

```

1 * GENERAL AUTOMATION, INC. ALL RIGHTS RESERVED
2 *****
3 *
4 * PROGRAM NAME FPH-22
5 *
6 * MODEL NUMBER 8F022
7 *
8 * PURPOSE FORTRAN PHASE-22
9 *
10 * PROGRAMMER DICK WALLMANN
11 *
12 ***** REVISION LIST *****
13 *
14 * RV DATE SCO BY REASON FOR CHANGE
15 * -----
16 *
17 * 01 11/16/70 NONE RPH INITIAL RELEASE
18 *
19 *****
20 *****
21 HDNG MPX FORTRAN ** STATEMENT ALLOCATION
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 * * ASSIGNS THE RELATIVE ADDRESSES TO STMT
27 * * FUNCTIONS AND NUMBERED STATEMENTS
28 * * INSERTS ALLOCATIONS INTO THE STRING
29 * * CREATES THE SUBROUTINE INITIALIZATION CALL-
30 * * .CALL SUBIN, ALONG WITH ITS DUMMY ARGUMENTS
31 * * GENERATES THE STMT FUNCTION RETURN CODE
32 *
33 *ENTRY POINTS-
34 * * NEO - PHASE 22 IS LOADED BY PHASE 21 VIA
35 * * THE ROLRX ROUTINE AND EXECUTION BEGUN
36 * * AT LOCATION LABELED NEO.
37 *
38 *INPUT-
39 * * STATEMENT STRING
40 * * SYMBOL TABLE
41 * * FCOM
42 *
43 *OUTPUT-
44 * * STATEMENT STRING
45 * * SYMBOL TABLE
46 * * FCOM
47 *
48 *EXTERNAL REFERENCES-
49 * * SUBROUTINES
50 * * ROLRX
51 * * OTHER FORTRAN PHASES
52 * * NONE (NO PRINTING IS DONE IN THIS PHASE)
53 *
54 *EXITS-
55 * * NORMAL-
56 * * PHASE 23 IS CALLED VIA ROLRX AND CONTROL
57 * * PASSED TO IT
58 * * ERRORS-
59 * * OVERLAP-

```

```

60 *          UPON DETECTING AN OVERLAP CONDITION
61 *          PHASE 23 IS CALLED VIA ROLRX.  IF THE
62 *          OVERLAP OCCURRED IN THIS PHASE THEN THE
63 *          ERROR WORD IS SET PRIOR TO EXITING PHASE
64 *          SYNTAX-
65 *          THERE ARE NO SYNTAX ERRORS DETECTED IN
66 *          THIS PHASE.
67 *
68 *TABLES/WORK AREAS-
69 * * FCOM
70 * * STRING AREA
71 * * SYMBOL TABLE AREA
72 *
73 * ATTRIBUTES-NONE
74 *
75 *NOTES-
76 * * ASF IS AN ABBREVIATION FOR ARITH STMT FUNC
77 * * COMMENTS START IN COL 44
78 * * THE SWITCHES USED BY THIS PHASE FOLLOW.
79 * * IF POSITIVE, THE SWITCH IS TRANSFER T
80 * * IF ZERO, THE SWITCH IS NORMAL N
81 *          TEMP5-
82 *          N NOT FIRST ASF STATEMENT
83 *          T ADDR OF FIRST ASF STATEMENT
84 *          ASFSW-
85 *          T POINTER NOT PAST ASF STATEMENTS
86 *          TR-
87 *          N SUBP INITIALIZE FOR ASF STATEMENTS
88 *          T 1 SUBP INITIALIZE FOR SUBP
89 *          T 2 STATEMENT ALLOCATION
90 *
91 *****
92          HDNG      MPX FORTRAN ** STATEMENT ALLOCATION
93          ABS      DEF CORE
94 *
95 *          SYSTEM AND FORTRAN EQUATES
96 *
97 MEMRY EQU      72EF CORE      MAXIMUM CORE SIZE
98 PHSIZ EQU      4*320      MAXIMUM PHASE SIZE
99 OVERL EQU      MEMRY-PHSIZ      PHASES 2-29 START
100 FCOM EQU      OVERL-22      FORTRAN COMM. TABLE
101 PHNTB EQU      FCOM-56      PHASE TABLE
102 ROLRX EQU      PHNTB-50      INTERPHASE CALL
103 AREA EQU      OVERL+3*320+100 PRINT DATA ADDRESS
104 PRINT EQU      AREA+1      PRINT ENTRANCE
105 *
106 *          FORTRAN COMMUNICATIONS AREA
107 *
108          ORG      FCOM
109 SOFS BSS      1      START OF STRING
110 EOFS BSS      1      END OF STRING
111 SOFST BSS      1      START OF STRING
112 SOFNS BSS      1      LENGTH OF PROGRAM
113 SOFXT BSS      1      SIZE OF WORK AREA (VARIES)
114 SOFGT BSS      1      SIZE OF CONSTANTS AREA
115 EOFST BSS      1      END OF SYMBOL TABLE
116 COMON BSS      1      RELATIVE ENTRY POINT
117 CSIZE BSS      1      SIZE OF COMMON
118 ERROR BSS      1      ERROR FLAG
119 *          BIT 15 OVERLAP ERROR

```

```

120 * BIT 14 OTHER ERROR
121 FNAME BSS 1 PROGRAM NAME
122 BSS 1 2ND WD OF NAME
123 SORF BSS 1 SUBR (-) OR FUNG (+)
124 CCWD BSS 1 CONTROL CARD WORD
125 * BIT 15 TRANSFER TRACE
126 * BIT 14 ARITHMETIC TRACE
127 * BIT 13 EXTENDED PRECISION
128 * BIT 12 LIST SYMBOL TABLE
129 * BIT 11 LIST SUBPROGRAM NAMES
130 * BIT 10 LIST SOURCE PROGRAM
131 * BIT 9 ONE WORD INTEGERS
132 IOCS BSS 1 IOCS CONTROL CARD WORD
133 *
134 * SEE PHASE ONE FOR BIT PATTERNS
135 *
136 *
137 DFCNT BSS 1 DEFINE FILE COUNT
138 *
139 LCOMN BSS 2 SIZE OF INSKEL COMMON
140 *
141 ICCER BSS 2 IOCS CONTROL CARD ERROR
142 *
143 BSS 2 SYSTEM LOADER USE
144 *
145 * END OF FORTRAN COMMUNICATION
146 * AREA
147 *****
148 HDNG MPX FORTRAN ** STATEMENT ALLOCATION
149 *
150 ORG OVERL
151 NEQ BSC L ENT BR TO INITIALIZATION
152 *
153 *
154 * TEST ROUTINE FOR SUBROUTINE
155 * IN CALL ARGUMENTS
156 *
157 SUBPR DC 0 SUBR ENTRY POIN
158 SLA 8 SHIFT INDICATOR BIT TO SIGN
159 BSC I SUBPR,- RETURN IF NOT ON
160 SLA 16 CLEAR ACC
161 BSI 3 OUTP-Z O/P 0 TO O/P STRING
162 MDX L LISTL,1 INCR LIST LENGTH
163 BSC I SUBPR RETURN TO CALLING PROGRAM
164 *
165 *
166 INCPT DC 0 LINK ENTRY POINT
167 MDX 1 1 INCR XR1 POINTER
168 MDX L NORM,-4 DECR STMT LENGTH COUNTER
169 SLA 0 NOP IN CASE OF SKIP
170 BSC I INCPT RETURN
171 *
172 *
173 H6200 DC /6200 CONSTANT
174 H7000 DC /7000 CONSTANT
175 H7800 DC /7800 CONSTANT
176 H0400 DC /0400 DUMMY VAR EXTRACTOR
177 H5E00 DC /5E00 GENERATED LABEL TEST CON
178 ZERO DC 0 CONSTANT
179 Z EQU ZERO

