO REFLECT THE DECLARED ELEMENT LENGTH.

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*

\* BAT \* (BUILD ARRAY TABLE)

\*\*\*\*\*

FORMAT:

8 BITS	4	VARIES		VARIES	
I 2882	I CNT	I STD_DESTINATION	I STD_SOURCE	I DIM_1	I...I DIM_CNT I
OPCODE		TABLE		LENGTH	

CNT ::= NUMBER OF DIMENSIONS

FORMAT OF DIM\_#::

VARIES		VARIES	
I STD_SOURCE	I STD_SOURCE	I	I
LOWER BOUND		UPPER BOUND	

OPERATION:

CONSTRUCT AN ARRAY DESCRIPTOR USING THE REMAINING INFO  
 OF THE FORMAT:

OFFSET	BIT(32) SEE 1 BELOW
DIM_INFO_1	RECORD1
DIM_INFO_2	RECORD1
...	
DIM_INFO_CNT	RECORD1
SIZE	BIT(32) SEE 2 BELOW

WHERE

$$A_1 = 1$$

$$A_{\#} = A_{(\#-1)} * D_{(\#-1)}$$

$$D_{\#} = (1 + \text{UPPER\_BOUND}_{\#} - \text{LOWER\_BOUND}_{\#})$$

RECORD1 FORMAT:

A_{\#}	BIT(32)
LOWER_BOUND	BIT(32)
UPPER_BOUND	BIT(32)

1. OFFSET =  $-(\text{LOWER\_BOUND}_1 * A_1 + \dots + \text{LOWER\_BOUND\_CNT} * A_{\text{CNT}})$
2. SIZE =  $A_{(\text{CNT}+1)} * \text{LENGTH}$  (OR WHAT IS PASSED IN IF ASSUMED-SIZE)
3. UPPER\_BOUND\_CNT IS REPLACED BY SIZE/LENGTH

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*

\* BAAT \* (BUILD ASSUMED SIZE ARRAY TABLE)

\*\*\*\*\*

FORMAT:

8 BITS	4	VARIES	20 BITS	VARIES		
I 28C2	I CNT	I STD_DEST.	I MEM_ADDR	I STD_SOURCE	I DIM1	I...I DIM_CNT
TABLE				LENGTH		

CNT = NUMBER OF DIMENSIONS.

FORMAT OF DIMENSION INFORMATION:

VARIES	VARIES
I STD_SOURCE	I STD_SOURCE
LOWER BOUND	UPPER BOUND

OPERATION:

SAME AS "BAT" EXCEPT:

1. MEM\_ADDR IS USED TO FETCH EITHER ASSUMED LENGTH AND/OR ASSUMED SIZE,
2. IF LENGTH = 0, USE ASSUMED LENGTH,
3. IF DIMENSION COUNT OF UPPER BOUND = 0, USE ASSUMED SIZE.

\*\*\*\*\*

\* CS \* (COMPUTE SUBSCRIPT VALUE)

\*\*\*\*\*

FORMAT:

8 BITS	4	VA	20	VA	...	VA	20	VA
I 28C2	I CNT	I S1	I M1	I S2	...	I S_CNT	I M_CNT	I STD_DEST.

CNT ::= NUMBER OF SUBSCRIPTS PRESENT

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

S# ::= STD\_SOURCE (SUBSCRIPTS)

M# ::= SCALE FACTOR

OPERATION:

STD\_DESTINATION := S1 \* M1 + S2 \* M2 + ... + S\_CNT \* M\_CNT

\*\*\*\*\*  
 \* CSB \* (CS WITH BOUNDS CHECKING)  
 \*\*\*\*\*

FORMAT:

8 BITS	4	VARIES				
I 28E2	I CNT	I S1	I S2	...	I S_CNT	I STD_DESTINATION
OPCODE						

S# FORMAT:

VARIES	VARIES	VARIES	20
I STD_SOURCE	I STD_SOURCE	I STD_SOURCE	I M#
SUBSCRIPT	LOWER BOUND	UPPER BOUND	SCALE

OPERATION:

SAME AS FOR "CS" EXCEPT THAT THE SUBSCRIPT IS CHECKED TO VERIFY THAT (LOWER BOUND IS LEQ SUBSCRIPT LEQ UPPER BOUND) BEFORE THE INDEX VALUE IS COMPUTED.

STD\_DESTINATION := S1 \* M1 + ... + S\_CNT \* M\_CNT

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B170C FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

\*\*\*\*\*

\* CSV \* (COMPUTE SUBSCRIPT VALUE WITH ARRAY TABLE, CHECK BOUNDS)

\*\*\*\*\*

FORMAT:

8 BITS	4	VA	VA	VA	...	VA	VA
I 28F2	I CNT	I STD_DEST1	I S1	I S2	I ...	I S_CNT	I STD_DEST2

CNT ::= 4 BIT LITERAL INDICATING NUMBER OF DIMENSIONS

S# ::= STD\_SOURCE (SUBSCRIPTS)

STD\_DEST1 IS ADDRESS OF ARRAY DESCRIPTOR AS DESCRIBED BY "BAT".

OPERATION:

STD\_DEST2 := OFFSET + ( S1 \* A\_1 + S2 \* A\_2 + ... + S\_CNT \* A\_CNT

SUBSCRIPT BOUNDS CHECKING IS ALWAYS PERFORMED.

