

```

1  * GENERAL AUTOMATION, INC, ALL RIGHTS RESERVED
2  *****
3  *
4  * PROGRAM NAME   FPH-24
5  *
6  * MODEL NUMBER  8F024
7  *
8  * PURPOSE       FORTRAN PHASE=24
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 ** LIST SYMBOLS
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION
26 * * LISTS THE FEATURES SUPPORTED BY THE PROGRAM
27 * AS INDICATED IN THE FORTRAN COMMUNICATIONS
28 * AREA WORD CCWD,
29 * * LISTS THE SYSTEM LIBRARY SUBROUTINES USED
30 * BY THE PROGRAM IF REQUESTED,
31 * * LISTS SUBPROGRAM NAMES FOUND IN THE SYMBOL
32 * TABLE, IF REQUESTED,
33 *
34 *ENTRY POINTS-
35 * * NEQ - PHASE 24 IS LOADED BY PHASE 23 VIA
36 * ROLRX, EXECUTION IS BEGUN AT LOCATION
37 * LABLED NEQ,
38 *INPUT-
39 * * THE STATEMENT STRING
40 * * THE SYMBOL TABLE
41 * * THE FORTRAN COMMUNICATIONS AREA
42 *
43 *OUTPUT-
44 * * THE STATEMENT STRING
45 * * THE SYMBOL TABLE
46 * * THE FORTRAN COMMUNICATIONS AREA
47 *
48 *EXTERNAL REFERENCES
49 * * SUBROUTINES
50 * ROLRX
51 * * OTHER FORTRAN PHASES
52 * PHASE 19 OR 30 OR NEITHER
53 *
54 *EXITS-
55 * * NORMAL-
56 * PHASE 25 IS LOADED VIA A CALL TO THE
57 * ROLRX ROUTINE AND CONTROL PASSED TO IT,
58 * * ERRORS-
59 * OVERLAP-

```

```

60 *           IF AN OVERLAP HAS BEEN DETECTED IN A
61 *           PREVIOUS PHASE THEN AN IMMEDIATE EXIT
62 *           TAKES PLACE, OTHERWISE, NO OVERLAP
63 *           ERROR IS DETECTED,
64 *           SYNTAX*
65 *           NO SYNTAX ERRORS ARE DETECTED
66 *
67 *TABLES/WORK AREAS
68 * * THE STATEMENT STRING
69 * * THE SYMBOL TABLE
70 * * THE FORTRAN COMMUNICATIONS AREA
71 * * BUF+40, A 40 WORD MESSAGE BUILDING AND
72 *   OUTPUTTING AREA,
73 *
74 *ATTRIBUTES=N/A
75 *
76 *NOTES-NONE
77 *****
78 ABS   REF CORE
79 *
80 *   SYSTEM AND FORTRAN EQUATES
81 *
82 MEMRY EQU   FFFF CORE   MAXIMUM CORE SIZE
83 PHSIZ EQU   4*320         MAXIMUM PHASE SIZE
84 OVERL EQU   MEMRY-PHSIZ   PHASES 2-29 START
85 FCOM EQU    OVERL-22      FORTRAN COMM, TABLE
86 PHNTB EQU   FCOM-56       PHASE TABLE
87 ROLRX EQU   PHNTB-50      INTERPHASE CALL
88 AREA EQU    OVERL+3*320+100 PRINT DATA ADDRESS
89 PRINT EQU   AREA+1        PRINT ENTRANCE
90 *
91 *   FORTRAN COMMUNICATIONS AREA
92 *
93 *   ORG      FCOM      FORTRAN COMM AREA
94 SOFS BSS    1         START OF STRING
95 EOFS BSS    1         END OF STRING
96 SOFST BSS   1         START OF SYMBOL TABLE
97 SOFNS BSS   1         LENGTH OF PROGRAM
98 SOFXT BSS   1         SIZE OF WORK AREA (VARIES)
99 SOFGT BSS   1         SIZE OF CONSTANTS AREA
100 EOFST BSS  1         END OF SYMBOL TABLE
101 COMON BSS   1         RELATIVE ENTRY POINT
102 CSIZE BSS   1         SIZE OF COMMON
103 ERROR BSS   1         ERROR FLAG
104 *           BIT 15 OVERLAP ERROR
105 *           BIT 14 OTHER ERROR
106 FNAME BSS   1         PROGRAM NAME
107      BSS    1         2ND WORD OF PROG NAME
108 SORF BSS    1         SUBK (=) OR FUNC (+)
109 CCWD BSS    1         CONTROL CARD WORD
110 *           BIT 15 TRANSFER TRACE
111 *           BIT 14 ARITHMETIC TRACE
112 *           BIT 13 EXTENDED PRECISION
113 *           BIT 12 LIST SYMBOL TABLE
114 *           BIT 11 LIST SUBPROGRAM NAMES
115 *           BIT 10 LIST SOURCE PROGRAM
116 *           BIT 9 ONE WORD INTEGERS
117 *           BIT 8 PUNCH
118 *           BIT 7 NONPROCESS PROGRAM
119 IOCS BSS    1         IOCS CONTROL CARD WORD

```

```

120 *
121 *          SEE PHASE ONE FOR BIT PATTERNS
122 *
123 DFCNT BSS      1          DEFINE FILE COUNT
124 *
125 LCOMN BSS      2          INSKEL COMMON SIZE
126 *
127 ICCER BSS      2          IOCS CONT CD ERROR WORD
128 *
129 *          END OF FORTRAN COMMUNICATION
130 *          AREA
131 *****
132 *
133          ORG      OVERL
134 NEQ      BSC      L      ENT          GO BEGIN PHASE
135 *
136 *
137 WDCNT DC          ***          WORD COUNT
138 BUF      DC          /4000      PRINT AREA
139          DC          /4000      EBC BLANKS
140          DC          /4000      *
141          DC          /4000      *
142 ENT      LD      L      ERROR
143          BSC      L      EXIT,Z          BK TO EXIT IF OVERLAP ER
144 *          INITIALIZATION PROGRAM IS EXECUTED
145 *          AT LOCATIONS THAT LATER SERVE AS
146 *          PRINT AREA
147 *
148 *
149 *          INITIALIZE TRANSFERVECTOR
150 *
151          LDX      L3 Z          LD XR3 WITH POINTER VAL
152 *
153 *
154 *          LIST FEATURES SUPPORTED
155 *
156 *          ANY FEATURES SUPPORTED
157 *
158 *
159          LD      IOCS          LD IOCS WORD
160          BSC      Z          SKIP IF ALL BITS OFF
161          MDX      BUF40        ELSE GO TO BUF+40
162          LD      L      CCWD          LD CONT CARD WD
163          AND      3 H0147-Z      COMPARE BITS 7,9,13,16,15
164          BSC      L      L1035, - BRANCH IF NO FEATURES SUPP
165          MDX      BUF40        ELSE GO TO BUF+40
166 *
167 *          IF ANY FEATURES SUPPORTED
168 *          PRINT TEXT "FEATURES SUPPORTED"
169 *
170          ORG      BUF+40
171 *
172 BUF40 BSI      L      PBLNK          GO BLANK BUFFER AREA
173          BSI      L      HTEST          GO UNPACK AND PRINT MSG
174          DC          9          WORD COUNT OF MSG
175          EBC          .FEATURES SUPPORTED,
176 *
177 *          PRINT NON-PROCESS
178 *          IF APPLICABLE
179 *

