

```

1  *   GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2  *****
3  *
4  *   PROGRAM NAME   FPH-13
5  *
6  *   MODEL NUMBER   8F013
7  *
8  *   PURPOSE        FORTRAN PHASE-13
9  *
10 *   PROGRAMMER     DICK WALLMANN
11 *
12 *****   REVISION LIST   *****
13 *
14 *   RV DATE        SCC   BY   REASON FOR CHANGE
15 *   --  -----  - - - - -  - - - - -  - - - - -
16 *
17 *   01 11/16/70  NCNE  RPH INITIAL RELEASE
18 *
19 *****
20 *****
21         HDNG      MPX FORTRAN ** ASCAN II
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 *   * CHECKS FIND, READ, WRITE, AND GO TO STMENTS
27 *   * FOR CORRECT SYNTAX, STATEMENT NUMBERS AND
28 *   * REFERENCES, AND VARIABLES.
29 *   * DETECTS IMPLIED DO LOOPS IN READ AND WRITE
30 *   * STATEMENTS. GENERATES THE NECESSARY
31 *   * INDICATORS.
32 *
33 *ENTRY POINTS-
34 *   * START-PHASE 13 IS READ INTO CORE BY PHASE
35 *   * 12 VIA ROLPX. EXECUTION IS BEGUN
36 *   * AT LOCATION START.
37 *
38 *INPUT-
39 *   * THE STATEMENT STRING
40 *   * THE SYMBOL TABLE
41 *   * THE FORTRAN COMMUNICATION AREA
42 *
43 *OUTPUT-
44 *   * THE STATEMENT STRING
45 *   * THE SYMBOL TABLE
46 *   * THE FORTRAN COMMUNICATION AREA
47 *
48 *EXTERNAL REFERENCES-
49 *   * SUBROUTINES
50 *   * ROLPX
51 *   * OTHER FORTRAN PHASES
52 *   * NONE
53 *
54 *EXITS-
55 *   * NORMAL-
56 *   * PHASE 14 IS CALLED VIA POLPX AND CONTROL
57 *   * IS PASSED TO IT.
58 *   * ERRORS-
59 *   * OVERLAP-PROCESSING IS HALTED, THE ERROR

```

```

60 *          WORD IS SET IN FCOM, PHASE 14 IS
61 *          CALLED VIA ROLRX.
62 *          SYNTAX- THE ERRONEOUS STATEMENT IS REPLACED
63 *          WITH AN ERROR NUMBER, PROCESSING
64 *          CONTINUES, PHASE 14 IS CALLED
65 *          VIA ROLRX.
66 *          THE ERRORS DETECTED ARE NUMBERS
67 *          43,44,45,46,47,48,49,50 AND 68
68 *
69 *TABLES/WORK AREAS-
70 *   * THE STATEMENT STRING
71 *   * THE SYMBOL TABLE
72 *   * THE FORTRAN COMMUNICATION AREA
73 *
74 *ATTRIBUTES-N/A
75 *
76 *NOTES-
77 *   THE SWITCHES USED IN PHASE 2 FOLLOW. IF NON-
78 *   ZERO THE SWITCH IS TRANSFER T. IF ZERO, THE
79 *   SWITCH IS NORMAL N.
80 *   * SW1-DIMENSION NAME SUBSCRIPTED
81 *       T NOT SUBSCRIPTED
82 *   * SW2-READ OR WRITE STATEMENT
83 *       T READ STATEMENT FOUND
84 *       N WRITE STATEMENT FOUND
85 *   * SW3-EQUAL SIGN
86 *       T EQUAL SIGN FOUND
87 *   * SW4-DIMENSIONED NAME
88 *       T NAME DIMENSIONED
89 *   * FND SW-FIND STATEMENT
90 *       T FIND STATEMENT FOUND
91 *****
92      HONG      MPX FORTRAN ** ASCAN II
93      ARS      REF CORE
94 *
95 *          SYSTEM AND FORTRAN EQUATES
96 *
97 MEMRY EQU      FFFF CORE      MAXIMUM CORE SIZE
98 PHSI7 EQU      4*320              MAXIMUM PHASE SIZE
99 OVERL EQU      MEMRY-PHSI7        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 *
104 *          FORTRAN COMMUNICATION AREA
105 *
106          ORG      FCOM      FORTRAN COMM AREA
107 SCFS BSS        1          START OF STRING
108 ECFS BSS        1          END OF STRING
109 SCFST BSS       1          START OF SYMPOLE TABLE
110 SOFNS BSS       1          START OF NON-STMNT NOS.
111 SOFXT BSS       1          START OF SUBSC TEMPS
112 SOFGT BSS       1          START OF GENERATED TEMPS
113 ECFST BSS       1          END OF SYMBOL TABLE
114 COMON BSS       1          NEXT AVAILABLE COMMON
115 CSIZE BSS       1          SIZE OF COMMON
116 ERROR BSS       1          OVERLAP ERRORP
117 FNAME BSS       1          PROGRAM NAME WD 1
118          BSS       1          PROGRAM NAME WD 2
119 SORF BSS        1          SUBR (-) OR FUNC (+)