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

DO-LOOP MAINTENANCE

\*\*\*\*\*  
 \* DO.UP \* (DO LOOP UPDATE)  
 \*\*\*\*\*

FORMAT:

8 BITS	3 BITS	VAR	VAR	VAR	VAR
OPCODE	VARIANT	OP.1	OP.2	OP.3	OP.4

VARIANT	OP.2	OP.3	OP.4	OP.5
0	STD_INDEX	STD_DEST.	STD_SOURCE	CODE_ADDRESS
1	STD_INDEX	STD_SOURCE	CCODE_ADDR.	STD_DESTINATION
2	STD_INDEX	STD_DEST.	STD_SOURCE	CODE_ADDRESS
3	STD_INDEX	STD_DEST1	CCODE_ADDR.	STD_DEST2
4	-	STD_SOURCE	CCODE_ADDR.	-
5	-	STD_DEST	CCODE_ADDR.	-

VAR	DO-TYPE	LOC	FUNCTION
0	DO 2 I=N,M	TOP	IF (STD_INDEX GT STD_SOURCE) THEN STD_DESTINATION = STD_INDEX GO TO CODE_ADDRESS END IF
1	"	BTM	STD_INDEX = STD_INDEX + 1 IF (STD_INDEX LE STD_SOURCE) * GO TO CODE_ADDRESS STD_DESTINATION = STD_INDEX
2	DO 2 I=N,M,L % STD_SOURCE=ITER_CNT % REG = CTR_VAR	TOP	IF (STD_SOURCE LE 0) THEN STD_DESTINATION = STD_INDEX GO TO CODE_ADDRESS END IF



BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

```

3      "
      % STD_INDEX=CTR_VAR
      % STD_DEST1(0)=INCR
      % STD_DEST1(1)=ITER_CNT
      % STD_DEST2=CTR_VAR
      BTM   STD_INDEX = STD_INDEX + STD_DEST1(0)
           STD_DEST1(1) = STD_DEST1(1) - 1
           IF (STD_DEST1(1) GT 0)
           * GO TO CODE_ADDRESS
           STD_DEST2 = STD_INDEX

4      DO 2 X=A,B,C
      % STD_SOURCE=ITER_CNT
      TOP  IF (STD_SOURCE LE 0)
           * GO TO CODE_ADDRESS

5      "
      % STD_DEST. =ITER_CNT
      BTM % CTR_VAR INCREMENTED SEPARATELY
           STD_DESTINATION = STD_DEST. - 1
           IF (STD_DEST. GT 0) GO TO CODE_ADDR.

6          NOT USED

7          NOT USED

```

NOTES: THE TOP "DOUP" S-OP IS GENERATED ONLY IF THE COMPILER CANNOT DETERMINE WHETHER OR NOT THE DO LOOP WILL BE EXECUTED AT LEAST ONCE.

STD\_DEST1 IN THE FORM OF VARIANT 3 IS A 2-ELEMENT ARRAY.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### CHARACTER TYPE S-OPS

\*\*\*\*\*

\* CREL \* (CHARACTER RELATION)

\*\*\*\*\*

#### FORMAT:

8 BITS	VARIES		3 BIT	VARIES
I 25E2	I STD_CHAR_SOURCE	I STD_CHAR_SOURCE	MSKI	STD_DESTINATION I
OPCODE	OPERAND.1	OPERAND.2	OP.3	OPERAND.4

#### OPERATION:

STD\_DESTINATION := TRUE IF OPERAND.1 RELATES TO OPERAND.2  
 IN ANY OF THE WAYS SPECIFIED BY THE MASK,  
 OTHERWISE FALSE.

FOR INTERPRETATION OF THE MASK, SEE "CRIF" (OF 25F2).

RIGHT BLANK FILL IS ASSUMED.

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* CRIF \* (CHARACTER RELATIONAL IF)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES		3 BIT	24 BITS
I 25F2	I STD_CHAR_SOURCE	I STD_CHAR_SOURCE	I MSK	I CODE_ADDRESS
OPCODE	OPERAND.1	OPERAND.2	OP.3	OPERAND.4

OPERATION:

FOR A MASK OF	THEN BRANCH TO CODE.ADDRESS IF
1	OPERAND.1 .GT. OPERAND.2
2	OPERAND.1 .LT. OPERAND.2
3	OPERAND.1 .NE. OPERAND.2
4	OPERAND.1 .EQ. OPERAND.2
5	OPERAND.1 .GE. OPERAND.2
6	OPERAND.1 .LE. OPERAND.2

RIGHT BLANK FILL IS ASSUMED.

\*\*\*\*\*  
 \* LEN \* (LENGTH)  
 \*\*\*\*\*

FORMAT :

8 BITS	VARIES	
I 26C2	I STD_CHAR_SOURCE	I STD.DESTINATION
OPCODE	OPERAND.1	OPERAND.2

OPERATION :

STD\_DESTINATION := CHAR\_LENGTH (STD\_CHAR\_SOURCE)

LENGTH OF STD\_CHAR\_SOURCE IS RETURNED IN BYTES.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (F)

\*\*\*\*\*  
 \* SSTR \* (SUBSTRING DESCRIPTOR)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES	VARIES
1 2762	1 STD_CHAR_DEST1	1 STD_SOURCE1	1 STD_SOURCE2	1 LAST_OP 1

WHERE LAST\_OP COULD BE

EITHER

VARIES
1 STD_CHAR_DEST2 1

OR

3 BITS	VARIES
1 272	1 STD_DESTINATION 1

OPERATION :

IF LAST\_OP IS OF STD\_CHAR\_DESTINATION FORM THEN  
 STD\_CHAR\_DEST2 = STD\_CHAR\_DEST1 (STD\_SOURCE1 : STD\_SOURCE2)

OTHERWISE ,

STD\_DESTINATION = CHARACTER\_DESCRIPTOR OF  
 ( STD\_CHAR\_DEST1 (STD\_SOURCE1 : STD\_SOURCE2) )

RIGHT TRUNCATION OR BLANK FILL WILL OCCUR.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 222 2749 (B)

\*\*\*\*\*  
 \* SSTL \* (SUBSTRING MOVE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES	VARIES
I 2772	I STD_CHAR_SOURCE	I STD_CHAR_DEST.	I STD_SOURCE1	I STD_SOURCE2

OPERATION:

STD\_CHAR\_DEST.(STD\_SOURCE1 : STD\_SOURCE2) = STD\_CHAR\_SOURCE

RIGHT TRUNCATION OR BLANK FILL WILL OCCUR.