```

```

180          LD      CCWD      LOAD CONTROL CARD WORD
181          SLA      7        PUT BIT 7 IN SIGN
182          BSC     L K1020,-  BRANCH IF PROCESS
183          BSI     L HTEST    ELSE GO UNPACK AND PRINT
184          DC      6         WORD COUNT OF MSG
185          EBC      . NONPROCESS ;
186          *
187          *          PRINT ,TRANSFER TRACE,
188          *          IF APPLICABLE
189          *
190 K1020 LD      CCWD      LOAD CONTROL CARD WD
191          SLA      15       PUT BIT 15 IN SIGN
192          BSC     L K1021,-  BR IF NO TRANSFER TRACE
193          BSI     L HTEST    ELSE GO UNPACK AND PRINT
194          DC      8         WORD COUNT OF MSG
195          EBC      . TRANSFER TRACE ;
196          *
197          *          PRINT ,ARITHMETIC TRACE,
198          *          IF APPLICABLE
199          *
200 K1021 LD      L CCWD      LOAD CONTROL CARD WD
201          SLA      14       PUT BIT 14 IN SIGN
202          BSC     L K1031,-  BR IF NO ARITHMETIC TRACE
203          BSI     L HTEST    ELSE GO UNPACK AND PRINT
204          DC      9         WORD COUNT OF MSG
205          EBC      . ARITHMETIC TRACE ;
206          *
207          *          PRINT ,ONE WORD INTEGERS,
208          *          IF APPLICABLE
209          *
210 K1031 LD      L CCWD      LOAD CONTROL CARD WD
211          SLA      9         PUT BIT 9 IN SIGN
212          BSC     L K1051,-  BR IF NO ONE WORD INTEGERS
213          BSI     L HTEST    ELSE GO UNPACK AND PRINT
214          DC      9         WORD COUNT OF MSG
215          EBC      . ONE WORD INTEGERS,
216          *
217          *          PRINT ,EXTENDED PRECISION,
218          *          IF APPLICABLE
219          *
220 K1051 LD      L CCWD      LOAD CONTROL CARD WD
221          SLA      13       PUT BIT 13 IN SIGN
222          BSC     L L1030,-  BR IF NOT EXTENDED PREC
223          BSI     L HTEST    ELSE GO UNPACK AND PRINT
224          DC      10        WORD COUNT OF MSG
225          EBC      . EXTENDED PRECISION ;
226          *
227          *          CHANGE SYSTEM SUBR,=NAME=LIST SO THA
228          *          APPLICABLE NAMES BEGIN WITH .E,
229          *          RATHER THAN WITH .F,
230          *
231          LDX      1 FAXIX-TABSS+2 GET TABLE LENGTH
232 L1022 BSI      3 FTOE=2    CHANGE NEXT NAME
233          MDX      1 -2     MOVE TABLE ENTRY POINTER
234          MDX      L1022    BR IF XR1 NOT ZERO
235          *
236          *
237          *          CHANGE SECOND CHAR IN TRACE ROUTINE
238          *          NAMES FROM F INTO E
239          *

```

```

240      LD      3 H0004-Z   LD A HEX 4 CONST
241      STO     L DIFF     PUT IT IN DIFF
242      LDX     1 FARI-TABSS+2 PUT TABLE COUNT IN XR1
243      BSI     3 FTOE=Z   GO CHANGE F TO E
244      LDX     1 FARIX-TABSS+2 PUT TABLE COUNT IN XR1
245      BSI     3 FTOE=Z   GO CHANGE F TO E
246      LDX     1 FIF-TABSS+2 PUT TABLE COUNT IN XR1
247      BSI     3 FTOE=Z   GO CHANGE F TO E
248      *
249      *           TEST FOR IOCS
250      L1030 LD     L IOCS   LOAD IOCS WORD
251      L1035 LD     L CCWD   LOAD CONTROL CARD WORD
252      SLA     11         PUT BIT 11 IN SIGN
253      BSC     L L2011,-   BRANCH IF NO LISTING
254      *
255      *           CHECK IOCS WORD AND FLAG APPROPRIATE
256      *           SUB-ROUTINE NAME TO BE PRINTED
257      *
258      LD     L ICCER     EXAMINE UTAPE/UDISK ERR WD
259      BSC     L L2011,Z  BR OUT IF ERROR COND EXIST
260      LD     L ZERO     FORCE NO I/O CALL LIST
261      SRT     14        PUT BITS 2 THRU 15 IN Q
262      BSI     L UTOY,E  GO CHENGE U TO Y IF UDISK
263      LDX     2 6        SET XR2 TO 6
264      SLT     2         RETRIEVE PLOTTER BIT
265      SLA     15        AND TEST IF PLOTTER IS
266      BSC     L OTHS,+  SPECIFIED BR OUT IF NOT
267      MDX     EXTPT     BR AROUND ROUTINE/CONTINUE
268      *
269      UTOY   DC     ***   LINK ADDR WORD
270      STX     1 XR1SV   SAVE XR1 HERE
271      LDX     L1 UCOMP-URED+2 GET A TABLE COUNT
272      UTOYL  LD     L1 URED=2 GET A TABLE ENTRY
273      A       H0400    CHANGE THE U TO A Y
274      STO     L1 URED=2 PUT IT BACK IN TABLE
275      MDX     1 -2     MOVE THE POINTER
276      MDX     UTOYL    LOOP BACK
277      *           UPON FALL THROUGH
278      LDX     L1 UIOI=UFIO+2 LOAD ANOTHER TABLE LENGT
279      LUTOY  LD     L1 UFIO=2 GET A TABLE ENTRY
280      A       H0400    CHANGE THE U TO Y
281      STO     L1 UFIO=2 PLACE IT BACK IN THE TABLE
282      MDX     1 -2     MOVE THE POINTER
283      MDX     LUTOY    LOOP BACK
284      LDX     L1 ***    RESTORE XR1 UPON COMPLETIO
285      XR1SV  EQU     *-1  XR1 SAVED THERE
286      BSC     I UTOY    EXIT UTOY TO CALLING POINT
287      *
288      H0400 DC     /0400  CONSTANT CHANGES U TO Y
289      *
290      *****
291      *           SUBROUTINE
292      *           TAG A SYSTEM SUBROUTINE NAME IN
293      *           LIST BY SETTING LEFTMOST BIT ON
294      *
295      EXTPT  LDX     L1 TABLL SET XR1 TO TABLE ADDR
296      LD     L CCWD   GET CONT CARD WORD RECORD
297      SLA     13     PUT BIT 13 IN SIGN
298      BSC     -      SKIP IF EXTENDED PRECISION
299      MDX     1 2    ELSE MOVE TABLE POINTER