```

180	ONE	DC	/0001	CONSTANT
181	TWO	DC	/0002	CONSTANT
182	THREE	DC	/0003	CONSTANT
183	FOUR	DC	/0004	CONSTANT
184	NORM	DC	0	NORM COUNTER
185	H07FC	DC	/07FC	MASK FOR NORM
186	HF800	DC	/F800	MASK FOR STMT TYPE
187	H6000	DC	/6000	.FORMAT.-TYPE
188	HD000	DC	/D000	ARITH STMT FUNC STMT TYPE
189	LOCC1	DC	0	LOC CTR 1
190	LOCC2	DC	0	LOC CTR 2
191	H5F00	DC	/5F00	.BLANK. AT OUTPUT
192	DUMVC	DC	0	DUMMY VAR CTR
193	H1800	DC	/1800	DIMENSION BIT MASK SYM TBL
194	TR	DC	0	TRACKSWITCH
195	DUMVP	DC	0	DUMMY VAR PT IN SYM TBL
196	H6008	DC	/6008	MASK
197	SUBIN	DC	/1000	CALL SUBIN
198	STADR	DC	0	SYM TBL ADDR COUNTER
199		DC	/8000	MASK
200	TESTN	DC	0	.TESTNAME.
201	TINP	DC	0	TEMP I/P POINTER
202	LISTL	DC	0	LIST LENGTH
203	OUTID	DC	0	ADDR OF O/P ID
204	HF803	DC	/F803	MASK F ELIMINATING NORM
205	H1000	DC	/1000	STMT TYPE END
206	H07FF	DC	/07FF	MASK TO GET SYM TBL ADDR
207	H7FFF	DC	/7FFF	CONSTANT
208	HF300	DC	/F300	CONSTANT
209	H0680	DC	/0680	CONSTANT
210	H0880	DC	/0880	CONSTANT
211	LDXL1	DC	/5800	.LDX L1.
212	SUBSC	DC	/1900	.CALL SUBSC.
213	HFF80	DC	/FF80	MASK FOR .CALL SUBPROG.
214	*			ALSO, .CALL IFIX. MINUS
215	*			.CALL FLT.
216	HFFF0	DC	/FFF0	CONSTANT
217	H007F	DC	/007F	CONSTANT
218	GLT80	DC	GLABT-1	GENERATED LABEL TBL ORIGIN
219	TEMP1	DC	0	TEMPORARY STORAGE
220	SBSCL	DC	/5000	CONSTANT
221	H500C	DC	/500C	STMT-ID FOR .SUBIN.
222	*			FORMER ID OF INTEGER STMT
223	*			
224	HD800	DC	/D800	CONSTANT
225	HD004	DC	/D004	CONSTANT
226	H8000	DC	/8000	CONSTANT
227	H4F80	DC	/4F80	CONSTANT
228	H0220	DC	/0220	CONSTANT
229	HF000	DC	/F000	STMT ID DEFN FILE STMT
230	HFFFF	DC	/FFFF	CONSTANT
231	TEMP3	DC	0	TEMPORARY STORAGE
232	TEMP4	DC	/8000	TEMP4 INITIALLY NEGATIVE
233	*			
234	*			TEMPORARIES AND SWITCH FOR SUBPR-INI
235	*			WHEN PROGRAM INCLUDES ASF-STATEMENTS
236	TEMP5	DC	0	TEMPORARY STORAGE
237	TEMP6	DC	0	TEMPORARY STORAGE
238	TEMP7	DC	0	TEMPORARY STORAGE
239	ASFSW	DC	0	ARITHMETIC STMT FUNC SW

```

240 SNTSW DC 0 SUBR NAME TRANS SW
241 ASFN DC 0 TEMP STORAGE OF ARITH
242 * *STMT FUNC NAME
243 *
244 * SUBR .GET SYM TABL ID-WORD.
245 *
246 GETST DC 0 SUBROUTINE ENTRY POINT
247 AND 3 H07FF-Z GET SYM TBL POINTER
248 STO GETS1 1 SAVE
249 LD L SOFST GET START OF SYM TBL
250 S GETS1 1 SUBTRACT SYM TBL PT 1 TIME
251 S GETS1 1 *2 TIMES
252 S GETS1 1 *3 TIMES
253 A 3 THREE-Z INCR BY 3
254 STO GETS1 1 SAVE SYM TBL LOC OF ID WD
255 GETS1 LD L *-* LOAD SYM TBL ID WORD
256 BSC I GETST RETURN TO CALLING PROGRAM
257 *
258 * SUBR. TRACKSWITCH
259 * SET UP BR TO LOC SPECIFIED IN THE
260 * SUBROUTINE CALLING SEQUENCE DEPENDEN
261 * ON THE VALUE IN TR 1,2 OR 3
262 *
263 TRSW DC 0 LINK
264 LD TRSW GET ADDR OF LOC TO BR TO
265 A 3 TR-Z STORED TRACKSW VALUE
266 STO TRSW1 1 SAVE IN LINK RETURN ADDR
267 TRSW1 BSC I 0 RETURN TO DESIRED LOC.
268 *
269 * SUBR .NAMETEST.
270 NAMT DC 0 LINK 76
271 S 3 TESTN-Z TESTNAME
272 AND 3 H07FF-Z
273 BSC Z SKIP IF SAME
274 MDX NAMT1 BR IF DIFFERENT
275 MDX L LISTL,1 INCR LISTLENGTH
276 *
277 * TEST IF SUBPROGRAM NAME TRANSFER
278 * SWITCH SNTSW SET
279 * IF SET, OUTPUT LOCTR2,
280 * OTHERWISE OUTPUT LOCTR2 1
281 *
282 LD SNTSW LD SW
283 BSC Z SKIP IF SWITCH NO SET
284 LD 3 HFFFF-Z LOAD MINUS ONE
285 A 3 ONE-Z ADD 1
286 A 3 LOCC2-Z LOCTR2
287 OR 3 H8000-Z TAG TO FACILITATE HANDLING
288 BSI 3 OUTP-Z OUTPUT
289 NAMT1 BSC I NAMT RETURN
290 *
291 * OUTPUT SUBROUTINE
292 OUTP DC 0 LINK
293 STO 2 1 MOCE CONTENTS ACC
294 * *TO OUTPUT STRING
295 MDX 2 1 INCR O/P POINTER
296 LD I OUTID GET ID NORM V1M
297 AND H07FC MASK OUT NORM V1M
298 A 3 FOUR-Z ADD ONE TO NORM V1M
299 SLA 4 LOOK AT BIT 4 V1M

```

```

300      BSC  L  NOOVF,-   SHOULD BE 0. BR IF SO  V1M
301      LD   3 FOUR-Z   NORM TOO BIG. PUT A FO V1M
302      STO  L  SOFS     IN ERROR PARAMETER FOR V1M
303      BSI  L  ROLRX    CALL PHASE-28
304      DC   28
305  NOOVF LD   I  OUTID   INCR NORM IN          V1M
306      A   3 FOUR-Z   *ID-WD
307      STO  I  OUTID   *OF OUTPUT STRING
308  *
309  *
310      STX  L1 TIN     SAVE O/P STRING ADDR
311      STX  L2 TOUT
312      LD   3 ASFSW-Z  AVOID OVERLAP TEST IF
313      BSC  I  OUTP,Z  *ASF SWITCH SET
314      LD   TIN     GET I/P STRING ADDR
315      S    TOUT    SUBTRACT O/P STRING ADDR
316      BSC  I  OUTP,Z- RETURN IF NO OVERLAP ERROR
317  *
318  *
319      MDX  L  ERROR,1  INCR ERROR FLAG
320      BSC  L  EXIT     BR TO EXIT THIS PHASE
321  *
322  TIN   DC   0        TEMP STO OF I/P POINTER XR1
323  TOUT  DC   0        TEMP STO OF O/P POINTER XR2
324  *
325  *
326  TEMP2 DC   0        TEMP STO I/P POINTER
327  *
328  XIN   DC   0        I/P LOCATION POINTER
329  XOUT  DC   0        O/P LOCATION POINTER
330  *
331  ENT   LD   L  ERROR   LOAD OVERLAP ERROR FLAG
332      BSC  L  EXIT,Z   BR TO EXIT IF OVERLAP ERROR
333  *
334      LOX  L3 ZERO    INITIALIZE TRANSFER VECTOR
335  *
336  *
337  *
338  *
339      LD   L  EOFS     LD END OF STRING
340      S    L  SOFS     -START OF STRING
341      A   3 ONE-Z     PLUS ONE
342      STO  MOVCT     SAVE IN CNT OF NO. WDS
343      LOX  I1 EOFST   FIND LOC NXT TO SYM TBL
344      MOX  1 -6       BUT SIX POSITIONS AWAY
345      LOX  I2 EOFS    PUT ENDO OF STRING PT XR2
346  *
347  *
348      STX  L1 XOUT    SAVE NEW O/P LOC POINTER
349      STX  L2 XIN     SAVE NEW I/P LOC POINTER
350      LD   XOUT     LD OUTPUT LOC PT
351      S    XIN     LESS INPUT LOC PT
352      BSC  L  MOVST,-Z BR IF NOT OVERLAP
353      MOX  L  ERROR,1 SET OVERLAP ERR FLAG
354      BSC  L  EXIT     EXIT FROM THIS PHASE
355  *
356  MOVST LD   2 0      MOVE STRING ITEM
357      STO  1 0      *FROM I/P STRING TO O/P
358      MOX  2 -1     DECR I/P POINTER
359      MOX  1 -1     DECR O/P POINTER