```

```

120 CCWD BSS 1 CONTROL CARD WD
121 * BIT 15 TRANSFER TRACE
122 * BIT 14 ARITHMETIC TRACE
123 * BIT 13 EXTENDED PRECISION
124 * BIT 12 LIST SYMBOL TABLE
125 * BIT 11 LIST SUBPROGRAM NAMES
126 * BIT 10 LIST SOURCE PROGRAM
127 * BIT 9 ONE WORD INTEGERS
128 IOCS BSS 1 IOCS CONTROL CARD WORD
129 *
130 * SEE PHASE ONE FOR BIT PATTEFNS
131 *
132 DFCNT BSS 1 DEFINED FILE COUNT
133 *
134 LCOMN BSS 2 SIZE OF INSKEL COMMON
135 *
136 ICCER BSS 2 IOCS CONTROL CARD ERROR
137 *
138 BSS 2 SYSTEM LOADER USE
139 *
140 * END OF FORTRAN COMMUNICATION
141 * AREA
142 *
143 HDNG MPX FORTRAN ** ASCAN II
144 ORG OVERL PHASE ENTRY
145 START LD ERROR TEST FOR OVERLAP ERROR
146 BSC L ORGIN, - BRANCH IF NO ERROR
147 *
148 * TRANSFERS TO THE ROL ROUTINE
149 * TO LOAD THE NEXT PHASE
150 OVERF BSI L ROLRX CALL DOWN PHASE 14
151 DC 14 NEXT PHASE NUMBER
152 *
153 * MOVES THE STMNT STRING ADJACENT
154 * TO THE SYM TRL. INITIALIZES
155 * THE I/P PT. CHECKS FOR THE
156 * TRACE AND IOCS INDICATORS IN
157 * THE FORTRAN COMMUNICATIONS AREA
158 ORGIN LD EOFS GET SIZE OF
159 S SOFS STRING - 1
160 STO * 1 INITIALIZE STRING
161 LDX L3 ** SIZE COUNTER. MODIFIABLE
162 MDX 3 1 SIZE OF STRING
163 LDX I2 EOFS LOOP INPUT PT
164 LDX I1 EOFST LOOP OUTPUT PT
165 ORGIP LD 2 0 LOAD WORD AND MOVE IT
166 STO 1 2 NEXT TO SYMBOL TABLE
167 MDX 1 -1 MOVE POINTERS
168 MDX 2 -1
169 MDX 3 -1 DECREMENT COUNTER
170 MDX ORGIP CONTINUE LOOP
171 MDX 1 3 INITIALIZE STRING INPUT PT
172 *
173 * INITIAL PHASE
174 LDX I2 SOFS INITIALIZE STRING O/P PT
175 *
176 * TRACE CONTROL RECORD PRESENT
177 LD CCWD IS THERE A TRACE
178 SLA 14 DEVICE REQUIRED
179 BSC L XY1, - BRANCH IF NO

```

```

180 *
181 *
182 LD SORF IS TRACE DEVICE PRESENT
183 OR IOCS OR IN MAINLINE PROGRAM
184 BSC L XY1,7 BRANCH IF YES
185 *
186 * PUT OUT ERROR NO. 50
187 LD SAERT LOAD ERROR ID WORD
188 BSI L CHECK OUTPUT ERROR ID WORD
189 LD S50TT LOAD ERROR NUMBER
190 BSI L CHECK OUTPUT ERROR NUMBER
191 MDX XY1 BEGIN BODY SCAN
192 *
193 * CONSTANTS
194 SAERT DC /A008 ERROR ID WORD
195 S50TT DC 50 ERROR NO. 50
196 S0001 DC /0001 ONE
197 S07FF DC /07FF NORM MASK
198 *
199 * INITIALIZES TO SCAN THE
200 * STATEMENT BODY
201 XY1 LD 1 0 LOAD STMT ID WORD
202 AND S07FF CLEAR OUT ID TYPE
203 SRA 2 RIGHT-JUSTIFY NORM
204 STO L XY27 1 SAVE NORM
205 STX L1 XT1 SAVE INPUT POINTER
206 A L XT1 ADD INPUT POINTER TO GET
207 STO L NXTID ADDRESS OF NEXT STMT
208 STX L2 IDSV2 SAVE OUTPUT PT
209 STX 1 IDSV1 SAVE INPUT PT
210 BSI L MOVE1 OUTPUT WORD, MOVE PT
211 LD 1 -1 LOAD STMT ID WORD
212 EOR S0001 TEST FOR STMT NUMBER
213 BSC L XY2,E BRANCH NO STMT NUMBER
214 BSI L MOVE1 OUTPUT WORD, MOVE PT
215 MDX L XY27 1,-1 DECREMENT NORM COUNTER
216 *
217 *
218 * CHECKS FOR THE END STMT
219 XY2 LD L *-* LOAD STMT ID WORD
220 IDSV1 EQU XY2 1 ADDRESS OF STMT ID WORD
221 * *MODIFIABLE
222 SPA 11 RIGHT-JUSTIFY ID TYPE
223 S L TEND IS IT END STMT
224 BSC L XY3,7 BRANCH IF NOT
225 *
226 *
227 XY34 MDX 2 -1 GET NEW END OF
228 STX L2 EOF8 STRING ADDRESS
229 MDX OVERF GO TO NEXT PHASE
230 *
231 *
232 * DETECTS FIND STMTS
233 XY4R S L TFIND IS IT FIND STMT
234 BSC L XY24,7 BRANCH IF NOT
235 STX 0 FND8W SET FIND STMT SWITCH
236 MDX NSW2 CHECK FOR DEVICE
237 *
238 *
239 * DETECTS READ STMTS

```

```

240 XY3 S L TREAD IS IT READ STMNT
241 BSC L XY4,Z BRANCH IF NOT
242 *
243 *
244 STX L0 SW2 SET SWITCH TO READ
245 MDX XY6 NORMALIZE FIND SWITCH
246 *
247 *
248 * DETECTS WRITE STMNTS
249 XY4 S L TWRIT IS IT WRITE STMNT
250 BSC L XY4R,Z BRANCH IF NOT
251 *
252 *
253 STO L SW2 SET SWITCH TO WRITE
254 *
255 *
256 * CHECKS THE SORF WORD IN THE
257 * FORTRAN COMMUNICATIONS AREA FOR
258 * A SUBPROGRAM INDICATOR
259 * CHECKS FOR THE PRESENCE OF IOCS
260 * INDICATORS IN A MAINLINE
261 * COMPILATION
262 XY6 STO FND SW NORMALIZE FIND SWITCH
263 NSW2 LDX 3 44 PUT ERROR NO. 44
264 STX L3 ERNO IN ERNO
265 *
266 *
267 LD L SORF IS THIS SUBROUTINE
268 BSC L XY6A,Z BRANCH IF YES
269 *
270 *
271 LD L IOCS CHECK FOR VALID
272 AND HD1FF I/O DEVICE
273 BSC L XY6A,Z BRANCH IF VALID DEVICE
274 *
275 *
276 XY6R LDX 3 45 PUT ERROR NO. 45
277 STX L3 ERNO IN ERNO
278 BSC L XY7 PUT ERROR STMNT ON STRING
279 *
280 *
281 * CHECKS THE SYNTAX OF FIND, READ
282 * AND WRITE STATEMENTS
283 * CHECKS VARIABLE NAMES AND
284 * INTEGER CONSTANTS FOR VALIDITY
285 * CHECKS FOR A FORMAT STMNT
286 * NUMBER REFERENCE
287 XY6A LD 1 0 LOAD WORD
288 S L TLP IS CHAR LEFT PARENTHESIS
289 BSC L XY7,Z BRANCH IF NOT
290 *
291 * CALL PUT I/O OPERATOR
292 STX 2 IOOFS SAVE OUTPUT POINTER
293 LD IOOF LOAD I/O OPERATOR
294 BSI L PUT PUT ON OUTPUT STRING
295 *
296 *
297 LD 1 0 LOAD WORD
298 BSI L CHECK PUT ON OUTPUT STRING
299 *