\*\*\*\*\*  
 \* STC \* (STORE CHARACTER)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES	VARIES
I 2782	I STD_CHAR_SOURCE	I STD_CHAR_DEST.	I STD_DEST.	I STD_SOURCE
OPCODE	OPERAND.1	OPERAND.2	OP.3	OPERAND.4

OPERATION:

STD\_CHAR\_DEST(STD\_DEST : STD\_DEST+STD\_SOURCE) =  
 STD\_CHAR\_SOURCE//STD\_CHAR\_SOURCE//STD\_CHAR\_SOURCE//...

STD\_DESTINATION := STD\_DESTINATION + STD\_SOURCE

ACTION WILL AUTO STOP IF LEN(STD\_CHAR\_DESTINATION) IS REACHED.

BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
F.S. 2222 2749 (E)

\*\*\*\*\*  
\* CATD \* (CHARACTER CONCATENATION WITH DESCRIPTOR)  
\*\*\*\*\*

FORMAT:

8 BITS	VARIES		VARIES		VARIES		VARIES	
I 2792	I	STD_CHAR_SRC1	I	STD_CHAR_SRC2	I	STD_CHAR_DEST	I	STD_DEST
OPCODE	OPERAND.1		OPERAND.2		OP.3		OP.4	

OPERATION:

STD\_CHAR\_DESTINATION = STD\_CHAR\_SRC1 // STD\_CHAR\_SRC2  
AND STD\_DESTINATION = CHARACTER\_POINTER\_ADDR DESCRIPTOR OF  
( STD\_CHAR\_DEST, MIN ( LEN(STD\_CHAR\_DEST),  
LEN(STD\_CHAR\_SRC1) + LEN(STD\_CHAR\_SRC2) ) )

RIGHT TRUNCATION OR BLANK FILL WILL OCCUR.

\*\*\*\*\*  
\* MVC \* (MOVE CHARACTERS)  
\*\*\*\*\*

FORMAT:

8 BITS	VARIES		VARIES	
I 27A2	I	STD_CHAR_SOURCE	I	STD_CHAR_DESTINATION
OPCODE	OPERAND.1		OPERAND.2	

OPERATION:

STD\_CHAR\_DESTINATION := STD\_CHAR\_SOURCE  
RIGHT TRUNCATION OR BLANK FILL AS NEEDED.

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 81860/81700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*

\* CAT \* (CHARACTER CONCATENATION)

\*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES
I 27B2	I STD_CHAR_SOURCE1	I STD_CHAR_SRC2	I STD_CHAR_DEST I
OPCCDE	OPERAND.1	OPERAND.2	OP.3

OPERATION:

STD\_CHAR\_DESTINATION = STD\_CHAR\_SOURCE1 // STD\_CHAR\_SRC2

RIGHT TRUNCATION OR BLANK FILL WILL OCCUR.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### SUBROUTINE LINKAGE S-OPS

\*\*\*\*\*

\* SAVE \* (SAVE REGISTERS)

\*\*\*\*\*

FORMAT:

	8 BITS	1	4 BITS		VARIES
I	26D2	I V	I NUMBER_TO_SAVE	I	STD_DESTINATION I

OPERATION:

IF V = 0 SAVE NDX\_REGS #6 THRU #(1+NUMBER\_TO\_SAVE) AT STD\_DEST  
 IF V = 1 RESTORE NDX\_REGS #6 THRU #(1+NUMBER\_TO\_SAVE) FROM STD\_DEST

\*\*\*\*\*

\* LB \* (LOCAL BASE)

\*\*\*\*\*

FORMAT:

	8 BITS	1 BIT	10 BITS		VARIES
I	26E2	I 0	I TV_INDEX	I	STD_DESTINATION I

OR

	8 BITS	1 BIT	20 BITS		VARIES
I	26E2	I 1	I BYTE OFFSET	I	STD_DESTINATION I

OPERATION:

IF VARIANT = 0 THEN,



EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

STD\_DESTINATION := BASE RELATIVE ADDRESS OF LOCAL DATA BLOCK  
 FOR ENTRY(TV\_INDEX).

IF VARIANT = 1 THEN,

STD\_DESTINATION := BASE RELATIVE ADDRESS OF LOCAL DATA BLOCK  
 POINTED AT BY CODE ADDRESS LOCATED  
 AT GIVEN BYTE OFFSET IN CURRENT LOCAL  
 DATA BLOCK.