```

```

360          MDX  L  MOVCT,-1  DECR WD CNT
361          MDX          MOVST      BR IF NO SKIP TO
362  *                                MOVING OF NEXT ITEM
363          LDX  I2 SOFS          SET XR2   START OF ORIGINAL
364          MDX  2  -1           *STRING-1  NOW O/P AREA
365  *
366  *                                IF DEFINE FILE OR DATA STATEMENT,
367  *                                MOVE STATEMENT WITHOUT COUNTING
368  *                                LOCATIONS
369  *
370  LDR      LD    1  1           LOAD STRING WORD
371          AND   3  HF800-Z     GET ID BITS
372          S     3  HF000-Z     SUBTRACT DEFN FILE ID
373          BSC   Z              SKIP IF ZERO  DEFN FILE
374          S     DATA          SUBTRACT DATA TEST ID
375          BSC   L  M1011,Z     BR IF NOT DEFN FILE OR DATA
376  *
377  *                                MOVE STATEMENT
378          LD    1  1           LD STMT ID
379          AND   3  H07FC-Z     GET NORM COUNT
380          STO          MOVCT     SAVE WD COUNT
381  LDR1     MDX  2  1           INCR INPUT CTR
382          MDX  1  1           INCR O/P CTR
383          LD    1  0           MOVE WD FROM INPUT AREA
384          STO   2  0           * TO OUTPUT AREA
385          MDX  L  MOVCT,-4     DECR WD CNT BY ONE
386          MDX  LDR1          BR IF NO SKIP
387          MDX  LDR           BR BACK TO TEST NXT STMT
388  *
389  MOVCT DC          0           COUNT ON NO. WDS MOVED
390  DATA DC         /F800-/F000 DATA TEST CONSTANT
391  *
392  *
393  *                                INSERT SIZE OF WORK AREA PLUS
394  *                                SIZE OF CONSTANTS AREA AS INITIAL
395  *                                VALUE OF LOCATION COUNTER
396  *                                DEFINEFILE AREA IS INCLUDED IN
397  *                                WORK AREA  SOFXT
398  *
399  M1011 LD    L  SOFXT         LD SIZE OF WORK AREA
400          A     L  SOFGT         ADD SIZE OF CONSTANTS AREA
401          STO   3  LOCC1-Z     SAVE IN LOC COUNTER
402  *
403  M1021 MDX  1  1           INCR INPUT POINTER
404          LD    1  0           LD STIRNG WORD
405          AND   3  H07FC-Z     STORE NORM
406          STO   3  NORM-Z       STMT LENGTH COUNTER
407          LD    1  0           GET STMT TYPE
408          AND   3  HF800-Z     MASK TO GET STMT ID
409          S     3  H6000-Z     FORMAT-TYPE
410          BSC   L  M1031, -    BR IF FORMAT
411          LD    1  0           LOAD STRING WORD
412          AND   3  HF800-Z     MASK TO GET STMT ID
413          S     3  HD800-Z     SUBTRACT CALL INTERNAL O/P
414          BSC   L  M1051,Z     BR IF NOT CALL IO
415  *
416  *                                STORE LOCCOUNTER1 FOR LATER USAGE
417  *                                AS ENTRY POINT
418  *                                FIRST ENCOUNTERED CALL IO STMT
419  *

```

```

420      LD      SW      LD CALL I/O INITLZ SW
421      BSC    L M1022,Z BR IF SWITCH SET
422      LD      3 LOCC1-Z LOAD LOC CTR
423      STO    3 TEMP4-Z SAVE IN TEMP STO
424      STX    3 SW      SET SWITCH
425      *
426      M1022 MDX    2 1      INCR O/P POINTER
427      MDX      M1042      BR TO MOVE WD TO O/P STRING
428      *
429      SW     DC      0      SWITCH INITIALLY ZERO
430      *
431      *      MOVE FORMAT-STMNT OR CALL-I/O-STMNT
432      *      UNALTERED TO OUTPUT
433      *      STRING. ALLOCATE STATEMENT
434      *
435      M1031 LD      1 1      LD STMNT NUMBER
436      STO    2 2      SAVE IN O/P STRING
437      BSI    3 GETST-Z GET SYM TBL ID WORD
438      OR     3 2      . ALLOCATED.
439      STO    I GETS1 1 SAVE IN SYM TBL
440      LD      1 0      LOAD STRING ID WORD
441      A      3 FOUR-Z INCR NORM CNT
442      STO    2 1      SAVE IN O/P STRING
443      LD      3 LOCC1-Z LD LOC COUNTER WD 1
444      STO    2 3      SAVE IN O/P STRING
445      MDX    1 1      INCR I/P POINTER
446      MDX    2 3      INCR O/P POINTER
447      MDX    L NORM,-8 DECR NORM COUNT
448      *
449      M1041 MDX    1 1      INCR I/P POINTER
450      MDX    2 1      INCR O/P POINTER
451      MDX    L LOCC1,1 INCR LOCCOUNTER1
452      M1042 LD      1 0      MOVE WORD AT INPUT POINTER
453      STO    2 0      *TO OUTPUT STRING
454      MDX    L NORM,-4 DECR STATEMENTLENGTHCOUNTER
455      MDX    M1041      BR IF NO SKIP
456      MDX    M1021      BR IF SKIP
457      *
458      *      PROCESS A STATEMENT OTHER THAN
459      *      FORMAT-STATEMENT
460      *
461      M1051 LD      1 0      LD I/P STRING ID WD
462      AND    3 HF800-Z GET ID TYPE
463      S      3 HD000-Z TEST ARITH STMNT FUNC TYPE
464      BSC    Z      SKIP IF ARITH STMNT FUNC
465      MDX    M1071      BR IF NOT
466      **
467      *      PROCESS AN ARITHMETIC STATEMENT
468      *      FUNCTION STATEMENT
469      *
470      *
471      *      STORE POINTERS IN ORDER TO
472      *      ESTABLISH LATER A PSEUDO STMT-ID-WOR
473      *
474      LD      3 HD004-Z PUT ARITH STMNT FUNC ID WD
475      STO    2 1      *PLUS 1WD NORM IN O/P
476      MDX    2 1      INCR O/P STRING PT
477      STX    L2 OUTID  SAVE O/P STRING ID WD PT
478      STX    L1 TEMP2  SAVE I/P STRING POINTER
479      *