```

```

300 *
301 MDX 1 1 MOVE INPUT POINTER
302 *
303 *
304 LD 1 0 LOAD WORD
305 BSI L GETID GET SYM TBL ID WORD
306 AND S580E CHECK TO SEE IF
307 EOR S4000 IT IS INTEGER
308 BSC L XY7,7 BRANCH IF NOT
309 *
310 *
311 BSI L MOVE1 OUTPUT WORD, MOVE POINTER
312 *
313 *
314 LD 1 0 LOAD WORD
315 S TCMA IS CHARACTER COMMA
316 BSC L XY5,7 BRANCH IF NOT
317 *
318 *
319 LD L IOCS IS THERE I/O DEVICE OTHER
320 AND HD17F THAN DISK OR IS THIS
321 OR L SORF SUBROUTINE
322 BSC L XY6R, - BRANCH IF NEITHER
323 * IN A MAINLINE ONLY.
324 *
325 *
326 BSI L MOVE1 OUTPUT WORD, MOVE POINTER
327 *
328 *
329 LD 1 0 LOAD WORD
330 BSI L GETID GET SYM TBL ID WORD
331 SLA 9 IS IT FORMAT STATEMENT
332 BSC L XY9, Z BRANCH IF YES
333 *
334 *
335 LDX 3 46 PUT ERROR NO. 46
336 STX 3 ERNC IN ERNC
337 BSC L XY7 PUT ERROR STMT ON STRING
338 *
339 HD1FF DC /D1FF MASK FOR DEVICE CHECK
340 FNDSW DC 0 FIND SWITCH
341 COLON DC /32 COLON
342 HD17F DC /D17F NON-DISK IOCS MASK
343 DI00P DC /30 DISK I/O OPERATOR
344 APOST DC /42-712 APOSTROPHE
345 *
346 *
347 * CHECKS FOR THE APOSTROPHE
348 * MARK - CHANGES IT TO A COMMA
349 * OPERATOR
350 XY5 S APOST IS CHARACTER APOSTROPHE
351 BSC L XY8,Z BRANCH IF NOT
352 *
353 *
354 LD TCOP REPLACE APOSTROPHE ON
355 STO 1 0 INPUT STRING WITH COMMA
356 *
357 *
358 LD L IOCS REMOVE ALL BITS FROM IOCS
359 SLA 8 WORD EXCEPT BIT FOR DISK

```

```

360          SRA      15          IS DISK DEVICE PRESENT OR
361          OR      L  SORF      IS THIS SUBROUTINE
362          BSC     L  XY6R, -   BRANCH IF NEITHER
363 *
364 *
365 *
366          LD      L  DFCNT     LOAD FILE COUNT
367          OR      L  SORF      COMBINE WITH SORF
368          BSC     L  SKOOL,Z    BRANCH IF HAVE EITHER
369 *
370 *
371          LDX     3  70         PUT ERROR NO. 70
372          STX    L3 ERNO       INTO ERNO
373          BSC     L  XY7        PUT ERROR STMT ON STRING
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 SKOOL LD      1  0           LOAD WORD
382          S      COLON        IS CHARACTER COLON
383          BSC     L  KOLON,+ - BRANCH IF YES                V1M
384          BSI     L  MOVE1     OUTPUT WORD, MOVE POINTER
385          MDX          SKOOL    GET NEXT WORD
386 REPL LD      RPARN         LOAD RIGHT PARENTHESIS
387          STO     1  0         PUT IN PLACE OF COLON
388          LD      DICOP       REPLACE I/O OPERATOR WITH
389          STO     L  *-*       DISK I/O OPERATOR
390 ICOPS EQU     *-1          ADDRESS OF I/O OPERATOR
391 *
392          MDX          CLMV1    CONTINUE STMT CHECK
393 KOLON LD      1  -1          V1M
394          S      ICMA         IS IT A COMMA                V1M
395          BSC     L  REPL,7    BRANCH NO                    V1M
396          MDX          REPL-3   YES                          V1M
397 *
398 *
399 XY9 BSI     L  MOVE1     OUTPUT WORD, MOVE POINTER
400 *
401 *
402 XY8 LD      1  0           LOAD WORD
403          S      TRP         IS CHAR RIGHT PARENTHESIS
404          BSC     L  XY7,7    BRANCH IF NOT
405 *
406 *
407 *
408 *
409 CLMV1 BSI     L  MOVE1     OUTPUT WORD, MOVE POINTER
410 *
411 *
412          LD      FNDSW       IS THIS FIND STMT
413          BSC     L  XY8A, -   BRANCH IF NOT
414          LD      1  0         IS CHARACTER SEMI-COLON
415          BSC     L  XY7,Z    BRANCH IF NOT
416          BSC     L  XY11     CORRECT STMT NORM
417 XY8A LD      SW2          IS IT READ STMT
418          BSC     -          SKIP IF YES
419          LD      1  0         IS IT SEMI-COLON