\*\*\*\*\*  
 \* PASS \* (PASS ACTUAL ARGUMENT)  
 \*\*\*\*\*

FORMAT:

	8 BITS	4	VARIABLES	VARIABLES
I	26FB	I V I	SOURCE I	REGISTER_OFFSET I

VAR	SOURCE	VALUE PASSED
0	STD_DESTINATION	WORD ADDRESS
1	STD_CHAR_DESTINATION	CHARACTER_DESCRIPTOR
2	CODE_ADDRESS	CODE ADDRESS (SEE BELOW)
3	STD_SRC, STD_DEST	ARRAY_DESCRIPTOR (SINGLE WORD)
4	STD_SRC, STD_DEST	ARRAY_DESCRIPTOR (DOUBLE WORD)
5	STD_SRC, STD_CHAR_DEST	CHARACTER ARRAY_DESCRIPTOR
6	STD_DEST, STD_SRC	ASSIGNED FORMAT (ARRAY_DESC_PASSED)
7	STD_SRC, STD_CHAR_DEST, STD_SRC, STD_SRC	CHARACTER ARRAY_DESCRIPTOR (USED FOR SUBSTRING)
8-F	UNUSED	

OPERATION:

A DESCRIPTOR IS CONSTRUCTED AND WRITTEN TO THE  
 LOCATION DESCRIBED BY THE REGISTER\_OFFSET. FURTHER  
 INFORMATION ABOUT THE SPECIFICS OF THIS OPERATION ARE  
 DESCRIBED BELOW.

- 1) WHEN THE VARIANT EQUALS 2, THE SPECIFIED CODE  
 ADDRESS IS OF THE FORM:

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B12C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

```

  1 BIT      20 BITS
  -----
  |  0      | PEP_ADDR |
  -----

```

OR

```

  1 BIT      10 BITS
  -----
  |  1      | TV_INDEX |
  -----

```

- 2) WHEN THE VARIANT EQUALS 3 OR 4, THE STANDARD SOURCE OPERAND IS THE ARRAY SIZE AND THE STANDARD DESTINATION OPERAND IS THE FIRST ARRAY ELEMENT.
- 3) WHEN THE VARIANT EQUALS 6, THE STANDARD DESTINATION OPERAND POINTS TO THE ASSIGNED FORMAT TABLE (CONSISTING OF N ROWS AND 2 COLUMNS). REFER TO THE FORMAT SECTION AT THE BEGINNING OF THIS DOCUMENT FOR A DESCRIPTION OF THE ASSIGNED FORMAT TABLE.

THE FOLLOWING ERROR CHECKING WILL BE DONE:

- A. IF THE MOST SIGNIFICANT BIT OF THE STANDARD SOURCE OPERAND IS ZERO, THEN THE VARIABLE IS NOT ASSIGNED A LABEL AND AN ERROR IS GIVEN.
- B. IF THE VALUE OF SUBBIT(STD\_SRC,1) IS GREATER THAN THE VALUE OF THE FIRST TABLE ENTRY, THEN THE ASSIGNED LABEL IS NOT IN THE TABLE AND AN ERROR IS GIVEN.
- C. IF THE ZERO-TH BIT OF THE TABLE ENTRY WHOSE ROW EQUALS THE STD\_SRC IN THE FIRST COLUMN OF THE TABLE EQUALS 0, THEN THE LABEL IS NOT A FORMAT LABEL AND AN ERROR IS GIVEN.

IF NO ERROR IS DETECTED, THE ARRAY DESCRIPTOR PASSED IS CONSTRUCTED WHICH POINTS TO THE FORMAT ADDRESS FOUND IN THE ROW EQUAL TO THE STANDARD SOURCE OPERAND IN COLUMN 2 OF THE ASSIGNED FORMAT TABLE. THE DESCRIPTOR PASSED WILL REFLECT 255 ELEMENTS AND A LENGTH EQUAL TO ZERO. THE FORMAT ADDRESS POINTED TO BY THE DESCRIPTOR IS THE LOCAL BASE RELATIVE BYTE ADDRESS OF THE FIRST ARRAY ELEMENT.

- 4) WHEN THE VARIANT EQUALS 7, THE FIRST STANDARD

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1200/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

SOURCE OPERAND IS THE ARRAY SIZE.  
 STD\_CHAR\_DEST(SECOND STD\_SRC:THIRD STD\_SRC) IS THE  
 FIRST ARRAY ELEMENT. THE SECOND STANDARD SOURCE  
 OPERAND IS THE FIRST CHARACTER AND THE THIRD  
 STANDARD SOURCE OPERAND IS THE LAST CHARACTER OF  
 THE STANDARD CHARACTER DESTINATION.

5) THE REGISTER\_OFFSET HAS ONE OF TWO FORMATS:

1 BIT 4 BITS 10 BITS

-----  
 | 0 | REGISTER\_NUMBER | BYTE OFFSET INTO FOREIGN BLOCK |  
 -----

OR

1 BIT 19 BITS 10 BITS

-----  
 | 1 | BYTE OFFSET INTO | BYTE OFFSET INTO FOREIGN BLOCK |  
 | 1 | LOCAL DATA BLOCK |  
 -----

THE BASE-RELATIVE BIT ADDRESS (THE INDEX) ADDED TO THE  
 BYTE OFFSET (DIRECT\_ADDR) IS THE ADDRESS OF WHERE TO  
 PUT THE 48-BIT DESCRIPTOR.

ALL ARGUMENTS ARE PASSED BY ADDRESS, EXCEPT CODE  
 ADDRESSES. IF AN ARGUMENT IS PASSED IN THE USER  
 PROGRAM BY VALUE, A COPY IS MADE AND THE ADDRESS OF  
 THE COPY IS PASSED AS THE ARGUMENT.

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*

\* CALL \* (SUBROUTINE CALL)

\*\*\*\*\*

FORMAT:

	8 BITS	10 BITS
I	270a	I TV_INDEX I

OPERATION:

BRANCH TO EXTERNAL CODE ADDRESS. SUBPROGRAM ADDRESS IS A  
 10-BIT TRANSFER VECTOR INDEX.