```



```

480 *          TEST IF FIRST FOUND ARITH STMT FUNC
481 *          IF SO, STORE PT AND LOC FOR USE
482 *          IN SUBPROG. INITIALIZE STRING CNT
483 *
484          LD      3 TEMP5-Z   LD ARITH STMT FUNC FOUND S
485          BSC    L M1052,Z   BR IF SET BEFORE
486          STX    L2 TEMP5    STORE POINTER
487          LD      3 LOCC1-Z   STORE LOCCTR
488          S      3 ONE-Z     SUBTRACT 1
489          STO     3 TEMP6-Z   SAVE IN TEMP STORAGE
490 M1052      LD      1 1      LD STRING WD
491          BSI     3 OUTP-Z    PUT ON O/P STRING
492          LD      3 LOCC1-Z   LD LOC COUNTER
493          BSI     3 OUTP-Z    PUT ON O/P STRING
494 *
495          LD      1 1      LD ARITH STMT FUNC NAME
496          STO     3 ASFN-Z    SAVE
497 *
498          BSI     3 GETST-Z   GET SYM TBL ID WORD
499          OR      3 TWO-Z     CON TWO ALLOCATION BIT
500          STO     I GETS1 1   SAVE IN SYM TBL ID WORD
501 M1061      LD      3 H5F00-Z  LD CONSTANT BLANK WD
502          BSI     3 OUTP-Z    OUTPUT BLANK
503          MDX    L LOCC1,1   INCR LOCCTR1
504          LD      3 ZERO-Z    ZERO TO TRACKSWITCH
505 *          0 .ARITH STMT FUNCTION.
506          BSI     L S2011    STRING COUNT
507          LD      3 LOCC2-Z   LOCCTR2 TO
508          STO     3 LOCC1-Z   *LOCCTR1
509          STO     3 TEMP7-Z   SAVE IN TEMP LOC
510          BSC    L M1021    BR TO GET NXT LOC
511 *
512 *
513 *          PROCESS A STATEMENT OTHER THAN
514 *          ARITHMETIC STATEMENT FUNCTION OR
515 *          FORMAT
516 M1071      LD      3 LOCC1-Z   LOCCTR1
517          STO     L COMON     SAVE AS RELATIVE ENTRY PT
518 *
519 *          SAVE XR1 FOR LATER REINITIALIZATION
520          STX    L1 TINP     SAVE XR1
521 *
522 *
523 *          IF TEMP4 SET, ADDRESS THERE REPLACES
524 *          ENTRY POINT
525 *
526          LD      3 TEMP4-Z   LD TEMP4 LOCC1 IF SET
527          BSC    L M1072, Z   BR IF NOT SET INITLZ TO -
528          STO     L COMON     SAVE AS RELATIVE ENTRY PT
529 *
530 M1072      MDX     1 1      INCR I/P POINTER
531          LD      L SORF     GET SUBR/FUNC FLAG
532          BSC    -         SKIP IF THIS IS A SUBPROGRA
533          MDX     M1092     BR IF NOT
534 *
535 *          OUTPUT .CALL SUBIN. WITH ITS
536 *          ARGUMENTS IF THIS A SUBPROGRAM WITH
537 *          DUMMY VARIABLES
538 *
539 *          TEST IF DUMMY VAIABLES IN SYM TBL

```

```

540 *
541 M1081 LD      3 0          ZERO TO DUMMY
542          STO    3 DUMVC-Z  *VARIABLE COUNTER
543          LD     L SOFST    GET START OF SYMBOL TABLE
544          STO    M1082 1    SAVE
545 M1082 LD     L *-*        GET FIRST SYM TBL ID WD
546          RTE    27        SHIFT DUMMY VAR BIT TO SIGN
547          BSC    -         SKIP IF DUMMY VAR
548          MDX    M1083     BR IF NOT
549          MDX    L DUMVC,1  INCR DUMMY VAR COUNTER
550 M1083 RTE    5          SHIFT TO GET SYM TBL ID ACC
551          AND    3 H1800-Z  MASK FOR DIMENSION BITS
552          BSC    -Z        SKIP IF NOT DIMENSIONED
553          LD     3 HFFFFD-Z LD -3 CONSTANT FOR DIM VAR
554          A      3 HFFFFD-Z ADD -3 CON FOR SYM TBL ENTR
555          A      M1082 1    ADD CURRENT SYM TBL ID ADDR
556          STO    M1082 1    SAVE ID ADDR OF NXT ID
557          S      L EOFST    SUBTRACT END OF SYM TBL
558          BSC    -Z        SKIP IF END OF SYM TBL
559          MDX    M1082     BR IF NOT
560          LD     3 DUMVC-Z  DUMMY VAR COUNTER
561          BSC    Z         SKIP IF SUBP HAS NO DUMMY
562 *
563          MDX    M1085     BR IF IT HAS
564 *
565 *
566 *
567 *
568 *
569          LD     3 H6008-Z  LD ID WD W/ NORM 2
570          STO    2 1        PUT ON O/P STRING
571          LD     3 ZERO-Z   LD ZERO WD
572          STO    2 2        PUT ON O/P STRING
573          MDX    2 2        INCR OUTPUT POINTER
574          MDX    L LOCC1,1  INCR LOCATION COUNTER
575          MDX    M1092     BR TO CONTINUE
576 M1085 A      3 TWO-Z     PUT NO. DUMMY VAR 2
577          STO    3 LISTL-Z  *IN LIST LENGTH
578 M1091 LD     3 ONE-Z     .ONE. TO TRACKSWITCH
579 *
580          BSI    S2011     1 .SUBPROGR. INITIALIZ..
581          LD     3 LOCC1-Z  STRING COUNT SUBR
582          A      3 ONE-Z   GET LOC COUNTER
583          A      3 LISTL-Z INCR BY 1
584          STO    3 LOCC1-Z INCR BY LIST LENGTH
585 M1092 LD     3 TWO-Z     SAVE IN LOC COUNTER
586 *
587          BSI    S2011     TWO TO TRACKSWITCH
588 *
589 *
590 *
591          LD     3 LOCC2-Z  2 .STATEMENT ALLOC..
592          BSC    E         STRING COUNT SUBR
593          A      3 ONE-Z   STORE PROGRAM LENGTH
594          STO    L SOFNS   MAKING IT EVEN
595          BSC    L EXIT    LD LOC CTR
596 *
597 *
598 *****
599 *

```

```

600 *          SUBR .STRING COUNT.
601 *
602 *          THIS SUBROUTINE PERFORMS THREE
603 *          DIFFERENT TASKS DEPENDING ON THE
604 *          SETTING OF TRACKSWITCH
605 *
606 *          IF TRACKSWITCH  0
607 *              IT INITIALIZES AN ARITHMETIC STATE
608 *              MENT FUNCTION STATEMENT
609 *              OUTPUTS .CALL SUBIN. WITH
610 *              ARGUMENTS
611 *
612 *          IF TRACKSWITCH  1
613 *              IT INITIALIZES SUBPROGRAM
614 *              OUTPUTS .CALL SUBIN. WITH
615 *              ARGUMENTS
616 *
617 *          IF TRACKSWITCH  2
618 *              IT ALLOCATES STATEMENTS
619 *              AND MOVES THE STRING
620 *              INSERTING ALLOCATIONS TO FOLLOW
621 *              STATEMENT NUMBERS IN LABELED STMENTS
622 *
623 *****
624 S2011 DC      0          LINK
625         STO   3 TR-Z    STORE TRACKSWITCH
626         LD    3 LOCC1-Z LOGCTR1
627         STO   3 LOCC2-Z *TO LOGCTR2
628         BSI   3 TRSW-Z  TRACKSWITCH
629         DC    S2061     ARITH STMENT FUNCTION
630         DC    S2021     SUBPROGRAM INITIALIZATION
631         DC    S2091     STMENT ALLOCATION
632 *
633 S2021 LD      L SOFST   INITLZ DUMMY VAR POINTER
634         STO   3 DUMVP-Z *IN SYMBOL TABLE
635         LD    3 0       *AND SYM TBL ADDR
636         STO   3 STADR-Z *COUNTER
637         LDX  I1 TINP    INITIALIZE INPUT POINTER
638 *
639 *          INITIALIZE OUTPUT STMENT
640         LD    3 H500C-Z  MOVE STMENT ID  SUBR INITLZ
641         STO   2 1       * TO OUTPUT ID WORD
642         MDX   2 1       INCR OUTPUT POINTER
643         STX  L2 OUTID   STORE ADDR OF OUTPUT-ID
644         LD    3 SUBIN-Z  .CALL SUBIN.
645         STO   2 2       TO OUTPUT STRING
646         LD    3 H5F00-Z  * TO OUTPUT STRING
647         STO   2 1
648         MDX   2 3       INCR OUTPUT POINTER
649         STX  L2 S2052 1  STORE ADDR OF BEGINNING OF
650 *          *LIST ON OUTPUT STRING
651         MDX   2 -1
652         MDX  L LOCC2,2  DECR O/P POINTER
653 S2031 LD    3 DUMVC-Z  INCR LOC COUNTER
654         BSC
655         MDX   S2051     BR IF ZERO
656         S     3 ONE-Z
657         STO   3 DUMVC-Z  DUMMY VAR COUNTER
658 S2032 MDX  L STADR,1   INCR SYM TBL ADDR COUNTER
659         LD    I DUMVP   LD DUMMY VAR PT FR SYM TBL