```

420	BSC	L	CKCMA,7	BRANCH IF NOT
421	LD	1	-3	IS IT UNFORMATTED WRITE
422	S		TLP	WITH NO LIST
423	BSC	L	XY7, -	BRANCH IF YES
424	MDX		CKCMA	CONTINUE
425	*			
426	CKCMA	LD	TCCF	PUT COMMA ON
427	BSI	L	CHECK	OUTPUT STRING
428	*			
429	*			
430	LDX	3	47	PUT ERROR NO. 47
431	STX	L3	ERNO	INTO ERNO
432	MDX		XY10	START LIST CHECK
433	*			
434	*			CONSTANTS
435	XT1	DC	0	TEMPORARY STORAGE
436	SW2	DC	0	SWITCH 2
437	TEND	DC	/02	END ID
438	TREAD	DC	/12-/02	READ ID
439	TWRIT	DC	/11-/12	WRITE ID
440	TFIND	DC	/10-/11	FIND ID
441	ERNO	DC	0	ERROR NUMBER
442	TLP	DC	/10	LEFT PARENTHESIS
443	ICOP	DC	/26	I/O OPERATOR
444	S4000	DC	/4000	MASK TO CHECK FOR INTEGER
445	TRP	DC	/02	RIGHT PARENTHESIS
446	RPARN	EQU	TRP	RIGHT PARENTHESIS
447	ICMA	DC	/12	COMMA
448	TCMA	EQU	ICMA	COMMA
449	TCOP	EQU	ICMA	COMMA
450	TRGLP	DC	/22-/10	SPECIAL SKIP INDICATOR
451	TWO	DC	2	TWO
452	S5BDE	DC	/5BDE	MASK FOR INTEGER VARIABLE
453	TRGOP	DC	/22	SPECIAL SKIP INDICATOR
454	DOAOP	DC	/2A	DOA INDICATOR
455	ALL	DC	0	INDEXING PARAMETER COUNTER
456	HLQSP	DC	/02-/12	RIGHT PARENTHESIS
457	FOUR	DC	4	FOUR
458	SF802	DC	/F803	MASK TO CLEAR OUT NORM
459	S0020	DC	/0020	VARIABLE DEFINED BIT MASK
460	SFFDE	DC	/FFDE	MASK FOR INTEGER VARIABLE
461	SDBDE	DC	/DBDE	CON OR VAR INTEGER MASK
462	*			
463	*			
464	XY10	LD	1 0	IS CHAR SEMI-COLON
465	BSC	L	XY11, -	BRANCH IF YES
466	*			
467	*			
468	*			CHECKS A VARIABLE FOR DEFINITION
469	*			IF DEFINED, SETS THE INDICATOR
470	*			IN THE SYM TBL ID WORD
471	XY14	LD	1 0	IS IT SPECIAL CHARACTER
472	BSC	L	XY12,-	BRANCH IF YES
473	*			
474	*			
475	BSI	L	GETID	GET SYM TBL ID WORD
476	BSI	L	TESTV	TEST VALIDITY OF VARIABLE
477	*			
478	*			
479	LD		SW2	IS IT READ STMT



```

480          BSC  L  XY14A, -  BRANCH IF NOT
481 *
482 *
483          LD   3 0          LOAD SYM TBL ID WORD
484          OR   S0020        ADD DEFINED VAR BIT
485          STO  3 0          PUT BACK IN SYM TBL
486 *
487 *
488 XY14A BSI  L  NAME          POSITION PT TO NEXT VAR
489 *
490 *
491 XY17  LD   1 0          LOAD WORD
492          S    TCMA         IS CHARACTER COMMA
493          BSC  L  XY13,Z     BRANCH IF NOT
494 *
495 *
496          BSI  L  MOVE1      OUTPUT COMMA, MOVE PT
497          MDX          XY14   START LIST CHECK AGAIN
498 *
499 *
500 XY12  S    TLP           IS CHAR LEFT PARENTHESIS
501          BSC  L  XY15,Z     BRANCH IF NOT
502 *
503 *
504          BSI  L  DO         CHECK FOR IMPLIED DO LOOP
505 *
506 *
507          LD   L  SW3       WAS EQUAL SIGN FOUND
508          BSC  L  XY7, -     BRANCH IF NOT
509          MDX          XY16   PROCESS IMPLIED DO LOOPS
510 *
511 *
512 XY15  S    TRGLP        IS CHAR SPECIAL SKIP OPTR
513          BSC  L  XY7,Z     BRANCH IF NOT
514 *
515 *
516          LDX  3 3          NO. WORDS TO BE SKIPPED
517 *
518 *
519          BSI  L  OUT       OUTPUT THREE WORDS
520 *
521 *
522          LD   1 0          LOAD CHARACTER
523          S    TCMA         IS CHARACTER COMMA
524          BSC  L  XY13,Z     BRANCH IF NOT
525 *
526 *
527          MDX  1 1          MOVE POINTER
528          MDX          XY14   CHECK REST OF LIST
529 *
530 *
531 *          CHECKS FOR A NON-DIMENSIONED
532 *          INTEGER VARIABLE
533 *          INDICATES DEFINITION IF DEFINED
534 *          GENERATES THE DO-INITIALIZE OPT
535 XY16  LDX  I1 00X        GET ADDRESS OF
536          MDX  1 -1        VARIABLE
537 *
538 *
539          STX  2 XT1        STORE OUTPUT POINTER