```

```

300      LD      1 0      LOAD TABLE WORD TO ACC
301      OR      3 H8000-Z  FLAG IT TURN BIT 0 ON
302      STO     1 0      RETURN WORD TO TABLE
303  OTHS  SLT     6      PUTS BITS 4 THRU 9 IN ACC
304      LDX    L1 TABLM-Z  PUT AN ADDR IN XR1
305  *
306  *          THIS ROUTINE FLAGS I/O AND CONVERSI
307  *          ROUTINE NAMES WHICH WILL BE CALLED
308  *          BY THE LOADER
309  *
310  OTHR  MDX     1 2      MOVE TABLE ENTRY POINTER
311      SLA     16      ZERO THE ACC
312      SLT     1      PUT A BIT OF Q IN ACC
313      BSC    L CKND, -  BRANCH IF BIT WAS OFF
314      LD      I1 0      ELSE LD C(TABLE ADDR)
315      OR      3 H8000-Z  TURN BIT ZERO ON
316      STO     I1 0      RESTORE THE WORD
317      LD      I1 1      LD C(NEXT TABLE ENTRY)
318      BSC    Z        SKIP IF ZERO
319      OR      3 H8000-Z  ELSE TURN BIT ZERO ON
320      STO     I1 1      RESTORE THE WORD
321  CKND  MDX     2 -1     MOVE TABLE POINTER BACK
322      MDX     OTHR      LOOP BACK UNTIL XR2 FLIPS
323  *
324      LDX    I1 SOFST  INITIALIZE SYM,TAB,POINTER
325      STX    L1 STP    PUT ADDR OF SYM TAB HERE
326  *
327  *          BRANCH TO PROGRAM AREA
328  *          OUTSIDE THE PRINT AREA
329  *
330      BSC    L L1041
331  *
332  *          UNPACK AND PRINT HEADER
333  HTEST  DC      ***    LINK ADDR WORD
334      LD      I HTEST  LD CONTENTS OF ADDR
335      STO     L WDCNT  PUT THEM IN WDCNT
336      STO     INC      ALSO IN HERE
337      LD      HTEST  RELOAD THE ADDR
338      A      3 ONE-Z  INCREASE IT BY ONE
339      STO     HTES2 1  PUT MOD ADDR BEHIND HTES2
340      A      INC      ADD WDCNT TO HTEST+1
341      STO     L XTEST 1  PUT HEADER LENGTH IN HERE
342      SLT     32      ZERO ACC AND Q
343      LDX    3 0      ZERO XR3
344      LDX    I2 WDCNT  LD THE WD COUNT INTO XR2
345  HTES2  LD      L ***    LD A PACKED MSG WORD
346      SRT     8      PUT ONE PACKED CHAR IN Q
347      SLA     8      LEFT JUSTIFY OTHER CHAR
348      STO     L3 BUF  STORE CHAR IN BUF STRING
349      SLT     16      GET OTHER CHAR BACK IN Q
350      STO     L3 BUF+1  STORE IT IN NEXT BUF LOC
351      MDX     3 2      INCREASE BUF STRING PTR
352      MDX    L HTES2 1,1  INCREASE PACKED WD PTR
353      MDX     2 -1     DECREMENT WORD COUNT
354      MDX     HTES2    LOOP BACK UNTIL XR2 FLIPS
355      LDX    L3 Z      RESTORE XR3 TO TBL PTR VAL
356      LD      3 PAPIN-Z  GET START ADDR OF BUF
357      STO     L AREA    PUT IT IN AREA
358      BSI     L PRINT  GO PRINT THE MSG
359      BSI     L BLKPA   GO PUT EBC BLANKS IN BUF

```

```

360 XTEST BSC L *** AND RETURN TO CALL PT
361 *
362 INC DC *** MSG WDCNT STORED HERE
363 *
364 *****
365 *
366 * .TRANSFER VECTOR, SUBROUTINES FOLLOW
367 *****
368 *
369 *
370 * SUBROUTINE
371 * GET SYMBOL TABLE ID WORD
372 *
373 GETST DC *** LINK ADDR WORD
374 AND 3 H07FF-2 SAVE ONLY BITS 5 THRU 15
375 STO GETS1 1 ENTRY NUM OF SYM TBL ENTRY
376 LD L SOFST LOAD ADDR OF SYM TBL
377 S GETS1 1 FOR EACH ENTRY THERE ARE
378 S GETS1 1 3 WORDS THUS 3 TIMES THE
379 S GETS1 1 ENTRY NUM GETS THE ADDR
380 A 3 3 ADD 3 TO GET 1ST WD
381 STO GETS1 1 STO THE SYM TBL ENT ADDR
382 GETS1 LD L *** IN HERE
383 BSC I GETST RETURN THRU LINK WD ADDR
384 *
385 *
386 * SUBROUTINE
387 * CHANGE ,F, IN FIRST CHARACTER OF A
388 * SUBROUTINE NAME INTO ,E,
389 * OR E,F INTO V,W
390 *
391 * DISPLACEMENT IN NAME LIST X1
392 DIFF DC /0100 CONSTANT WITH BIT 7 ON
393 FTOE DC *** LINK ADDR WORD
394 LD L1 TABSS-2 GET FIRST WORD IN NAME
395 S DIFF CHANGE NAME FROM F TO E
396 * SUBTRACTING DIFF
397 STO L1 TABSS-2 RESTORE THE MOD NAME
398 BSC I FTOE RETURN THRU LINK ADDR WD
399 *****
400 *
401 * SUBROUTINE TO CHANGE U TO Y IN FIO CALLS
402 *
403 *****
404 *
405 * SUBROUTINE
406 * .TAG, SYSTEM SUBR, NAME IN LIST BY
407 * MAKING LEFTMOST BIT ONE
408 *
409 TAGSR DC *** LINK ADDR WORD
410 LD L2 TABSS-2 GET AN ENTRY FROM TABLE
411 OR 3 H8000-2 TURN BIT ZERO ON
412 STO L2 TABSS-2 RESTORE WORD INTO TABLE
413 BSC I TAGSR RETURN THRU LINK ADDR WORD
414 * MOVE STRING POINTER
415 MOVEP DC *** LINK ADDR WORD
416 MDX 1 1 MOVE STRING WORD POINTER
417 SLA 0 A NO OP FOR SKIPPING
418 MDX L NRM,-4 DECREMENT STMT WDCNT
419 SLA 0 A NO OP FOR SKIPPING