```

```

660 RTE 27 SHIFT BIT TO SIGN POS
661 BSC - SKIP IF DUMMY VAR
662 MDX S2041 BR IF NOT
663 RTE 5 SHIFT ID WD TO ORIGINAL POS
664 *
665 SLA 8 SHIFT SUBP NAME BIT TO SIGN
666 BSC L S203Y,- BR IF NOT SUBPROGRAM NAME
667 MDX L LISTL,1 INCR LIST LENGTH
668 MDX L SNTSW,1 SET SWITCH
669 *
670 S203Y LD 3 STADR-Z LD SYM TBL ADDR COUNTER
671 STO 3 TESTN-Z TESTNAME
672 *
673 * TEST IF PROGRAM INCLUDES ASF STMENTS
674 * IF IT DOES, SPECIAL-INITIALIZE
675 *
676 S203X LD 3 TEMP6-Z
677 BSC L S2033, - BR IF PROGR INCLUDES NO ASF
678 STO 3 LOCC2-Z INITIALIZE LOCCTR
679 LDX I1 TEMP5 INITIALIZE POINTER
680 MDX L ASFSW,1 SET SWITCH
681 BSC L S2093 BR TO PROCESS ASF
682 *
683 S2033 LD 3 LOCC1-Z LOCCTR1
684 STO 3 LOCC2-Z *TO LOCCTR2
685 BSC L S2092 BR TO ALLOCATE STMENT
686 *
687 S2041 RTE 5 PUT SYM TBL ID WD ALL IN AC
688 AND 3 H1800-Z TEST FOR DIMENSIONED VAR
689 BSC L S2042, - BR IF NONE
690 MDX L STADR,1 INCR SYM TBL ADDR COUNTER
691 LD 3 HFFFD-Z GET -3 FOR DIM VAR IN SYMTB
692 S2042 A 3 HFFFD-Z INCR BY -3 SYM TBL CONSTANT
693 A 3 DUMVP-Z ADD DUMMY VAR PT
694 STO 3 DUMVP-Z SAVE IN DUMMY VAR PT
695 MDX S2032 BR TO PROCESS NXT VAR
696 S2051 LD 3 LOCC2-Z INCR LOCCTR2 BY LIST LENGTH
697 A 3 LISTL-Z LISTLENGTH
698 STO 3 LOCC2-Z SAVE
699 LD 3 ONE-Z OUTPUT CONSTANT ONE
700 BSI 3 OUTP-Z BR TO O/P SUBR
701 *
702 * NOTE
703 * XR2 IS NOT RESTORED, SINCE IT
704 * WILL GET CORRECT VALUE IN
705 * THE PROCESS OF CORRECTING
706 * ARGUMENTS OF .SUBIN.
707 *
708 S2052 LDX L2 *-* XR2 START OF LIST-O/P STRIN
709 *
710 S205X LD 2 0 LD LIST ITEM
711 BSC - SKIP IF LIST ITEM NONZERO
712 MDX S2053 BR IF ZERO
713 SRT 1 SHIFT OUT 1 BIT
714 BSC L S205Y,Z BR IF LISTITEM NOT .ONE.
715 *
716 * END OF SUBIN ARGUMENTS
717 BSI 3 TRSW-Z TRACKSWITCH
718 DC S2054 ARITH STMENT FUNCTION
719 DC S2059 SUBPROGRAM INITIALIZE

```

```

720 *
721 *           INCR .SUBIN. ARGUMENTS IF GT TEMP7
722 S205Y LD      2 0           LD ARGUMENT
723         AND    3 H7FFF-Z   REMOVE SIGN BIT
724         S      3 TEMP7-Z   SUBTRACT TEMP7
725         BSC   L S2053, Z   BR IF LT TEMP7
726         LD    3 LISTL-Z   INCR VALUE
727         A     2 0           *IN ARGUMENT LIST
728         STO   2 0           *BY LIST LENGTH
729 *
730 S2053 MDX    2 1           INCR ARGUMENT POINTER
731         MDX   S205X       BR TO TEST NXT LIST ITEM
732 *
733 *           MOVE THE BODY OF
734 *           ARITH STMT FUNCTION
735 *
736 S2054 LDX    I1 TINP      LD START OF STMT XR1
737         LD    1 0           LD STMT ID WD
738         AND    3 H07FC-Z   MASK TO GET NORM
739         S      3 FOUR-Z    DECR BY 1
740         STO   3 NORM-Z    SAVE NORM
741 S2057 BSI    3 INCPT-Z    BR TO INCR I/P POINTER
742         LD    1 0           LOAD STRING WORD
743         BSI   3 OUTP-Z    BR TO PUT ON O/P STRING
744 *
745 S2058 LD     3 NORM-Z     LD NORM
746         BSC   L S2057,Z-  BR IF NORM POSITIVE
747         LD    3 H4F80-Z   LD CODE FOR BSC I W/NAME
748         BSI   3 OUTP-Z    BR TO O/P CODE
749         LD    3 ASFN-Z    LD ARITH STMT FUNC NAME
750         BSI   3 OUTP-Z    BR TO O/P NAME
751         MDX   L LOCC2,2   INCR LOCATION COUNTER
752 *
753 S2059 BSC    I S2011     RETURN
754 *
755 *
756 *           ARITHMETIC STATEMENT FUNCTION INIT.
757 *
758 S2061 MDX    1 1           INCR I/P POINTER
759         STX   L1 DUMVP    INITIAL DUMMY VAR POINTER
760         LD    3 ZERO-Z    CLEAR
761         STO   3 DUMVC-Z   *DUMMY VAR COUNTER
762 S2062 MDX    1 1           INCR I/P POINTER
763 *
764 *           MOVE DUMMY ARGUMENTS ONE STEP
765 *           TO MAKE ROOM FOR PSEUDO STMT-ID
766 *
767         LD    1 0           LD STRING WD
768         STO   1 -1        MOVE TO CURRENT LOC-1
769 *
770         BSC   -           SKIP IF DUMMY ARGUMENT
771 *           *NAME OR PACKED INSTRUCTION
772         MDX   S2063       BR IF NOT
773         AND    3 H7800-Z   MASK NAME ID BITS
774         BSC   Z           SKIP IF DUMMY ARGUMENT NAME
775         MDX   S2063       BR IF NOT
776         MDX   L DUMVC,1   INCR DUMMY VAR COUNT
777         MDX   S2062       BR TO GET NXT VAR
778 *
779 S2063 LD     3 DUMVC-Z   DUMMY VAR COUNTER 1