```

```

540      LD      XT1      CHECK TO SEE IF
541      A        TWO      OVERLAP BETWEEN INPUT
542      S      L DOY      AND OUTPUT STRING
543      BSC     L CK1,-    BRANCH IF OVERLAP
544 *
545 *
546      LD      1 0      NON-DIMENSIONED INTEGER VARIABLE
547      BSI     L GETID    GET SYM TBL ID WORD
548      SLA     16      MANIPULATIONS TO
549      S      L SORF     DETERMINE WHICH
550      SLA     1        MASK TO USE
551      LD      SDRDE    MAINLINE OR SUBROUTINE MAS
552      BSC     C        BRANCH NOT FUNCTION
553      LD      SFFDE    FUNCTION MASK
554      AND     3 0      TEST TO SEE IF VALID
555      EOR     S4000    NON-DIMENSIONED INTEGER VA
556      BSC     L XY7,7   BRANCH IF NOT VALID
557 *
558 *
559      LD      3 0      LOAD SYM TBL ID WORD
560      OR      S0020    PUT IN VARIABLE DEFINED BI
561      STO     3 0      PUT BACK IN SYMBOL TABLE
562 *
563 *
564      BSI     L MOVE1   OUTPUT VARIABLE, MOVE PT
565 *
566 *
567      LD      TRGCP    REPLACE VARIABLE IN INPUT
568      STO     1 -1     STRING WITH SKIP OPERATOR
569 *
570 *
571      LD      DOACP    PUT DOA OPERATOR
572      BSI     L PUT     ON OUTPUT STRING
573 *
574 *
575      LD      TLP      REPLACE EQUAL SIGN IN
576      STO     1 0     INPUT STRING
577 *
578 *
579      SLA     16      SET INDEXING PARAMETER
580      STO     ALL     COUNTER TO ZERO
581 *
582 *
583      MDX     1 1     MOVE PCINTER
584 *
585 *
586 XY21 STX     2 XT1   STORE OUTPUT POINTER
587      LD      XT1     CHECK TO SEE IF
588      A        TWO    OVERLAP BETWEEN INPUT
589      S      L DOY    AND OUTPUT STRING
590      BSC     L CK1,- BRANCH IF OVERLAP
591 *
592 *
593      LD      1 0     GET INDEXING PARAMETER
594      BSI     L GETID  GET SYM TBL ID WORD
595      AND     S5BCE   CHECK FOR VALID INTEGER
596      EOR     S4000   CONSTANT OR VARIABLE
597      BSC     L XY7,7 BRANCH IF NOT VALID
598 *
599 *

```

```

600      BSI  L  MOVE1      OUTPUT WORD, MOVE PT
601  *
602  *
603      LD      ALL      IS PARAMETER CTR ZERO
604      BSC  L  XY18, -    BRANCH IF YES
605  *
606  *
607      BSI  L  DELET      REMOVE WORD FROM STRING
608      MDX      XY19      CHECK NEXT WORD ON LIST
609  *
610  *
611  XY18  LD      TRGOP     REPLACE INDEXING PARAMETER
612      STO  1 -1          WITH SKIP OPERATOR
613  *
614  *
615  XY19  MDX  L  ALL,1     INCREMENT IDX PARAM CTR
616  *
617  *
618      LD      1 0        LOAD WORD
619      S      L  TCCP     IS CHARACTER COMMA
620      BSC  L  XY20,7     BRANCH IF NOT
621  *
622  *
623      BSI  L  MOVE1      OUTPUT COMMA, MOVE PT
624  *
625  *          CALL DELET
626      BSI  L  DELET      REMOVE WORD FROM STRING
627      MDX      XY21      CHECK NEXT INDEXING PARAM
628  *
629  *
630  XY20  S      L  HLCSP   IS CHAR RIGHT PARENTHESIS
631      BSC  L  XY7,7     BRANCH IF NOT
632  *
633  *
634      BSI  L  MOVE1      OUTPUT WORD, MOVE PT
635  *
636  *
637      BSI  L  DELET      REMOVE WORD FROM STMT
638  *
639  *
640      LD      L  ALL      IS INDEXING PARAM CTR
641      S      L  ONE      LESS THAN ONE
642      BSC  L  XY7,      BR IF YES
643  *
644      S      L  TWC      IS IT .GT. THREE
645      BSC  L  XY7,-Z     BR IF YES
646  *
647      LDX  I1 00Y      MOVE START LIST PT TO
648      MDX  1 1          NEXT WORD IN LIST
649      BSC  L  XY10      CHECK REST OF LIST
650  *
651  *
652  XY13  LD      1 0        IS CHAR SEMI-COLON
653      BSC  L  XY7,7     BRANCH IF NOT
654  *
655  *          MOVES A WORD TO THE O/F STRING
656  *          ADJUSTS THE STMT NORM
657  XY11  BSI  L  PUT      PUT CHAR ON OUTPUT STRING
658  *
659  *          CORRECT STATEMENT NORM

```

660		LD	I	IDSV2		CLEAR NORM
661		AND	L	SF802		SAVE ID WORD
662		STO		XT4		SAVE OUTPUT POINTER
663		STX	2	XT3		GET NEW NORM BY SUBTRACTIN
664		LD		XT3		ADDRESS OF ID WORD FROM
665		S		IDSV2		PRESENT POINTER
666		SLA		2		
667		A		XT4		PUT NORM IN ID WORD
668		STO	I	IDSV2		PUT BACK ON OUTPUT STRING
669		MDX		XY22		MOVE TO NEXT STMN
670	*					
671	*					CONSTANTS
672	XT3	DC		0		TEMPORARY STORAGE
673	XT4	DC		0		TEMPORARY STORAGE
674	ERID	DC		/A008		ERROR ID WORD
675	*					
676	*					REPLACES THE ERRONEOUS STMN
677	*					BY AN ERROR MESSAGE
678	XY7	LDX	L2	*-*		LOAD OUTPUT POINTER WITH
679	IDSV2	EQU		XY7 1		ADDRESS OF OUTPUT ID WORD
680	*					*MODIFIABLE
681		LD	2	0		LOAD OUTPUT WORD
682		EOR		FIVE		BRANCH IF HAS NO
683		BSC	L	* 5,E		STATEMENT NUMBER
684		LD		FIVE		PUT STMN NO. BIT AND ADD
685		A		ERID		ONE TO NORM OF ID WORD
686		STO	2	0		PUT ON OUTPUT STRING
687		MDX	2	1		MOVE POINTER
688		MDX		* 3		GO PUT ERROR NO. ON STRING
689		LD	L	ERID		LOAD ERROR ID WORD
690		STO	2	0		PUT ON OUTPUT STRING
691		LD	L	ERNO		LOAD ERROR NUMBER
692		STO	2	1		PUT ON STRING
693		MDX	2	2		MOVE POINTER
694	*					
695	*					MOVE TO NEXT STATEMENT
696	XY22	LDX	L1	*-*		LOAD INPUT POINTER WITH
697	NXTID	EQU		XY22 1		NEXT STMN ADDRESS
698	*					*MODIFIABLE
699		BSC	L	XY1		SCAN STMN
700	*					
701	*					
702	*					CHECKS THE SYNTAX OF GO TO
703	*					STMNS - CHECKS THE STMN NO.
704	*					LIST FOR VALID REFERENCES
705	XY24	S		IGOTC		IS IT GO TO STMN
706		BSC	L	W11,7		BRANCH IF NOT
707	*					
708	*					
709		LDX	3	48		PUT ERROR NO. 48
710		STX	L3	ERNO		IN ERNO
711	*					
712	*					
713		LD		GOTCF		LOAD GO TO OPERATOR
714		BSI		CHECK		PUT ON OUTPUT STRING
715	*					
716	*					
717		LD	1	0		LOAD WORD
718		S		ULP		IS CHAR LEFT PARENTHESIS
719		BSC	L	XY26, -		BRANCH IF YES