```

```

420          BSC  I  MOVEP      RETURN THRU LINK ADDR WORD
421  *
422  *          SUBR, MOVE SYMBOL TABLE POINTER
423  MSTP  DC          ***      LINK ADDR WORD
424          LD    1  0          LD ID WORD IN SYM TBL
425          AND  3  MASK3-Z     SAVE ONLY BITS 3 AND 4 IF
426          BSC  Z          SKIP IF NOT DIMENSIONED
427          LD    3  CM3-Z      MINUS THREE
428          S    3  3          SET ACC TO =3 OR =6
429          A    3  STP-Z      ADD IN SYM TBL PTR ADDR
430          STO  3  STP-Z      STOR STP=STP-3, =6 HERE
431          LDX  I1 STP        PUT POINTER IN XR1
432          BSC  I  MSTP      RETURN THRU LINK ADDR WORD
433  *
434  D60   DC    32          A CONSTANT FOR WDCNT
435  PAPIN DC    BUF        INITIAL VALUE OF PAP
436  PAP   DC    BUF        PRINT AREA POINTER
437  C120  DC    64          ANOTHER CONST FOR WDCNT
438  Z     DC    0          CONSTANT ZERO
439  ZERO  EQU   Z          A POINTER ADDR
440  ONE   DC    1          CONSTANT ONE
441  TWO   DC    2          CONSTANT TWO
442  THREE DC    3          CONSTANT THREE
443  FOUR  DC    4          CONSTANT FOUR
444  H0004 EQU   FOUR      HEX 4 FOR MASKING
445  H07FF DC    /07FF     MASK TO SAVE BITS 5 THRU 1
446  *
447  H0100 DC    /0100     MASK TO SAVE BIT 7
448  H8000 DC    /8000     MASK TO FLIP BIT 0
449  STP   DC    ***      SYM_T, POINTER
450  MASK2 DC    /0080     ALS/, CONSTANT
451  MASK3 DC    /1800     MASK FOR BITS 3,4
452  CM3   DC    /FFFD     CONST, = MINUS THREE
453  MASK5 DC    /F800     FOR EXTRACTING ST-ID
454  MASK6 DC    /07FC     FOR EXTRACTING NORM
455  NRM   DC    ***      STOKED NORM
456  STID  DC    ***      STMT-ID
457  FTEST DC    /6000     STMT-ID = FORMAT
458  ENDDID DC /1000     STMT-ID = END
459  H1800 DC    /1800     BITS 3 AND 4 MASK
460  H1900 DC    /1900     CALL SUBSC
461  H1E00 DC    /1E00     THESE ARE DIFFERENCE
462  H5B00 DC    /5B00     CODES FOR
463  H0500 DC    /0500     FINDING ENTRIES
464  H0147 DC    /0147     IN THE TABSS
465  H0080 DC    /0080     TABLE
466  H0008 DC    /0008     FOR MOVING PCNT
467  H7800 DC    /7800     BITS 1,2,3,4 MASK
468  H1D00 DC    /1D00     CALL SUBIN
469  H1600 DC    /1600     CALL FIUA
470  H1680 DC    /1680     CALL FIUAF
471  H5D00 DC    /5D00     SPEC BSC L
472  H4000 DC    /4000     EBC BLANK
473  HD800 DC    /D800     FORMAT TYPE CODE
474  H4700 DC    /4700     ,BS, LS, CODE
475  DFILE DC    /F000     DEFINE FILE ID
476  DATA DC    /F800, /F000 DATA STMT ID
477  *
478  *          SYSTEM SUBR, TABLE
479  TABSS DC    /0604     FADD CODE 008

```


480		DC	/4100		
481		DC	/0604	F_ADDX	010
482		DC	/4127		
483		DC	/068A	F_SUB	018
484		DC	/4080		
485		DC	/068A	F_SUBX	020
486		DC	/40A7		
487		DC	/0651	F_MPY	028
488		DC	/7A00		
489		DC	/0651	F_MPYX	030
490		DC	/7A27		
491		DC	/0610	F_DIV	038
492		DC	/9940		
493		DC	/0610	F_DIVX	040
494		DC	/9967		
495		DC	/064C	F_LD	048
496		DC	/4000		
497		DC	/064C	F_LDX	050
498		DC	/49C0		
499		DC	/068A	F_STO	058
500		DC	/3580		
501		DC	/068A	F_STOX	060
502		DC	/35A7		
503		DC	/0688	F_SBR	068
504		DC	/2640		
505		DC	/0688	F_SBRX	070
506		DC	/2667		
507		DC	/0612	F_DVR	078
508		DC	/5640		
509		DC	/0612	F_DVRX	080
510		DC	/5667		
511		DC	/0606	F_AXI	088
512		DC	/7240		
513	FAXIX	DC	/0606	F_AXIX	090
514		DC	/7267		
515		DC	/0626	F_IXI	098
516		DC	/7240		
517		DC	/0626	F_IXIX	0A0
518		DC	/7267		
519	FARI	DC	/1418	M_FAR	0A8
520		DC	/1640		
521	FARIX	DC	/1418	M_FARX	0B0
522		DC	/1667		
523	FIAR	DC	/1424	M_IAR	0B8
524		DC	/1640		
525	FIARX	DC	/1424	M_IARX	0C0
526		DC	/1667		
527	AIFIX	DC	/0918	I_FIX	0C8
528		DC	/99C0		
529	AFLT	DC	/064D	F_FLOAT	0D0
530		DC	/6063		
531		DC	/0359	C_OMGU	0D8
532		DC	/41D6		
533		DC	/098A	I_STOX	0E0
534		DC	/35A7		
535	FIIF	DC	/1424	M_IIF	0E8
536		DC	/9180		
537	FIF	DC	/1418	M_IIF	0F0
538		DC	/9180		
539	FGOTO	DC	/141D	M_GOTU	0F8

540		DC	/68D6		
541		DC	/1310	LDFAC	100
542		DC	/6043		
543		DC	/228C	STFAC	108
544		DC	/6043		
545		DC	/2208	SBFAC	110
546		DC	/6043		
547		DC	/0494	DVFAC	118
548		DC	/6043		
549	URED	DC	/2464	URED	120
550		DC	/5100		
551		DC	/2499	UWRI	128
552		DC	/98C0		
553	UCOMP	DC	/240D	UCOMP	130
554		DC	/6517		
555		DC			138
556		DC			
557	AERD	DC	/1464	MRED	140
558		DC	/5100		
559		DC	/1499	MWRT	148
560		DC	/98C0		
561		DC	/140D	MCOMP	150
562		DC	/6517		
563	AFIO	DC	/1418	MFIO	158
564		DC	/9580		
565		DC	/1425	MIOA1	160
566		DC	/6049		
567	AIOAF	DC	/1425	MIOAF	168
568		DC	/6046		
569		DC	/1425	MIOFX	170
570		DC	/61A7		
571		DC	/1425	MIOIX	178
572		DC	/6267		
573		DC	/1425	MIOF	180
574		DC	/6180		
575	IOI	DC	/1425	MIOI	188
576		DC	/6240		
577	SUBSC	DC	/2290	SUBSC	190
578		DC	/2883		
579		DC			198
580		DC			
581	BCKSP	DC	/020D	BCKSP	1A0
582		DC	/2897		
583	EOF	DC	/0558	EOF	1A8
584		DC	/6000		
585	REWND	DC	/1916	REWND	1B0
586		DC	/6544		
587		DC	/228D	STOP	1B8
588		DC	/65C0		
589		DC	/1706	PAUSE	1C0
590		DC	/4885		
591	RSIGN	DC	/2255	SNR	1C8
592		DC	/9000		
593		DC	/2290	SUBIN	1D0
594		DC	/2255		
595	UFIO	DC	/2418	UFIO	1D8
596		DC	/9580		
597		DC	/2425	UIOA2	1E0
598		DC	/6049		
599		DC	/2425	UIOAF	1E8