```

```

780      A      3 ONE-Z      *TO LIST
781      STO    3 LISTL-Z    *LENGTH
782      MDX    1 -1        DECR I/P COUNTER
783      STX    L1 TINP      INSERT STORED INITIAL VALUE
784      *
785      *
786      *          PRODUCE PSEUDO-STMNT-ID WHICH IS
787      *          REQUIRED BY OTHER PARTS OF THIS
788      *          ROUTINE
789      *          FOR HANDLING NORM PROPERLY
790      *
791      LD      L3 TEMP2-Z    LD START OF STMNT PT
792      S        3 TINP-Z    SUBTRACT VAL OF I/P POINTER
793      SLA     2            SHIFT TO INCR NORM COUNT
794      A        I TEMP2     INCR NORM COUNT OF STMNT ID
795      STO     1 0          SAVE IN I/P STRING
796      *
797      LD      3 SUBIN-Z    LD .SUBIN. ID WD
798      BSI     3 OUTP-Z    OUTPUT .CALL SUBIN.
799      *
800      *          STORE LOCATION COUNTER FOR REINITIAL
801      LD      3 LOCC2-Z    LD LOC COUNTER
802      STO     3 TEMP3-Z    SAVE
803      *
804      *          INSERT ADDR F BEGINNING OF
805      *          ARGUMENT LIST
806      *
807      MDX     2 1          INCR O/P POINTER
808      STX     L2 S2052 1    SAVE O/P ADDR
809      MDX     2 -1         DECR O/P POINTER
810      *
811      S2081 LD      3 DUMVC-Z  DUMMY VAR COUNTER
812      BSC
813      MDX     S2051        BR IF ZERO END OF LIST
814      S        3 ONE-Z    DECR DUMMY VAR COUNTER
815      STO     3 DUMVC-Z    SAVE
816      LD      I DUMVP      DUMMY VAR TO .TESTNAME.
817      STO     3 TESTN-Z    SAVE
818      BSI     3 GETST-Z    BR TO GET SYM TBL ID WD.
819      BSI     3 SUBPR-Z    TEST IF SUBP NAME
820      *
821      *          ERASE DUMMY VAR NAME IN SYM TBL TO
822      *          AVOID MIXUP WITH SUBPROGRAM DUMMIES
823      *
824      LD      3 H0220-Z    LD STMNT NO, DEFN VAR FLAG
825      STO     I GETS1 1    SAVE AS SYM TBL ID WD.
826      *
827      MDX     L DUMVP,1    INCR DUMMY VAR.POINTER
828      *
829      *          RESET LOCATION COUNTER
830      LD      3 TEMP3-Z    LD WD FROM TEMP STORAGE
831      STO     3 LOCC2-Z    SAVE
832      *
833      MDX     S2092        CONTINUE
834      *
835      S2091 LDX    I1 TINP    SET INPUT POINTER
836      *
837      *          TEST IF FIRST STATEMENT IS AN
838      *          .END.-STATEMENT
839      *

```

840		LD	1 0	LD STMT ID WD
841		AND	3 HF800-Z	MASK TO GET TYPE BITS.
842		S	3 H1000-Z	SUBTRACT END STMT ID
843		BSC	L S2132, -	BR IF END STATEMENT
844	*			
845		LD	1 0	OUTPUT ID-WD NORM 4
846		AND	3 HF803-Z	MASK ID BITS
847		A	3 FOUR-Z	COMPUTE NORM 4
848		MDX	2 1	INCR O/P POINTER
849		STO	2 0	STORE IN OUTPUT STRING
850		STX	L2 OUTID	STORE ADDR OF OUTPUT-ID
851	S2092	LDX	I1 TINP	INITIAL I/P POINTER
852	S2093	LD	1 0	INITIALIZE STMT LENGTH
853		AND	3 H07FC-Z	*COUNTER
854		STO	3 NORM-Z	* NORM
855		BSI	3 TRSW-Z	TRACKSWITCH
856		DC	S2106	ARITH STMT FUNC
857		DC	S2101	SUBP INITIALIZATION
858		DC	S2101	STMT ALLOCATION
859	S2101	LD	3 ASFSW-Z	GET ASF SW
860		BSC	L S2115,Z	BR IF ASF SW SET
861		LD	1 0	LD STMT ID WD
862		BSC	E	SKIP IF STMT HAS NO LABEL
863		MDX	S2102	BR IF IT HAS
864		MDX	S2106	BR TO INCR I/P POINTER
865	S2102	BSI	3 TRSW-Z	TRACKSWITCH
866		DC	S2104	ARITH STMT FUNC
867		DC	S2104	SUBP INITIALIZATION
868		DC	S2103	STMT ALLOCATION
869	S2103	LD	1 1	GET STMT ID
870		BSI	3 OUTP-Z	PUT IN O/P STRING
871		LD	1 1	GET SYM TBL ID OF STMT NAM
872		BSI	3 GETST-Z	INSERT ALLOCATION BIT
873		OR	3 2	
874		STO	I GETS1 1	SAVE BACK IN SYM TBL
875		LD	3 LOCC2-Z	GET LOC CTR
876		BSI	3 OUTP-Z	PUT ON O/P STRING
877	S2104	BSI	3 INCPT-Z	INCR INPUT POINTER
878	S2106	BSI	3 INCPT-Z	INCR INPUT POINTER
879	S2111	LD	3 NORM-Z	STMT COUNTER FR NORM
880		BSC	Z-	SKIP IF END OF STMT
881		MDX	S2141	BR IF NOT
882		BSI	3 TRSW-Z	TRACKSWITCH
883		DC	S2112	ARITH STMT FUNC
884		DC	S2121	SUBP INITIALIZATION
885		DC	S2121	STATEMENT ALLOCATION
886	S2112	LD	3 0	LD A CONSTANT ZERO
887		BSI	3 OUTP-Z	PUT IN O/P STRING
888		MDX	L LOCC2,1	INCR LOC COUNTER
889		MDX	S2081	BR TO TEST DUMMY VAR COUNT
890	*			
891	*			ASF SWITCH SET, INCR INPUT POINTER
892	*			ONE EXTRA WORD ALLOCATION IS ON STR
893	*			
894	S2115	BSI	3 INCPT-Z	BR TO INCR I/P POINTER ONCE
895		MDX	S2104	BR TO CONTINUE TO INCR PT
896	*			
897	*			TEST IF SWITCH SET IF I/P PT REFERS
898	*			TO ASF-STMT IN OUTPUT STRING
899	*			

```

900 S2121 LD      3 ASFSW-Z  GET ASF SW
901      BSC    L  S2124,Z  BR IF SWITCH SET
902 *
903      LD      1 0        LD STMT ID WD
904      AND     3 HF800-Z  GET STMT TYPE
905      S       3 H1000-Z  SUBTRACT.END.STMNT CON
906      BSC     -         SKIP IF NXT STMT NOT .END.
907      MDX    S2131      BR IF .END.
908      LD      1 0        INITIALIZE STMT LENGTH
909      AND     3 H07FC-Z  *COUNTER
910      STO     3 NORM-Z   * NORM
911      BSI     3 TRSW-Z   TRACKSWITCH
912      DC      S2101      ARITH STMT FUNC
913      DC      S2101      SUBPR INITIALIZATION
914      DC      S2122      STATEMENT ALLOCATION
915 S2122 LD      1 0        O/P NXT ID
916      AND     3 HF803-Z  INITIALIZE NORM
917      A       3 FOUR-Z   COMPUTE NORM 4
918      STO     2 1        SAVE IN O/P STRING
919      MDX     2 1        INCR O/P STRING POINTER
920      STX    L2 OUTID    SAVE OUTPUT ID WORD
921      MDX     S2101      BR TO CONTINUE
922 *
923 *          HANDLE ASF- AND FORMAT STAEMENTS
924 *          AT SUBPROGRAM INITIALIZE STRING COUN
925 *
926 S2124 LD      1 0        LD I/P STRING ID WORD
927      AND     3 HF800-Z  MASK TYPE BITS
928      S       3 H6000-Z  SUBTRACT FORMAT TYPE CON
929      BSC    L  S2126,Z  BR IF STMT NOT FORMAT
930 *
931 *          MOVE POINTER TO NEXT STATEMENT
932 *          AND COUNT IN LOCATION COUNTER
933 *
934      LD      1 0        LD ID WORD
935      AND     3 H07FC-Z  MASK TO GET STMT NORM
936      SRA     2         RIGHT JUSTIFY
937      STO     S2125 1    SAVE NORM COUNT
938      S       3 THREE-Z  DO NOT COUNT ID WORD,
939 *          STMT NO. AND ALLOCATION
940      A       3 LOCC2-Z  INCR LOC COUNTER
941      STO     3 LOCC2-Z  SAVE IN LOC COUNTER
942 S2125 MDX    L1 *-*     TEST FOR ZERO NORM
943      MDX     S2124      BR IF NOT ZERO
944 *
945 S2126 S       3 H7000-Z  0000 - 6000
946      BSC    L  S2093, - BR IF ASF
947 *
948 *          SWITCH POINTER TO REFER TO INPUT
949 *          STRING, CHANGE SWITCH
950 *
951      LD      3 ZERO-Z   CLEAR
952      STO     3 ASFSW-Z  *ASF SW
953      LD      3 LOCC1-Z  MOVE LOC
954      STO     3 LOCC2-Z  *COUNTER
955      LDX    I1 TINP    LD INPUT POINTER
956      MDX     S2093      BR
957 *
958 S2131 BSI     3 TRSW-Z  TRACKSWITCH
959      DC      S2133      *SHOULD NEVER USE