720	*				
721	*				
722		BSI	L	LIST	CHECK FOR CORRECT FORMAT
723	*				
724	*				
725		LD	L	CNT	IS NO. OF PLACES TO BRANCH
726		S		THER	GREATER THAN ONE
727		BSC	L	XY7,7	BRANCH IF YES
728	XYZ13	MDX		XY13	CHECK FOR END OF STMT
729	*				
730	*				
731	XY26	BSI		MOVE1	OUTPUT WORD, MOVE POINTER
732	*				
733	*				
734		BSI	L	LIST	CHECK FOR CORRECT FORMAT
735	*				
736	*				
737		LD	1	0	IS CHARACTER
738		S		URP	RIGHT PARENTHESIS
739		BSC	L	XY7,7	BRANCH IF NOT
740	*				
741	*				
742		LD	L	CNT	LOAD COUNT OF STMT NOS.
743		EOR		FIVE	IS COUNT ODD
744		BSC	L	XY7,8	BRANCH IF YES
745	*				
746	*				
747		LDX	3	49	PUT ERROR NO. 49
748		STX	L3	ERN0	IN ERNO
749	*				
750	*				
751		BSI		MOVE1	OUTPUT WORD, MOVE POINTER
752	*				
753	*				
754		LD	1	0	IS CHARACTER
755		S		UCMA	COMMA
756		BSC	L	XY7,7	BRANCH IF NOT
757	*				
758	*				
759		BSI		MOVE1	OUTPUT WORD, MOVE POINTER
760	*				
761	*				
762		LD	1	0	LOAD WORD
763		BSI		GETID	GET SYM TBL ID WORD
764		AND		T0BDE	IS IT NON-DIMENSIONED
765		EOR		T4000	INTEGER VARIABLE
766		BSC	L	XY7,7	BRANCH IF NOT
767	*				
768	*				
769		BSI		MOVE1	OUTPUT WORD, GET NEXT WORD
770		MDX		XYZ13	CHECK FOR END OF STMT
771	*				
772	*				CONSTANTS
773	T4000	DC		/4000	INTEGER MASK - PART 2
774	T0BDE	DC		/0BDE	INTEGER MASK - PART 1
775	FIVE	DC		5	STMT NO. MASK
776	TGOTO	DC		/0E- /10	GO TO ID TYPE
777	ULP	DC		/10	LEFT PARENTHESIS
778	URP	DC		/02	RIGHT PARENTHESIS
779	UCMA	DC		/12	COMMA

```

780 GCTOP DC      /16      GO TO STMT ID TYPE
781 THER DC      1      CNE
782 *****
783 *              GETS THE ID WORD FROM THE
784 *              SYMBOL TABLE ENTRY.
785 *****
786 *
787 *
788 GETID DC      0      ENTRY POINT
789     BSC L XY7,-     BRANCH IF SPECIAL OPERATOR
790     AND S01FF      GET NO. OF TEL ENTRY
791     SLA 2          GET ADDRESS RELATIVE TO
792     STC GETIE      START OF SYM TBL BY
793     SRA 2          MULTIPLYING NO. OF ENTRY P
794     S GETIE       THREE SIZE OF ENTRY
795     A L S0FST     GET ABSOLUTE ADDRESS - 3
796     STO * 1       OF SYM TBL ENTRY
797     LDX L3 *-*    INITIALIZE SYM TBL PT
798 GETIE EQU     *-1   *MODIFIABLE
799     MOX 3 3       GET SYM TBL ENTRY ADDR
800     LD 3 0        LOAD SYM TBL ID WORD
801     BSC I GETID   RETURN
802 *
803 *              CONSTANTS
804 SC1FF DC      /07FF     MASK TO GET SYM TBL NO.
805 *****
806 *              MOVES THE WORD IN THE ACC TO
807 *              THE OUTPUT STRING
808 *              MOVES THE POINTER
809 *****
810 *
811 *
812 PUT DC      0      ENTRY POINT
813     STO 2 0       PUT WORD ON OUTPUT STRING
814 *
815 *
816     MOX 2 1       MOVE POINTER
817 *
818 *
819     BSC I PUT     RETURN
820 *****
821 *              CHECKS FOR AN OVERLAP CONDITION
822 *              BETWEEN THE INPUT AND OUTPUT
823 *              STMT STRINGS
824 *****
825 *
826 *
827 CHECK DC     0      ENTRY POINT
828     BSI PUT      OUTPUT WORD
829 *
830 *
831     STX 1 CKTEM   STORE INPUT PT
832     LD CKTEM     LOAD INPUT PT
833     STX 2 CKTEM   STORE OUTPUT PT
834     S CKTEM      SUBTRACT OUTPUT PT
835     BSC L CK1,    BRANCH IF OVERLAP
836 *
837 *
838     BSC I CHECK   RETURN
839 *