\*\*\*\*\*

\* DCAL \* (DYNAMIC SUBROUTINE CALL)

\*\*\*\*\*

FORMAT:

	8 BITS	VARIES
I	271a	I STD_DESTINATION I

OPERATION:

SAME AS CALL EXCEPT BRANCH TO CODE\_SEGMENT\_DISPLACEMENT  
 FOUND AT ADDRESS CONTAINED IN THE STANDARD DESTINATION.

BLRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* RTA \* (RETURN)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS
-----
I  2722  I
-----
  
```

OPERATION:

NEXT\_INST\_PTR := RETURN ADDRESS (FROM MEMORY LOCATION "0"  
 IN LOCAL DATA BLOCK).

RETURNS TO ADDRESS SPECIFIED IN LOCAL DATA BLOCK.

\*\*\*\*\*  
 \* SFCL \* (STATEMENT FUNCTION CALL)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS   1       24 BITS           VARIES
-----
I  2732  I  0  I CODE_ADDRESS I STD_DESTINATION I
-----
  
```

OR

```

      8 BITS   1       VARIES           VARIES
-----
I  2732  I  1  I STD_DESTINATION I STD_DESTINATION I
-----
  
```

WHEN THE ONE BIT VARIANT = 1, THE FIRST STANDARD DESTINATION  
 CONTAINS AN EXTERNAL CODE ADDRESS.

OPERATION:

STORE NEXT\_INST\_PTR AS EXTERNAL\_CODE\_ADDRESS AT  
 STD\_DESTINATION (THE FIRST OPERAND FOLLOWING THE VARIANT),  
 THEN BRANCH TO SPECIFIED CODE ADDRESS (WHEN VARIANT = 0)  
 OR TO EXTERNAL CODE ADDRESS CONTAINED IN FIRST STD\_DESTINATION  
 (WHEN VARIANT = 1).

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

\*\*\*\*\*  
 \* SFRTN \* (STATEMENT FUNCTION RETURN)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	
I 2742	I	STD_DESTINATION I

OPERATION:

BRANCH TO EXTERNAL\_CODE\_ADDRESS STORED IN STD\_DESTINATION.

\*\*\*\*\*  
 \* SPAM \* (SCRAMBLE AND PROVIDE ARGUMENTS)  
 \*\*\*\*\*

FORMAT:

8 BITS	8 BITS	8 BITS	VARIES
I 2752	I	#PARMS_TO_ADD I	#PARMS_PASSED I ORDER_INFO I

OPERATION:

#PARMS\_PASSED ARE COPIED TO AN INTERPRETER IMPLIED LOCATION. (#PARMS\_PASSED \* #PARMS\_TO\_ADD) 48-BIT DESCRIPTORS ARE ZEROED OUT. #PARMS\_PASSED ARE COPIED BACK AS SPECIFIED BY ORDER\_INFO.

ORDER\_INFO IS AN ARRAY OF #PARMS\_PASSED ELEMENTS; EACH ELEMENT IS BIT(20) AND CONTAINS ADDRESSES WHICH ARE THE LOCATIONS TO WHERE THE 48-BIT ARGUMENT DESCRIPTORS ARE TO BE WRITTEN. LENGTH OF ORDER\_INFO IS (#PARMS\_PASSED \* 20).

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### SPECIAL FUNCTION S-OPS

\*\*\*\*\*

\* STPN \* (FORTRAN STATEMENT NUMBER)

\*\*\*\*\*

FORMAT:

```

      8 BITS    14 BITS
-----
 1  2912    1 LITERAL 1
-----
  
```

OPERATION:

MOVE THE 14-BIT LITERAL TO THE FORTRAN STATEMENT NUMBER  
 FIELD IN THE BASE-LIMIT AREA.

DEBUG INTERPRETERS WILL ALSO CHECK FOR BREAKPOINT MATCHING.

\*\*\*\*\*

\* HMON \* (HARDWARE MONITOR)

\*\*\*\*\*

\*\*\* NOT YET IMPLEMENTED \*\*\*

FORMAT:

```

      8 BITS          VARIES
-----
 1  2922    1 STD_CHAR_SOURCE 1
-----
  OFCODE
  
```

OPERATION:

ISSUE EACH BYTE OF STD\_CHAR\_SOURCE FROM LEFT TO  
 RIGHT AS THE ARGUMENT OF A HARDWARE MONITOR MICRO OP.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

### PRIVILEGED USER S-OPS

**N O T E :** ALL REFERENCES TO BUFF, OFFSET, SIZE, EZFLG, ETC.  
 (COMMON REFERENCES USED IN INTRINSICS) INDICATE  
 PREDEFINED LOCATIONS IN MEMORY. THESE VALUES ARE NOT  
 PASSED IN THE OP ITSELF.

\*\*\*\*\*

\* ADDR \* (BASE RELATIVE ADDRESS)

\*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES
I 2642	I STD_DESTINATION	I STD_DESTINATION
OPCODE	OP.1	OP.2

OPERATION:

OP.2 := ADDRESS( OP.1 ) - BASE\_REGISTER

\*\*\*\*\*

\* DESC \* (INDIRECT DESCRIPTOR OF NUMERIC ARRAYS)

\*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES	VARIABLES	VARIABLES
I 26B2	I STD_CHAR_DEST	I STD_SRC1	I STD_SRC2	I STD_DESTINATION
OPCODE	OP.1	OFFSET	LENGTH	OP.4

OPERATION:

STD\_DESTINATION := (SEGMENT, DISPLACEMENT, LENGTH)  
 DESCRIPTOR OF SUBSTR (STD\_CHAR\_DEST, OFFSET, LENGTH)



BUROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

NOTE : STD\_CHAR\_DEST DESCRIBES THE FIRST BYTE OF A NUMERIC  
 FIELD. NO CHECKS ARE MADE FOR VALID VALUES OF OFFSET OR  
 LENGTH.