```



```

960          DC      S2133  SUBP INITIALIZATION
961          DC      S2132  STMT ALLOCATION
962  *
963  *                OUTPUT .END.-STATEMENT
964  S2132 LD      1 0      LD .END. WORD
965          STO     2 1      PUT IN O/P STRING
966          MDX     2 2      INCR O/P POINTER
967          STX     L2 EOF5  INSERT END-OF-STRING
968          BSC     I S2011  RETURN
969  *
970  S2133 LD      3 ZERO-Z  OUTPUT
971          BSI     3 OUTP-Z  CONSTANT ZERO
972  *
973  *                IF SUBPROG NAME TRANSFER SWITCH
974  *                SNTSW SET, RUN SAME DUMMY VARIABLE
975  *                NAME ONCE MORE
976          LD      3 SNTSW-Z LD TRANSFER SW
977          BSC     L S2134, - BR IF SW NOT SET
978          LD      3 ZERO-Z  ZERO TO SWITCH
979          STO     3 SNTSW-Z
980          BSC     L S203X    BR BACK
981  S2134 LD      3 DUMVC-Z  DUMMY VAR COUNTER
982          BSC     L S2051, - BR IF COUNT IS ZERO
983          S       3 ONE-Z   DECR DUMMY VAR COUNTER
984          STO     3 DUMVC-Z
985          LD      I DUMVP    GET DUMMY VAR NAME
986          BSC     L S2041 1  TO BRANCH TO S2031 PRECEDED
987  *                BY MOVING DUMMY VAR POINTER
988  S2141 LD      1 0      LD STRING I/P WORD
989          BSC     -        SKIP IF WORD AT PT PACKED
990  *                *OR NAME ONLY
991          MDX     S2151     BR IF NOT
992          AND     3 H7800-Z  MASK TO GET NAME BITS
993          BSC     L S2143, - BR IF NAME ONLY
994  *
995          MDX     L LOCC2,1  COUNT EXTRA FOR PACKED INST
996  *
997  S2143 BSI     3 TRSW-Z    TRACKSWITCH
998          DC      S2144     ARITH STMT FUNC
999          DC      S2144     SUBP INITLZ
1000         DC      S2145     STMT ALLOCATION
1001  S2144 LD      1 -1     LD STRING WD
1002          S       3 H6200-Z  TEST IF
1003         BSC     L S2146, -  BR IF YES
1004         LD      1 0      LD NXT STRING WD
1005         BSI     3 NAMT-Z    NAMETEST
1006         MDX     S2146     BR TO INCR LOCCTR
1007  *
1008  S2145 LD      1 0      LD STRING WORD
1009         BSI     3 OUTP-Z    OUTPUT WORD AT POINTER
1010  S2146 MDX     L LOCC2,1  INCR LOC CTR
1011         MDX     S2106     BR TO CONTINUE
1012  *
1013  *                OPERATOR .LDX L1. HAS BEEN
1014  *                ENCOUNTERED
1015  *
1016  S2151 LD      1 0      LD STRING WD
1017          S       3 LDXL1-Z  TEST FOR .LDX L1.
1018         BSC     Z          SKIP IF .LDX L1.
1019         MDX     S2161     BR IF NOT

```

1020	MDX	L	LOCC2,1	INCR LOCCOUNTER2
1021	BSI	3	INCPT-Z	INCR I/P POINTER
1022	BSI	3	TRSW-Z	TRACKSWITCH
1023	DC		S2146	ARITH STMT FUNC
1024	DC		S2146	SUBP INITLZ
1025	DC		S2152	STMT ALLOCATION
1026	S2152	LD	1 -1	LD .LDX L1 WD FR STRING
1027	BSI	3	OUTP-Z	OUTPUT .LDX L1.
1028	MDX		S2145	BR TO O/P NXT WD IN STRING
1029	*			
1030	*			
1031	S2161	LD	1 0	LD STRING WORD
1032	S	3	SUBSC-Z	
1033	BSC		-	TEST FOR .CALL SUBSC.
1034	MDX		S2171	BR IF .CALL SUBSC.
1035	LD	1	0	LD STRING WD
1036	AND	3	HFF80-Z	TEST FOR .CALL SUBPROGRAM.
1037	BSC	L	S3011, -	BR IF .CALL SUBPROGRAM.
1038	*			* 2-WD CALL
1039	*			
1040	*			TEST IF GENERATED LABEL OPERATOR
1041	*			
1042	LD	1	0	LD STRING WD
1043	AND	3	HFF80-Z	MASK TO GET ONLY OPERATOR
1044	S	3	H5E00-Z	TEST IF GENERATED LABEL
1045	BSC		-	SKIP IF NO
1046	MDX		S2201	BR IF YES
1047	*			
1048	*			TEST IF CALL SUBIN
1049	LD	1	0	LD STRING WD
1050	S	3	SUBIN-Z	TEST FOR CALL SUBIN
1051	BSC	L	S2165, -	BR IF CALL SUBIN
1052	*			
1053	S	3	H0680-Z	TEST FOR SDAF
1054	BSC		Z	SKIP IF SDAF
1055	S	3	HFF80-Z	TEST FOR SDAT
1056	BSC		Z	SKIP IF SDAI OR SDAF
1057	S	3	HF300-Z	TEST FOR FIOAF
1058	BSC		Z	SKIP IF ANY OF ABOVE 3
1059	S	3	HFF80-Z	TEST FOR FIOAI
1060	BSC		Z	SKIP IF ANY OF ABOVE 4
1061	S	3	H0880-Z	TEST FOR UIOAF
1062	BSC		Z	SKIP IF ANY OF ABOVE 5
1063	S	3	HFF80-Z	TEST FOR VIOAI
1064	BSC	L	S2162,Z	BR IF NOT ANY OF ABOVE
1065	BSI	3	INCPT-Z	INCR I/P POINTER
1066	MDX	L	LOCC2,1	INCR LOC CTR
1067	BSI	3	TRSW-Z	TRACKSWITCH
1068	DC		TINE	ASF
1069	DC		FORK	SUBP INITLZ
1070	DC		PRONG	STMT ALLOCATION
1071	PRONG	LD	1 -1	LD STRING WD
1072	BSI	3	OUTP-Z	PUT OUT LIBF
1073	LD	1	0	LD NXT STRING WD
1074	BSI	3	OUTP-Z	PUT OUT LIBF
1075	LD	1	1	LD NXT STRING WD
1076	BSI	3	OUTP-Z	PUT OUT ALLOCATION
1077	TINE	MDX	L LOCC2,2	INCR LOC CTR
1078	BSC	L	S2104	CONTINUE THROUGH STMT
1079	FORK	LD	1 0	LD STRING WD SUBP INITLZ

```

1080          BSI      3 NAMT-Z   TEST FOR NAME
1081          MDX      TINE       BR TO INCR LOC CTR
1082 S2162 BSI      3 TRSW-Z   TRACK SW
1083          DC        S2146    ASF
1084          DC        S2146    SUBP INITLZ
1085          DC        S2145    STMT ALLOCATION
1086 *
1087 *           CALL SUBIN  HAS BEEN ENCOUNTERED
1088 *           MOVE POINTER PAST ARG LIST, COUNT LO
1089 *
1090 S2165 MDX      L LOCC2,1   INCR LOC CTR
1091          LD        1 0        LD STRING PTR
1092          S         3 ONE-Z   TEST FOR END OF ARGUMENTLIS
1093          BSC      L S2106, - BR IF END
1094          BSI      3 INCPT-Z  BR TO INCR I/P PT
1095          MDX      S2165     BR TO CONTINUE THROUGH LIST
1096 *
1097 *           .CALL SUBSC. HAS BEEN ENCOUNTERED
1098 S2171 BSI      3 TRSW-Z   TRACKSWITCH
1099          DC        S2173    ASF
1100          DC        S2173    SUBP INITLZ
1101          DC        S2172    STMT ALLOCATION
1102 *
1103 *           PROCESS .CALL SUBSC., SGT, D4
1104 *
1105 S2172 LD        1 0        LD STRING WD 0
1106          BSI      3 OUTP-Z  BR TO O/P WD
1107          LD        1 1        LD STRING WD 1
1108          BSI      3 OUTP-Z  BR TO O/P WD
1109          LD        1 2        LD STRING WD 2
1110          BSI      3 OUTP-Z  BR TO O/P WD
1111 *
1112 S2173 BSI      3 INCPT-Z  INCR I/P POINTER ONCE
1113          BSI      3 INCPT-Z  INCR I/P POINTER AGAIN
1114          MDX      L LOCC2,3  INCR LOC CTR BY 3
1115 S2181 BSI      3 TRSW-Z   TRACKSWITCH
1116          DC        S2182    ASF
1117          DC        S2182    SUBP INITLZ
1118          DC        S2183    STMT ALLOCATION
1119 *
1120 *           PROCESS D1 OR D2 OR D3
1121 *           AND A VARIABLE
1122 *
1123 S2182 LD        1 1        LD STRING WD.
1124          BSI      3 NAMT-Z   NAMETEST
1125          MDX      S2184     BR TO CONTINUE
1126 S2183 LD        1 1        LD STRING WD1
1127          BSI      3 OUTP-Z  PUT ON O/P STRING
1128          LD        1 2        LD STRING WD2
1129          BSI      3 OUTP-Z  BR TO PUT ON STRING
1130 S2184 BSI      3 INCPT-Z  INCR I/P POINTER ONCE
1131          BSI      3 INCPT-Z  INCR I/P POINTER TWICE
1132          MDX      L LOCC2,2  INCR LOC CTR BY 2
1133 *
1134 *           TEST IF LAST TAGGED ARGUMENT
1135          LD        1 0
1136          BSC      L S2106, Z LD NXT STRING WD
1137          MDX      S2181     BR IF WD NEGATIVE
1138 *           BR TO CONTINUE THROUGH STRI
1139 *