600		DC	/6046		
601		DC	/2425	UIOIX	1F0
602		DC	/61A7		
603		DC	/2425	UIOIX	1F8
604		DC	/6267		
605		DC	/2425	UIO_	200
606		DC	/6180		
607	UIOI	DC	/2425	UIOI	208
608		DC	/6240		
609	MDFIO	DC	/1410	MDFIO	210
610		DC	/6256		
611		DC	/1411	MDRED	218
612		DC	/9144		
613		DC	/1412	MDWRT	220
614		DC	/6663		
615		DC	/1410	MDCOM	228
616		DC	/3594		
617		DC	0		
618		DC	0		
619	*				RESERVED TO KEEP CODE
620	*				DIFF CONSTANT
621	*				BETWEEN DISK IO
622	*				ROUTINES AND OTHERS
623					SHOULD NOT BE USED FOR OTHER
624					NAME, IS CHANGED FOR SINGLE DEVICE
625		DC	/1410	MDAI	238
626		DC	/1240		
627		DC	/1410	MDAF	240
628		DC	/1180		
629		DC	/1410	MDFX	248
630		DC	/1410		
631		DC	/69C0	MDIX	250
632		DC	/1410		
633		DC	/99C0	MDF	258
634		DC	/1410		
635	MDI	DC	/6000	MDI	260
636		DC	/1410		
637	MDFND	DC	/9000	MDFND	268
638		DC	/1410		
639		DC	/6544		270
640	TABLL	DC	/050C	ECHR1	
641		DC	/8649		
642		DC	/060C	FCHR1	
643		DC	/8649		
644		DC	/23A1	TYPEN	
645		DC	/7155		
646		DC	/0859	HOLEB	
647		DC	/3142		
648		DC	/1404	MAGT	
649		DC	/78C0		
650		DC	/1765	PRNTN	
651		DC	/58D5		
652		DC	/0509	EBPRI	
653		DC	/7663		
654		DC	/1705	PAPIN	
655		DC	/78D5		
656		DC	/1705	PAPEB	
657		DC	/7142		
658		DC	/0305	CARDN	
659	EOTAB	DC	/9115		
			/0000	NOT USED	

```

660          DC          /0000
661      *
662      TABLM DC          TABLL 4      SUBR NAME GROUP ADDR
663          DC          TABLL 6      ADDRESS CONSTANT
664          DC          TABLL 8      ADDRESS CONSTANT
665          DC          TABLL 20     ADDRESS CONSTANT
666          DC          TABLL 10     ADDRESS CONSTANT
667          DC          TABLL 12     ADDRESS CONSTANT
668          DC          TABLL 4      ADDRESS CONSTANT
669          DC          TABLL 12     ADDRESS CONSTANT
670          DC          TABLL 14     ADDRESS CONSTANT
671          DC          TABLL 16     ADDRESS CONSTANT
672          DC          TABLL 18     ADDRESS CONSTANT
673          DC          TABLL 6      ADDRESS CONSTANT
674      *
675      *****
676      *
677      *
678      L1041 LD          3 STP-Z      LOAD ADDR OF SYM TBL ID
679          S          L          EOFST  SUB ADDR OF SYM TBL END
680          BSC          SKIP IF NOT END SYLBOL TBL
681          MDX          L2011        BR IF END SYMBOL TBL
682          LD          1 0          ELSE LD ID WORD
683      *
684      *          TEST IF CONSTANT
685      *          EXTENDED PREC CONST MAY HAVE BITS
686      *          IN ID-WORD TO MAKE IT SIMILAR TO
687      *          SUBPROGRAM NAMES
688          BSC          L          L1045, Z  BRANCH IF BIT 0 IS ON
689      *
690          AND          3 MASK2-Z     ELSE SAVE ONLY BIT 8
691          BSC          SKIP IF SUBPROGRAM NAME
692          MDX          L1045        BRANCH IF NOT
693      *
694      *          TEST IF SUBPROGRAM NAME IS THE
695      *          NAME OF THE COMPILED SUBPROGRAM
696      *          ITSELF
697      *          HAS /FFFF IN LAST WORD
698      *          IN THAT CASE DO NOT PRINT
699      *
700          LD          1 2          LD 3RD WD OF ENTRY
701          A          3 1          ADD 1 AND SEE IF ACC IS 0
702          BSC          SKIP IF DIFF
703          MDX          L1045        BRANCH IF SAME
704      *
705      *          TEST IF ARITH STMT FUNCT NAME
706      *          IN THAT CASE DO NOT PRINT
707      *
708          LD          1 0          LD SYM TBL ID WORD
709          AND          3 H0100-Z     SAVE ONLY BIT 7
710          BSC          L          L1045, Z  BR IF ARITH STMT FUNC NAM
711      *
712      *          GET NAME FROM SYMBOL TABLE
713      *          MOVE NAME TO PRINT AREA
714      *
715          BSI          L          MNAME      GO MOVE THE NAME
716      *
717      *
718      L1045 BSI          3 MSTP-Z     INCR SYMBOL TBL POINTER
719          MDX          L1041        LOOP TO CHECK NEXT