```

```

840 *                               SET UP OVERLAP ERROR
841 CK1  MDX  L  ERROR,1  SET OVERLAP ERROR
842      BSC  L  OVERF    GO TO NEXT PHASE
843 *
844 *                               CONSTANTS
845 CKTEM DC          0          TEMPORARY STORAGE
846 *****
847 *                               MOVES ONE WORD FROM THE INPUT
848 *                               STRING TO THE OUTPUT STRING
849 *                               INCREMENTS THE POINTER
850 *****
851 *
852 *
853 MOVE1 DC          0          ENTRY POINT
854      LD   1  0          LOAD WORD
855      BSI          PUT       PUT ON OUTPUT STRING
856 *
857 *
858      MDX  1  1          MOVE INPUT POINTER
859 *
860 *
861      BSC  I  MOVE1      RETURN
862 *****
863 *                               MOVES WORDS TO THE OUTPUT
864 *                               STRING USING XE3 AS A COUNT
865 *                               CONTROL
866 *****
867 *
868 *
869 OUT   DC          0          ENTRY POINT
870      BSI          MOVE1     OUTPUT WORD, MOVE POINTER
871 *
872 *
873      MDX  3 -1          DECREMENT COUNTER
874 *
875 *
876      MDX          OUT  1    CONTINUE LOOP
877 *
878 *
879      BSC  I  OUT        RETURN
880 *****
881 *                               CHECKS NAMES FOR DIMENSIONING
882 *                               DETERMINES THE DIMENSION LEVEL
883 *                               MOVES THE NAME AND DIMENSION
884 *                               INFORMATION TO THE O/P STRING
885 *                               ALLOWS SPACE IN THE O/P STRING
886 *                               ACCORDING TO THE DIM LEVEL.
887 *****
888 *
889 *
890 NAME  DC          0          ENTRY POINT
891      LD   1  0          LOAD WORD
892      BSI          PUT       PUT
893 *
894 *
895      BSI          GETID     GET SYM TBL ID WORD
896      AND          S1800     IS IT DIMENSIONED NAME
897      BSC  L  NAME1,7      BRANCH IF YES
898 *
899 *

```

```

900          MDX      1 1          MOVE PCINTER
901 *
902 *
903          STX      0 SW7        TAG DIMENSION NAME SWITCH
904 *
905 *
906 NAME3 SLA        16          CLEAR DIMENSION NAME
907          STO        SW1        SUBSCRIPTED SWITCH
908 *
909 *
910 NAME8 BSC      I  NAME        RETURN
911 *
912 *
913 NAME1 SLA        16          CLEAR DIMENSIONED
914          STO        SW7        NAME SWITCH
915 *
916 *
917          MDX      1 1          MOVE INPUT POINTER
918 *
919 *
920          LD        1 0
921          S          NP3        IS IT 3-DIMENSIONED VAR
922          BSC      L  NAME2,7    BRANCH IF NOT
923 *
924 *
925          LDX      3 15        SET MOVE-WORD COUNTER
926 *
927 *
928 NAME4 BSI        CUT          PUT WORDS ON OUTPUT STRING
929          MDX        NAME3      NORMALIZE SWITCH 1
930 *
931 *
932 NAME2 S          NP2        IS IT 2-DIMENSIONED VAR
933          BSC      L  NAME5,2    BRANCH IF NOT
934 *
935 *
936          LDX      3 11        LOAD MOVE-WORD CTR
937          MDX        NAME4      GO TO PUT ON STRING
938 *
939 *
940 NAME5 S          NP1        IS IT 1-DIMENSIONED VAR
941          BSC      L  NAME6,7    BRANCH IF NOT
942 *
943 *
944          LDX      3 7         LOAD MOVE-WORD CTR
945          MDX        NAME4      GO TO PUT ON STRING
946 *
947 *
948 NAME6 S          NP0        IS IT LITERAL
949          BSC      L  NAME7,7    BRANCH IF NOT
950 *
951 *
952          LDX      3 3         LOAD MOVE-WORD CTR
953          MDX        NAME4      GO TO PUT ON STRING
954 *
955 *
956 NAME7 STX        0 SW1        SET SWITCH TO NOT SUBSCRIP
957          MDX        NAME8      RETURN
958 *
959 *          CONSTANTS

```



```

960 SW1 DC 0 SWITCH 1
961 SW7 DC 0 SWITCH 7
962 S1800 DC /1800 MASK TO TEST FOR DIMENSION
963 NP3 DC /1E 3 OPERATOR
964 NP2 DC /1C-/1E 2 OPERATOR
965 NP1 DC /1A-/1C 1 OPERATOR
966 NP0 DC /18-/1A LITERAL OPERATOR
967 *****
968 * CHECKS FOR IMPLIED DO LOOPS IN
969 * INPUT AND OUTPUT LISTS
970 * CHECKS THE SYNTAX OF THE
971 * IMPLIED DO
972 *****
973 *
974 *
975 DO DC 0 ENTRY POINT
976 STX 1 DOY SAVE INPUT POINTER
977 *
978 *
979 LDX 3 1 INITIALIZE
980 STX 3 DOPAR PARENTHESIS COUNTER
981 *
982 *
983 DO3 MDX 1 1 MOVE POINTER
984 *
985 *
986 LD 1 0 IS CHARACTER SEMI-COLON
987 BSC L DO1, - BRANCH IF YES
988 *
989 *
990 S DLPAR IS CHAR LEFT-PARENTHESIS
991 BSC L DO2, Z BRANCH IF NOT
992 *
993 *
994 MDX L DOPAR, 1 INCREMENT PARENTHESIS CTR
995 MDX DO3 GET NEXT WORD
996 *
997 *
998 DO2 S DPMES IS CHARACTER EQUAL SIGN
999 BSC L DO4, - BRANCH IF YES
1000 *
1001 *
1002 S DEMPE IS CHAR RIGHT PARENTHESIS
1003 BSC L DO6, Z BRANCH IF NOT
1004 *
1005 *
1006 MDX L DOPAR, -1 DECREMENT PARENTHESIS CTR
1007 *
1008 *
1009 MDX DO3 GET NEXT WORD
1010 *
1011 *
1012 DO1 SLA 16 CLEAR EQUAL
1013 STO SW3 SIGN SWITCH
1014 *
1015 *
1016 DO5 LDX L1 *-* RESTORE
1017 DOY EQU DO5 1 INPUT COUNTER. MODIFIABLE
1018 *
1019 *