```

```

1140 *          GENERATED LABEL OPERATOR HAS BEEN
1141 *          ENCOUNTERED
1142 *
1143 S2201 LD      1 0          LD STRING WD
1144         AND    3 H007F-Z   MASK TO GET PT IN TABLE
1145         A      3 GLT80-Z   TABLE ORIGIN ADDRESS
1146         STO    S2202 1     SAVE ADDR
1147 S2202 LD      L *-*      LD TABLE WD
1148         BSC    Z          SKIP IF NOT IN TABLE
1149         MDX    S2203      BR IF IN TABLE
1150 *
1151 *          INSERT INTO TABLE
1152         LD      3 LOCC2-Z   INSERT LOCCTR2 AS
1153         STO    I S2202 1   * TABLE VALUE
1154         BSC    L S2106     BR TO CONTINUE
1155 *
1156 *          ELIMINATE FROM TABLE
1157 S2203 STO     3 TEMP1-Z    STORE TABLE VALUE
1158         LD      3 0        ZERO TO TABLE
1159         STO    I S2202 1
1160         BSI    3 TRSW-Z    TRACKSWITCH
1161         DC     S2216       ASF
1162         DC     S2216       SUBP INITLZ
1163         DC     S2211       STMNT ALLOCATION
1164 *
1165 *
1166 *          OUTPUT SPECIAL BSC L
1167 *
1168 S2211 LD      3 SBSCL-Z    OUTP .SPEC BSC L.
1169         BSI    3 OUTP-Z    BR TO O/P SUBR
1170         LD      3 TEMP1-Z  OUTPUT VALUE FR GENERATED
1171         BSI    3 OUTP-Z    * LABEL TABLE
1172 S2216 MDX    L LOCC2,2    INCR LOC CTR BY 2
1173         BSC    L S2106    BR TO CONTINUE
1174 *
1175 *
1176 *          .CALL.-OPERATOR HAS BEEN ENCOUNTERED
1177 *
1178 *          INCR INPUT POINTER AND LOCCOUNTER 2
1179 *
1180 S3011 BSI     3 INCPT-Z    BR TO INCR I/P POINTER
1181         MDX    L LOCC2,1   INCR LOC CTR
1182 *
1183         BSI    3 TRSW-Z    TRACKSWITCH
1184         DC     S3012       ASF
1185         DC     S3012       SUBP INITLZ
1186         DC     S3013       STMNT ALLOC
1187 *
1188 S3012 LD      1 0          LD STRING WD
1189         BSI    3 NAMT-Z    TEST IF NAME
1190         BSC    L S2146    BR TO INCR LOC CTR
1191 *
1192 *
1193 *          TEST IF NAME
1194 *          DUMMY VARIABLE
1195 *
1196 S3013 LD      1 0          LD STRING WD
1197         BSI    3 GETST-Z   BR TO GET SYM TBL ID WD
1198         AND    3 H0400-Z   MASK FOR DUMMY VAR BIT
1199         BSC    L S3015,Z   BR IF DUMMY VARIABLE

```

```

1200 *
1201 *
1202 *      OUTPUT .CALL SUBPROGRAM. AND
1203 *      NAME
1204 *      WORDS AT POINTER - 1 AND
1205 *      AT POINTER
1206 *
1207      LD      1 -1      LD WD AT POINTER-1
1208      BSI     3 OUTP-Z   BR TO O/P
1209      LD      1 0      LD WD AT POINTER
1210      BSI     3 OUTP-Z   BR TO O/P
1211      BSC     L S2146   BR TO CONTINUE THROUGH STMN
1212 *
1213 *
1214 *      NAME IS BOTH DUMMY VARIABLE AND
1215 *      EXTERNAL
1216 *      OUTPUT BSI I, NAME AT POINTER
1217 *
1218      S3015 LD      H4780   LD .BSI I. CONSTANT
1219      BSI     3 OUTP-Z   PUT ON O/P STRING
1220      LD      1 0      LD NAME FR I/P STRING
1221      BSI     3 OUTP-Z   PUT ON O/P STRING
1222      BSC     L S2146   BR TO CONTINUE
1223 *
1224      H4780 DC      /4780   .BSI I.
1225 *
1226 *
1227 *****
1228 *
1229      GLABT DC      0      GENERATED LABEL TABLE
1230      DC      0      WORKING WORDS
1231      DC      0      WORKING WORDS
1232      DC      0      WORKING WORDS
1233 *
1234      EXIT  BSI     L ROLRX   CALL DOWN PHASE 23
1235      DC      23      NEXT PHASE NUMBER
1236      BSS     OVERL-+320*3  PHASE-22 PATCH AREA
1237      END      NEQ

```