\*\*\*\*\*  
 \* STC \* (STORE CHARACTER)  
 \*\*\*\*\*

OP CODE : 2782

STC IS ALSO A CHARACTER-TYPE S-OP. SEE OPCODE 2782 ABOVE.

\*\*\*\*\*  
 \* COMP \* (COMMUNICATE)  
 \*\*\*\*\*

FORMAT:

	8 BITS		VARIES		VARIES
	-----				
	27C2		STD_DESTINATION		STD_SOURCE
	-----				
	OPCODE				

OPERATION:

COMMUNICATE (TO THE MCP) THE INFORMATION WHICH IS  
 LEFT-JUSTIFIED IN STANDARD DESTINATION. STD\_SOURCE  
 INDICATES THE NUMBER OF BITS TO BE COMMUNICATED.

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* LCR \* (LOAD COMMUNICATE REPLY)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS          VARIES
-----
 I   27C2   I  STD_DESTINATION I
-----
      0PCCDE
  
```

OPERATION:

STD\_DESTINATION := LAST 24 BITS OF THE COMMUNICATE REPLY.  
 IT IS EXPECTED THAT STD\_DESTINATION IS OF TYPE INTEGER.

\*\*\*\*\*  
 \* TIME \* (PROCESSOR TIME)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS          VARIES
-----
 I   27E2   I  STD_DESTINATION I
-----
  
```

OPERATION:

STD\_DESTINATION := PROCESSOR\_TIME.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* DS \* (DISCONTINUE JOB)  
 \*\*\*\*\*

FORMAT:

8 EITS	VARIES	
I 27F2	I	STD_CHAR_DESTINATION I
OPCODE		

OPERATION:

CAUSE HOST TO BE ABORTED GIVING THE MESSAGE CONTAINED  
 AT STD\_CHAR\_DESTINATION.

\*\*\*\*\*  
 \* WID \* (WRITE INTEGER DIGITS)  
 \*\*\*\*\*

FORMAT:

8 EITS	VARIES	VARIES	
I 2802	I	STD_SOURCE I	STD_SOURCE I
OPCODE	OP1	OP2	

OPERATION :

WRITES OP1 TO PREDEFINED MEMORY LOCATION THE VALUE:  
 DECIMAL(OP1) FOR THE LENGTH SPECIFIED IN OP2 (IN  
 BYTES).

LEFT TRUNCATION OR "0" FILL OCCURS AS NEEDED.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

\*\*\*\*\*  
 \* BNRY \* (BINARY CONVERSION)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS          VARIES
-----
I  2812  I STD_DESTINATION I
-----
  
```

OPERATION:

WRITE TO THE LOCATION SPECIFIED BY THE STANDARD DESTINATION THE VALUE WHICH IS THE SUBSTR OF (BUFF, OFFSET BY BFRPTR, FOR THE LENGTH SPECIFIED BY SIZE).

BINARY CONVERSION WILL STOP WHEN A NON-NUMERIC-AND-NON-BLANK IS ENCOUNTERED. THE OFFSET AND SIZE LOCATIONS ARE UPDATED TO REFLECT THE INDEX AND LENGTH OF THE REMAINING SUB-STRING.

IF BZFLG .TRUE. THEN BLANKS WILL BE TREATED AS ZEROES, OTHERWISE BLANKS WILL BE TREATED AS NULLS.

\*\*\*\*\*  
 \* SIGN \* (SEARCH FOR SIGN)  
 \*\*\*\*\*

FORMAT:

```

      8 BITS          VARIES
-----
I  2822  I STD_DESTINATION I
-----
  
```

OPERATION:

REDUCE SUBSTR (BUFF, OFFSET, SIZE) UNTIL FIRST NEG " "  
 (BY ADJUSTING OFFSET AND SIZE).  
 ON EGS DO; STD\_DESTINATION := .FALSE.; RETURN; END;

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

STD\_DESTINATION := (FIRST CHAR) .EQ. "-" ;

IF (FIRST CHAR) = "+" OR "-" THEN REDUCE THE ADJUSTED SUB-STRING  
 TO SKIP THE SIGN.

STD\_DESTINATION WILL REFLECT WHETHER OR NOT A MINUS SIGN IS FOUND.

\*\*\*\*\*

\* REAL \* (GET REAL VALUE)

\*\*\*\*\*

FORMAT:

8 BITS	1 BIT	VARIES	
-----			
I 2832	I VARIANT	I STD_DESTINATION	I STD_DESTINATION I
-----			
		OP1	OP2

OPERATION:

OP1 := FLOAT THE BINARY VALUE OF  
 SUBSTR (BUFF, OFFSET, SIZE)

OP2 := NUMBER OF DECIMAL DIGITS FOUND FOLLOWING THE DECIMAL POINT

REAL VALUE CONVERSION WILL STOP WHEN IT ENCOUNTERS A CHARACTER  
 WHICH IS NONE OF THE FOLLOWING : A DECIMAL DIGIT, A BLANK, A  
 DECIMAL POINT. OFFSET AND SIZE ARE UPDATED TO REFLECT  
 THE INDEX AND LENGTH OF THE REMAINING SUB-STRING.

IF BZFLG IS .TRUE. THEN BLANKS WILL BE TREATED AS ZEROES,  
 OTHERWISE BLANKS WILL BE TREATED AS NULLS.

IF VARIANT = 2(1)12 THEN STD\_DESTINATION IS DOUBLE PRECISION  
 ELSE STD\_DESTINATION IS SINGLE PRECISION

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (E)

\*\*\*\*\*  
 \* NEXT \* (EXAMINE NEXT CHARACTER)  
 \*\*\*\*\*

FORMAT:

8 BITS	8 BITS	24 BITS
-----		
2842	LITERAL	CODE_ADDRESS
-----		
OPCODE	OP.1	CP.2

OPERATION:

```

IF (SIZE .EQ. 0) THEN RETURN ;
IF SUBSTR (BUFF,OFFSET,1) .EQ. OP.1
  THEN DO ;
      BUMP OFFSET BY 1 ;
      DECREMENT SIZE BY 1 ;
      BRANCH TO CODE_ADDRESS ;
  END ;

```

SIZE IS THE LENGTH OF THE SUBSTRING REPRESENTED BY BUFF.

\*\*\*\*\*  
 \* WEF \* (WRITE E-FORMAT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES
-----	
2852	STD_SOURCE
-----	

OPERATION:

WRITE TO A PREDEFINED MEMORY LOCATION IN E-FORMAT  
 THE VALUE CONTAINED IN THE STANDARD SOURCE.

EURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (8)

\*\*\*\*\*  
 \* WFF \* (WRITE F-FORMAT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	
I 2862 I	STD_SOURCE	I

OPERATION:

WRITE TO A PREDEFINED MEMORY LOCATION IN F-FORMAT  
 THE VALUE CONTAINED IN THE STANDARD SOURCE.

\*\*\*\*\*  
 \* FANC \* (FETCH AND CLEAR ERROR CONDITION)  
 \*\*\*\*\*

8 BITS	3 BITS	VARIES	
I 2872 I	MASK	I	STD_DESTINATION I

OPERATION:

WRITE TO STD\_DESTINATION THE VALUE:  
 MASK .AND. ERROR\_CONDITION\_INFO

ERROR\_CONDITION\_INFO := (.NOT. MASK) .AND. ERROR\_CONDITION\_INF

WHERE ERROR\_CONDITION\_INFO IS 3 BITS OF INFORMATION LOCATED AT  
 A BASE-RELATIVE ADDRESS. IT HAS THE SAME FORMAT AS THE MASK :

FIRST BIT : OVERFLOW CONDITION.  
 SECOND BIT : EXPONENT UNDERFLOW CONDITION.  
 THIRD BIT : DIVIDE-BY-ZERO CONDITION.

THE .AND. OPERATION IS DONE BIT BY BIT.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

\*\*\*\*\*  
 \* XTRACT \* (EXTRACT BITS)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES	VARIABLES	VARIABLES
1 2882	1 STD_DEST.	1 STD_SOURCE	1 STD_SOURCE	1 STD_DEST.
	OP.1	OP.2	CP.3	OP.4

OPERATION:

OP.4 := SUBBIT (OP.1, OP.2, OP.3)

\*\*\*\*\*  
 \* INSERT \* (INSERT BITS)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES	VARIABLES	VARIABLES
1 2892	1 STD_SOURCE	1 STD_DEST.	1 STD_SOURCE	1 STD_SOURCE
	OP.1	CP.2	CP.3	OP.4

OPERATION:

SUBBIT(OP.2, CP.3, CP.4) := OP.1



BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
P.S. 2222 2749 (B)

\*\*\*\*\*  
\* WIF \* (WRITE I-FORMAT)  
\*\*\*\*\*

FORMAT:

8 BITS      VARIES  
-----  
I 2932 I STD\_SOURCE I  
-----

OPERATION:

WRITE TO A PREDEFINED MEMORY LOCATION IN I-FORMAT  
THE VALUE CONTAINED IN THE STANDARD SOURCE.

N O T E : OPS 2942 - 2A22 MAY ALSO BE USED AS PRIVILEGED OPS.  
THEY ARE DESCRIBED UNDER THE SECTION TRIGONOMETRIC AND  
OTHER FUNCTIONS.

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (E)

### TRIGONOMETRIC AND OTHER FUNCTIONS

**N O T E:** ALL STD\_SOURCES AND STD\_DESTINATIONS ARE REAL.  
 ANY OF THESE OPS MAY ALSO BE USED AS PRIVILEGED OPS.

\*\*\*\*\*  
 \* AMCC \* (REMAINDER)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES	VARIES
I 2942	I STD_SOURCE	I STD_SOURCE	I STD_DESTINATION I
	OP1	OP2	

OPERATION:

STD\_DESTINATION := CP1 - (AINT(CP1/CP2) \* CP2)

\*\*\*\*\*  
 \* SIN \* (SINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 2952	I STD_SOURCE	I STD_DESTINATION I

OPERATION:

STD\_DESTINATION := SINE OF STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 P.S. 2222 2749 (B)

\*\*\*\*\*  
 \* COS \* (COSINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 2962	I STD_SOURCE	I STD_DESTINATION I

OPERATION:

STD\_DESTINATION := COSINE OF STD\_SOURCE

\*\*\*\*\*  
 \* TAN \* (TANGENT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 2972	I STD_SOURCE	I STD_DESTINATION I

OPERATION:

STD\_DESTINATION := TANGENT OF STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* SINH \* (HYPERBOLIC SINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 2982	I STD_SOURCE	I STD_DESTINATION I

OPERATION:

STD\_DESTINATION := HYPERBOLIC SINE OF STD\_SOURCE

\*\*\*\*\*  
 \* COSH \* (HYPERBOLIC COSINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 2992	I STD_SOURCE	I STD_DESTINATION I

OPERATION:

STD\_DESTINATION := HYPERBOLIC COSINE OF STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B18C0/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

\*\*\*\*\*  
 \* TANH \* (HYPERBOLIC TANGENT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES
I 29A2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := HYPERBOLIC TANGENT OF STD\_SOURCE

\*\*\*\*\*  
 \* ASIN \* (ARCSINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLES	VARIABLES
I 29B2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := ARCSINE OF STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (8)

\*\*\*\*\*  
 \* ACOS \* (ARCCOSINE)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 29C2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := ARCCOSINE OF STD\_SOURCE

\*\*\*\*\*  
 \* ATAN \* (ARCTANGENT)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
I 29D2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := ARCTANGENT OF STD\_SOURCE

BURROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 B1800/B1700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

\*\*\*\*\*  
 \* AINT \* (FLOOR)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLE	VARIABLE
I 09E2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := THE INTEGER WHOSE MAGNITUDE IS THE LARGEST  
 INTEGER WHICH DOES NOT EXCEED THE MAGNITUDE  
 OF STD\_SOURCE AND WHOSE SIGN IS THE SAME AS  
 THAT OF STD\_SOURCE.

IF  $-1 < \text{STD\_SOURCE} < 1$ , THEN  $\text{STD\_DESTINATION} := 0.0$

AS AN EXAMPLE,  $\text{AINT}(-3.7) = -3.0$

\*\*\*\*\*  
 \* ALOG \* (LOG TO BASE e)  
 \*\*\*\*\*

FORMAT:

8 BITS	VARIABLE	VARIABLE
I 09F2	I STD_SOURCE	I STD_DESTINATION

OPERATION:

STD\_DESTINATION := NATURAL LOG OF STD\_SOURCE

BURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1600/B1700 FORTRAN77 S-LANGUAGE  
F.S. 2222 2749 (B)

\*\*\*\*\*  
\* ALGG10 \* (LOG TO BASE 10)  
\*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
-----		
I 2A02	I STD_SOURCE	I STD_DESTINATION I
-----		

OPERATION:

STD\_DESTINATION := LOG TO BASE 10 OF STD\_SOURCE

\*\*\*\*\*  
\* SQRT \* (SQUARE ROOT)  
\*\*\*\*\*

FORMAT:

8 BITS	VARIES	VARIES
-----		
I 2A12	I STD_SOURCE	I STD_DESTINATION I
-----		

OPERATION:

STD\_DESTINATION := SQUARE ROOT OF STD\_SOURCE



EURROUGHS CORPORATION  
COMPUTER SYSTEMS GROUP  
SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
B1800/B1700 FORTRAN77 S-LANGUAGE  
F.S. 2222 2749 (B)

\*\*\*\*\*  
\* EXP \* (EXFONENTIAL)  
\*\*\*\*\*

FORMAT:

```
      8 BITS      VARIES      VARIES  
-----  
I  2A22  I STD_SOURCE I STD_DESTINATION I  
-----
```

OPERATION:

```
STD_DESTINATION := e ** STD_SOURCE
```

ELRROUGHS CORPORATION  
 COMPUTER SYSTEMS GROUP  
 SANTA BARBARA PLANT

COMPANY CONFIDENTIAL  
 81800/81700 FORTRAN77 S-LANGUAGE  
 F.S. 2222 2749 (B)

## INDEX

ADDRESSES 3-7  
 ALPHABETICAL LISTING 2-1  
 ARITHMETIC REPLACEMENT S-OPS 2-5, 4-1  
 ASSIGNED GOTO AND FORMAT TABLE 3-5

BASE-LIMIT MEMORY LAYOUT 1-1  
 BRANCH S-OPS 2-7, 6-1

CHARACTER TYPE S-OPS 2-8, 10-1

CG LOOP MAINTENANCE 2-8  
 CO-LOOP MAINTENANCE 9-1

ERROR CONDITION INFORMATION 3-1

FORMATS 3-1

INSTRUCTION SET 2-1

LAYOUT TABLE 3-3  
 LISTING BY NUMERIC OP-CODE 2-5  
 LOCAL DATA BLOCK 3-2  
 LOGICAL REPLACEMENT AND IF STATEMENT S-OPS 5-1  
 LOGICAL REPLACEMENT AND IF STATEMENT S-OPS 2-7

PRIVILEGED USER S-OPS 2-9, 13-1

REGISTERS 3-1  
 RUN-TIME DIMENSION TABLE 3-12

SPECIAL FUNCTION S-OPS 2-8, 12-1  
 STANDARD CHARACTER DESTINATION 3-12  
 STANDARD CHARACTER SOURCE 3-11  
 STANDARD DESTINATION 3-10  
 STANDARD INDEX 3-7  
 STANDARD SOURCE 3-10  
 SUBROUTINE LINKAGE MECHANISM 3-3  
 SUBROUTINE LINKAGE S-OPS 2-8, 11-1  
 SUBSCRIPT VALUE COMPUTATION S-OPS 2-7, 8-1

TRANSFER VECTOR 3-4  
 TRIGONOMETRIC AND OTHER FUNCTIONS 2-9, 14-1  
 TYPE AND SIGN CONVERSION S-OPS 2-7, 7-1

VALUES 3-2