```

```

720 *                               SYMBOL TABLE ENTRY
721 *
722 L2011 LDX  I1 SOFS             INITIALIZE STRING POINTER
723 *
724 L2012 LD   1 0                 EXTRACT AND STORE STMT ID
725         AND 3 MASK5-Z          SAVE BITS 0 THRU 4
726         STO 3 STID=Z          KEEP STMT TYPE HERE
727         LD  1 0                 EXTRACT AND STORE NORM
728         AND 3 MASK6-Z          SAVE BITS 5 THRU 13
729         STO 3 NRM-Z           KEEP STMT NORM HERE
730         BSI 3 MOVEP-Z         MOVE POINTER
731 *
732 *                               TEST IF END-STATEMENT
733 L2021 LD   3 STID=Z           GET STMT TYPE CODE
734         S   3 ENDID=Z         SUB END STMT TYPE CODE
735         BSC L L2041, -        BR IF END
736 *
737 *
738 *                               TEST TO FIND STATEMENTS THAT
739 *                               CONSIST OF ONE WORD ONLY
740 *
741         LD   3 NRM-Z           GET THE STMT NORM
742         BSC L L2012, -        BR IF ONE=WD STMT
743 *
744 *
745         LD   3 STID=Z          GET STMT TYPE CODE
746         S   3 FTEST-Z         SUB FORMAT STMT TYPE CODE
747         BSC L L2022, -        BR IF FORMAT STMT
748 *
749 *                               TEST IF LOADER=OVERLAY SMT
750 *                               IN A SUBPROGRAM
751 *                               OR DEFINE FILE, DATA
752 *
753         LD   3 STID=Z          GET STMT TYPE CODE
754         S   3 DFILE=Z         SUB DEF FILE TYPE CODE
755         BSC L L2022, -        BR IF DEFINE FILE STMT
756         S   3 DATA=Z         SUB DATA STMT TYPE CODE
757         BSC L L2022, -        BR IF DATA STMT
758 *
759         LD   3 STID=Z          GET STMT TYPE CODE
760         S   3 HD800-Z         SUB INT O/P FMT TYPE CODE
761         BSC L L2031,Z         BR IF STMT INE, D800
762         LD   L SORF           GET SUBRTN OR FCTN IND WD
763         BSC L L2022,Z         BR IF LOADER OVERLAY STMT
764         MDX L2031             BR IF NOT
765 *
766 *
767 *                               MOVE POINTER TO NEXT STATEMENT
768 *
769 L2022 LD   3 NRM-Z           GET THE STMT NORM
770         BSC L L2012, -        BR IF NORM COUNT ZERO
771         BSI 3 MOVEP-Z         MOVE POINTER
772         MDX L2022             LOOP UNTIL STMT NORM ZERO
773 *
774 *
775 L2031 LD   1 0                 LD SYM TBL POINTER
776         BSC -                 SKIP IF PACKED INST/NAME
777         MDX L2033             BR IF NOT
778         AND 3 H7800-Z         SAVE BITS 1,2,3,4
779         BSC L L2032, -        BR IF NAME

```

```

780 *
781 *      CHANGE PACKED INSTRUCTION IF
782 *      NAME REFERS TO DIMENSIONED
783 *      VARIABLE
784 *
785      LD      1 0      LD SYM TBL POINTER
786      BSI    L GETST    GO GET SYM TBL ENTRY
787      AND    3 H1800-Z  SAVE BITS 3 AND 4 OF ID WD
788      BSC    L L2032, - BR IF NOT DIMENSIONED
789 *
790 *      CHANGE INSTRUCTION CODE
791 *      FROM LDL TO LDL1 ETC
792 *      WAS NOT DONE IN MACRO 1 BECAUSE
793 *      THE PHASE WAS SPACE CRITICAL
794 *
795      LD      1 0      LOAD PACKED INSTRUCTION
796      OR      3 H4000-Z  SET BIT 1 SO HEX DIGIT
797      STO    1 0      ONE IS 4 GTR, RE=STORE
798 *
799 *
800 L2032 BSI    3 MOVEP-Z  MOVE POINTER
801      LD      3 NRM-Z    GET WORD COUNT
802      BSC    L L2031,Z  BR IF NRM NOT ZERO
803      MDX    L2012      BR IF NRM ZERO
804 *
805 *
806 L2033 LD      1 0      LD SYM TBL POINTER
807      SLA    1          LOOK AT BIT 1
808      BSC    L L3011,-  BR IF ,CALL,
809 *
810 *      TEST IF HARDW=INSTR CODE
811 *      REFERS TO ,SPEC BSC L,
812 *      IF SO, MOVE POINTER ONE WORD
813 *      EXTRA SPEC BSC L IS FOLLOWED
814 *      BY A DISPLACEMENT VALUE RATHER
815 *      THAN A NAME
816 *
817      LD      1 0      LD SYM TBL POINTER
818      S      3 H5D00-Z  SEE IF SPEC BSC L
819      H L2034, - BR IF SPEC BSC
820 *
821 *
822      LD      1 0      LD SYM TBL POINTER
823      S      3 H5800-Z  SEE IF SPEC LDX L
824      BSC    L L2032,Z  BR IF LDX L1
825 *
826 *      THIS MOVE WILL EFFECT A DOUBLE MOVE
827 *
828 L2034 BSI    3 MOVEP-Z  MOVE POINTER BY 1
829      MDX    L2032      BR BACK AND MOVE 2ND TIME
830 *
831 *
832 L2041 LDX    L1 TABSS-EOTAB-2 TABLE LENGTH
833      LD      L CCWD    PUT CONT CARD WD INTO ACC
834      SLA    11        PUT BIT 11 IN SIGN
835      BSC    L EXIT,-  BR IF SYMT LIST NOT REQ
836 L2042 MDX    1 2      ELSE MOVE XR1 BY 2 AND
837      MDX    L2044      BR TO L2044
838 *
839 *      SKIP OCCURS AT END OF LIST

```

```

840 *
841 *      TEST IF HALF FILLED LINE REMAINS
842 *      TO BE PRINTED
843 *
844      LD      L      PCNT      GET PRINT COUNT
845      BSC     L      EXIT, -   BR IF PRINT COUNT ZERO
846      SRA     1          ELSE DIVIDE BY TWO
847      STO     L      WDCNT     AND CALL IT A WDCNT
848      LD      3      PAPIN-Z   GET THE STARTING PRINT ADD
849      STO     L      AREA     PUT THIS ADDR IN AREA
850      BSI     L      PRINT    PRINT A LINE IF NOT
851      BSC     L      EXIT     EXIT THE PHASE
852 *
853 *      PRINT ,TAGGED, SYSTEM SUBROUTINE
854 *      NAMES
855 *
856      L2044 LD      L1      EOTAB      LD A TABLE ENTRY
857      BSC     -          SKIP IF SUBR NAME TAGGED
858      MDX     L2042     BR IF NOT TO MOVE XR1
859      MDX     L1      EOTAB-1   SET TBL ENTRY ADDR-1 IN XR
860      SLA     0          NO OP FOR SKIPPING
861      BSI     L      MNAME     MOVE NAME TO PRINT AREA
862      MDX     L1      -EOTAB 1  RESTORE XR1 FOR TBL SEARCH
863      SLA     0          NO OP FOR SKIPPING
864      MDX     L2042     LOOP BACK AND MOVE XR1
865 *
866 *      TWO-WORD SUBPROGR CALL
867 *      OR SYSTEM SUBROUTINE CALL ENCOUNTERED
868 *
869      L3011 LD      1      0      GET SYM TBL POINTER WD
870      SRA     7          SAVE ONLY BITS 0 THRU 8
871 *
872      BSC     L      L3051, -   BR IF TWO WORD CALL
873 *
874 *      THIS IS A SYSTEMS SUBROUTINE CALL
875 *
876 *      TEST IF LAST WORD IN STATEMENT
877      LD      3      NRM-Z     GET NORM FROM ID WORD
878      S       3      FOUR-Z   REDUCE COUNT BY ONE UNIT
879      BSC     L      L3012,    BR IF LAST
880 *
881 *      TEST IF ,CALL SUBIN,
882 *      TEST MUST BE MADE HERE SINCE ITS
883 *      FOLLOWING ARGUMENTS COULD BE
884 *      MISTAKEN FOR DIMENSIONED
885 *      VARIABLE NAMES, AT OCCASION
886 *
887      LD      1      0      GET SYM TBL POINTER WD
888      S       3      H1D00-Z   SEE IF IT'S A CALL SUBIN
889      BSC     L      L3012, -   BR IF ,SUBIN,
890 *
891 *
892 *      IF FOLLOWED BY A DIMENSIONED NAME,
893 *      INCREMENT CALL CODE
894 *      CORRECTION DONE HERE BECAUSE
895 *      MACRO 1 PHASE APPROACHING CRITICAL
896 *      SIZE
897 *
898      LD      1      1      LD WD AFTER SYM TBL PNTR
899      BSC     L      L3012,-   BR IF NOT FOLLOWED BY

```

```

900 *                               NAME OR PACKED INSTRUCTIO
901   AND   3 H7800-Z   ELSE SAVE BITS 1,2,3,4
902   BSC   L  L3012,Z   BR IF PACKED INSTR
903 *
904 *                               TEST IF NAME DIMENSIONED
905   LD     1 1         LD WD AFTER SYM TBL PNTR
906   BSI   L  GETST     GO GET NEXT ID WD
907   AND   3 H1800-Z   SAVE BITS 3 AND 4
908   BSC   L  L3012, - BR IF NAME NOT DIMENSIONED
909 *
910 *                               NAME IS DIMENSIONED,
911 *                               CHANGE CALL CODE
912 *
913   LD     1 0         LOAD ID WORD
914   A     3 H0080-Z   TURN BIT 8 ON
915   STO   1 0         RE=STORE THE ID WD
916 *
917 *
918 *                               TAG SYSTEMS SUBROUTINE NAME IN TABLE
919 *
920 L3012 LD     1 0         LD SYM TBL POINTER WORD
921       SRA    6         SAVE BITS 0 THRU 9
922       STO    L3013 1   STORE THEM BEHIND 3013
923 L3013 LD    L2 *-*     PUT THESE BITS INTO XR2
924       BSI   3 TAGSR-Z  AND GO TAG THE NAME
925 *
926 *                               TEST IF SYSTEM SUBROUTINE
927 *                               HAS SPECIAL ARGUMENT LIST TO
928 *                               MOVE POINTER PAST THIS ARG LIST
929 *
930   LD     1 0         LD SYM TBL POINTER WORD
931   S     3 H1600-Z   COMPARE TO FIOAI CODE
932   BSC   L  L3015, - BR IF CALL FIOAI
933 *
934   LD     1 0         LD SYM TBL POINTER WD
935   S     3 H1680-Z   COMPARE TO FIOAF CODE
936   BSC   L  L3015, - BR IF CALL FIOAF
937 *
938   LD     1 0         LD SYM TBL POINTER WD
939   S     3 H1D00-Z   COMPARE TO SUBIN CODE
940   BSC   L  L3041, - BR IF CALL SUBIN
941 *
942   LD     1 0         LD SYM TBL POINTER WD
943   S     3 H1900-Z   COMPARE TO SUBSC CODE
944   BSC   L  HERE, -  BR IF NOT ,CALL SUBSC,
945   LD     1 0         LD SYM TBL POINTER WD
946   S     3 H1E00-Z   COMPARE DIFF CODE
947   BSC   Z          SKIP IF UIOAI
948   S     3 H0080-Z   COMPARE DIFF CODE
949   BSC   Z          SKIP IF UIOAF
950   S     3 H0500-Z   COMPARE DIFF CODE
951   BSC   Z          SKIP IF MDAI
952   S     3 H0080-Z   COMPARE DIFF CODE
953   BSC   L  L2032,Z  BRANCH IF NONE OF ABOVE
954   MDX   L3015      ELSE GO MOVE POINTER
955 *
956 *                               MOVE POINTER PAST ARGUM LIST
957 *                               OF ,CALL SUBSC,
958 *
959 HERE BSI   3 MOVEP-Z  MOVE POINTER AT LEAST 4

```



```

960          BSI      3 MOVEP-Z  WORDS THEN IN
961 L3042  BSI      3 MOVEP-Z  GROUPS OF 2 UNTIL BIT
962          BSI      3 MOVEP-Z  0 IS FOUND TO BE ZERO
963          LD       1 0       LD A STMT STRING WORD
964          BSC     L  L2032,Z  BR IF PAST ARG STRING
965          MDX          L3042  BR IF NOT
966 *
967 *
968 *          MOVE POINTER PAST TWO ARGUMENTS
969 *
970 L3015  BSI      3 MOVEP-Z  MOVE STRING POINTER
971          BSI      3 MOVEP-Z  TWO WORDS AND
972          BSC     L  L2032  RETURN TO CALL PT
973 *
974 *          MOVE POINTER PAST ARGUM LIS
975 L3041  SLA          0       A NU OP
976 L3016  BSI      3 MOVEP-Z  GO MOVE STRING POINTER
977          LD       1 0       LD THE STRING WORD
978          S        3 ONE-Z   CONST 1 IS ENDOFLIST IND
979          BSC     L  L3016,Z  BR IF NOT END OF
980 *          ARGUMENT LIST
981          BSC     L  L2032  BR IF END OF ARGUM LIST
982 *
983 *          TWO-WORD CALL ENCOUNTERED
984 *          IF CALL IS TO ARITH STMT FUNCT,
985 *          CHANGE TO BSI L
986 *
987 L3051  LD       1 1       LD SECOND WD OF CALL
988          BSI     L  GETST   GO GET SYM TBL ENTRY
989          AND     3 H0500-Z  SAVE ONLY BITS 5 AND 7
990          BSC     L  L2032,  BR IF NOT ARITH STMT FUNC
991 *          OR DUMMY VARIABLE NAME
992          LD     3 H4700-Z  CODE FOR ,BSC L
993          STO     1 0       REPLACE ,CALL, WITH ,BSI L
994          BSC     L  L2032  GO MOVE POINTER
995 *
996 *          PRINT HEADER IF NECESSARY
997 *          BUILD PRINT LINE
998 *
999 MNAME  DC      ***      LINK ADDR WORD
1000          BSI     L  PBLNK  GO PRINT A BLANK LINE
1001          BSI     L  HTEST  GO UNPACK AND PRINT MSG
1002          DC      9       MSG WORD COUNT
1003          EBC     ,CALLED  SUBPROGRAMS,  MSG ITSELF
1004          LDX     L2 RWM-1  SET XR2 TO NEXT ADDR
1005          STX     2 MNAME 2  REPLACE HTEST WITH NEXT AD
1006 RWM     BSI     2 TOPAB   GO SET UP TO PRINT MSG
1007          LD      1 1       LOAD SUBROUTINE NAME
1008          BSI     1 TOPA    GO PUT CHAR IN BUF STRING
1009          LD      1 1       LD SUBR NAME AGAIN
1010          SLA     6       GET 2ND CHAR OF NAME
1011          BSI     1 TOPA    PUT 2ND CHAR ON BUF STRING
1012          LD      1 2       GET 2ND WORD OF NAME
1013          RTE     16      SHIFT IT INTO THE Q
1014          LD      1 1       PUT FIRST WORD BACK IN ACC
1015          SLT     12      SET UP 3RD CHAR IN ACC
1016          BSI     1 TOPA    PUT 3RD CHAR ON BUF STRING
1017          LD      1 2       GET 2ND WORD OF NAME
1018          SLA     2       SET UP 4TH CHAR IN ACC
1019          BSI     1 TOPA    PUT 4TH CHAR ON BUF STRING

```

```

1020      LD      1 2      GET 2ND WORD AGAIN
1021      SLA     8      SET UP 5TH CHAR IN ACC
1022      BSI     TOPA    PUT 5TH CHAR ON BUF STRING
1023      BSI     TOPAB   INSERT TWO EBC BLANKS
1024      BSI     TOPAB   AFTER SUBROUTINE NAME
1025      *
1026      *          INCREMENT PRINT COUNT
1027      LD      PCNT    GET PRINT COUNT INTO ACC
1028      A      3 H0008-Z  ADD ONE UNIT TO IT
1029      STO     PCNT    RE=STORE SUM IN PCNT
1030      *
1031      *      TEST IF LINE FULL
1032      S      3 C120-Z  IS PRINT COUNT 120 YET
1033      BSC    I MNAME, Z  RETURN IF LINE NOT FULL
1034      LD      3 PAPIN-Z  LD INITIAL BUF ADDR
1035      STO     3 PAP-Z    PUT IT IN PRINT AREA PNTR
1036      STO     L AREA    ALSO PUT IT IN AREA
1037      LD      3 D60-Z   GET A WORD COUNT
1038      STO     L WDCNT   AND SET WDCNT TO IT
1039      BSI     L PRINT   PRINT A LINE
1040      SLA     16        ZERO THE ACC
1041      STO     PCNT    ZERO PRINT COUNT
1042      BSI     BLKPA   GO PUT EBC BLANKS IN BUFI
1043      BSC    I MNAME   RETURN THRU LINK ADDR
1044      *
1045      *          SUBROUTINE
1046      *          CHAR IN A=REG TO PRINT AREA UNCHANGE
1047      *
1048      TOPAU  DC     ***      LINK ADDR WORD
1049      STO     STOCH   STORE CHARACTER HERE
1050      TOPAX  LD     STOCH   LOAD THE MSG CHAR
1051      STO     I PAP    PUT IT IN PRINT AREA PNTR
1052      MDX    L PAP,1  MOVE PRINT AREA POINTER
1053      BSC    I TOPAU  RETURN THRU LINK ADDR
1054      *
1055      *          SUBROUTINE
1056      *          CONVERT CHAR IN A=REG INTO
1057      *          EBC-CODE, THEN MOVE TO PRINT AREA.
1058      *
1059      TOPA   DC     ***      LINK ADDR WORD
1060      AND    H3F00   SAVE BITS 2 THRU 7
1061      BSC    L TOPA2, - BR IF ACC IS ZERO
1062      *
1063      *          NOTE
1064      *          IF OTHER SPECIAL CHARACTERS THAN
1065      *          BLANK ARE EXPECTED, TESTING SHOULD
1066      *          BE HERE
1067      OR     HC000   TURN BITS 0 AND 1 ON
1068      TOPA1  STO     STOCH   STORE CHARACTER HERE
1069      LD     TOPA    MOVE LINK ADDR WORD
1070      STO     TOPAU   RE=STORE IT HERE
1071      MDX    TOPAX   BR TO GET MSG CHAR
1072      TOPA2  LD      3 H4000-Z  PUT AN EBC BLANK IN ACC
1073      MDX    TOPA1   GO STORE IT
1074      *
1075      TOPAB  DC     ***      LINK ADDR WORD
1076      SLA     16        ZERO THE ACC
1077      BSI     TOPA    BRANCH UP
1078      BSC    I TOPAB  RETURN THRU LINK ADDR
1079      *

```

```

1080 *          CONSTANTS
1081 PCNT  DC      ***  WHERE PRINT COUNT IS
1082 H3F00 DC      /3F00  MASK FOR SAVING BITS
1083 HC000 DC      /C000  MASK FOR ORING IN BITS
1084 STOCH DC      ***  TEMP STORAGE FOR MSG CHAR
1085 *
1086 *          SUBROUTINE
1087 *          TO PRINT BLANK LINE
1088 *
1089 PBLNK DC      ***  LINK ADDR WORD
1090     LD    3 H4000-Z  PUT AN EBC BLANK IN ACC
1091     STO   L  BUF     BLANK A WD IN BUF
1092     STO   L  BUF+1   BLANK ANOTHER
1093     LD    3 PAPIN-Z  GET BUF BEGIN ADDR
1094     STO   L  AREA    PUT IT IN AREA
1095     LD    3 ONE-Z    GET CONSTANT FOR WDCNT
1096     STO   L  WDCNT   SET WDCNT TO ONE
1097     BSI   L  PRINT   GO PRINT THE BLANK LINE
1098     BSC   I  PBLNK   GO PRINT BLANK LINE
1099 *
1100 * PUT EBC BLANKS IN BUF STRING
1101 BLKPA DC      ***  LINK ADDR WORD
1102     LD    3 H4000-Z  PUT EBC BLANK IN ACC
1103     LDX   I3 WDCNT   PUT WD COUNT INTO XR3
1104 BLKP  STO   L3 BUF-1  BLANK A BUF WORD
1105     MDX   3  -1      DECREMENT XR3 BY 1
1106     MDX   BLKP      LOOP TO BLANK ANOTHER
1107     LDX   L3 Z       RESTORE XR3 TO PT VAL
1108     BSC   I  BLKPA   RETURN THRU LINK ADDR
1109 *
1110 *          EXIT ROUTINE
1111 *
1112 EXIT  SLA      0
1113 *
1114 *
1115     BSI   L  ROLRX   CALL DOWN PHASE 25
1116     DC    25        NEXT PHASE NUMBER
1117     BSS   OVERL-**+320*3  PHASE-24 PATCH AREA
1118     END   NEQ

```