```

```

1020          BSC  I  00          RETURN
1021  *
1022  *
1023  004  LDX  L3  *-*          LOAD
1024  00PAR EQU          004  1    PARENTHESIS COUNTER
1025  *          *MODIFIABLE
1026          MDX   3  -1          DECREMENT PARENTHESIS CTR
1027          MDX          003      GET NEXT WORD
1028  *
1029  *
1030          STX   1  00X          SAVE INPUT POINTER
1031  *
1032  *
1033          STX   0  SW3          SET EQUAL SIGN SWITCH
1034          MDX          005      RETURN
1035  *
1036  *          CONSTANTS
1037  SW3  DC          0          SWITCH 3
1038  0LPAR DC          /10        LEFT PARENTHESIS
1039  0PMES DC          /0E- /10    EQUAL SIGN
1040  0EMRP DC          /02- /0E    RIGHT PARENTHESIS
1041  0OX  DC          0          INPUT POINTER AT EQUAL SIG
1042  0SPMR DC          /18- /02     3-DIMENSIONED OPERATOR
1043  0SPDL DC          /1E- /18    0-DIMENSIONED OPERATOR
1044  0CS1 DC          /07         SEVEN
1045  *
1046  *
1047  006  S          DSPMR        IS CHARACTER A SPECIAL
1048          BSC  L  003, 7      LEFT PARENTHESIS
1049          S          DSPDL        GET NEXT CHARACTER
1050          BSC  L  003, -7      IF NOT
1051  *
1052  *          MOVE PT OVER SUBSCRIPT
1053          A          00S1        ADD NINE
1054          SLA          1          MULTIPLY BY TWO
1055          STO          * 1        INCREMENT INPUT PT
1056          MDX  L1  *-*          TO MOVE OVER SUBSCRIPT
1057  *          *MODIFIABLE
1058          MDX          003      GET NEXT WORD
1059  *****
1060  *          CHECKS THE STMT NUMBER LIST
1061  *          IN GO TO STMT FOR VALID
1062  *          REFERENCES.
1063  *          LABEL LIST
1064  *****
1065  *
1066  *
1067  LIST  DC          0          ENTRY POINT
1068          SLA          16        CLEAR COUNT
1069          STO          CNT        OF STMT NOS.
1070  *
1071  *          NON-FORMAT STATEMENT NUMBER
1072  LIST3 LD          1  0          LOAD WORD
1073          BSI  L  GETID          SET SYM TBL 10 WORD
1074          AND          TFFDE        IS WORD A
1075          EOR          S0200        STMT NO.
1076          BSC  L  LIST2, Z        BRANCH IF NOT
1077  *
1078  *
1079          BSI          MOVE1        OUTPUT WORD, MOVE POINTER

```

```

1080 *
1081 *
1082 LIST1 MDX L CNT,1 MODIFY COUNTER
1083 *
1084 *
1085 LD 1 0 LOAD WORD
1086 S LCMA IS CHARACTER COMMA
1087 BSC L LIST4, - BRANCH IF YES
1088 MDX LIST5 RETURN
1089 *
1090 *
1091 LIST2 LDX 3 43 PUT ERROR NO. 43
1092 STX L3 ERNO IN ERNO
1093 *
1094 *
1095 LIST5 BSC I LIST RETURN
1096 *
1097 *
1098 LIST4 MDX L CNT,1 INCREMENT COUNTER
1099 *
1100 *
1101 BSI L MOVE1 OUTPUT WORD, MOVE POINTER
1102 MDX LIST3 GET NEXT WORD
1103 *
1104 * CONSTANTS
1105 CNT DC 0 STMT NO. COUNTER
1106 LCMA DC /12 COMMA
1107 TFFDE DC /FFDE STMT NO. MASK - PART 1
1108 S0200 DC /0200 STMT NO. MASK - PART 2
1109 *****
1110 * DELETES ONE WD FROM I/P STRING
1111 *****
1112 *
1113 *
1114 DELET DC 0 ENTRY POINT
1115 MDX 1 -1 ADDRESS OF PREVIOUS WORD
1116 *
1117 *
1118 STX 1 DELX SAVE ADDRESS
1119 *
1120 *
1121 DLT2 LD 1 1 MOVE WORD DOWN ONE
1122 STO 1 0 POSITION IN STRING
1123 *
1124 *
1125 BSC L DLT1, - EXIT IF SEMI-COLON
1126 *
1127 *
1128 MDX 1 1 MOVE POINTER
1129 MDX DLT2 CONTINUE LOOP
1130 *
1131 *
1132 DLT1 LDX L1 *-* RESTORE
1133 DELX EQU DLT1 1 INPUT POINTER. MODIFIABLE
1134 *
1135 *
1136 BSC I DELET RETURN
1137 *
1138 * CHECKS FOR VALID VARIABLES IN
1139 * READ, WRITE, AND GO TO STMTS

```

1140	TESTV	DC		0	ENTRY POINT
1141		LD	L	SORF	LOAD SCRF
1142		BSC	L	TESTW,	BRANCH NOT FUNCTION
1143		LD		1 0	LOAD NAME
1144		S	L	FNAME	IS IT PROGRAM NAME
1145		BSC	I	TESTV, -	BRANCH IF YES
1146		LD	L	SW2	IS IT WRITE STMNT
1147		BSC	L	TESTW, -	BRANCH IF YES
1148		LD		SA7DE	LOAD READ MASK
1149		MDX		TESTX	GO TO TEST
1150	TESTW	LD		S83DE	TEST FOR
1151	TESTX	AND		3 0	VALID VARIABLE
1152		BSC	L	XY7,7	BRANCH IF INVALID
1153		BSC	I	TESTV	RETURN
1154	S83DE	DC		/83DE	WRITE MASK
1155	SA7DE	DC		/A7DE	READ MASK
1156	*				
1157	*				MOVES A STMNT TO THE O/P STRING
1158	XY27	LDX	L3	*-*	LOAD NORM. MODIFIABLE
1159		MDX		3 -1	DECREMENT NORM
1160		MDX		* 1	SKIP NEXT INSTRUCTION
1161		MDX		XY27A	MOVE TO NEXT STMNT
1162		BSI	L	MOVE1	OUTPUT WORD, MOVE POINTER
1163		MDX		3 -1	DECREMENT NORM
1164		MDX		XY27 3	CONTINUE LOOP
1165	XY27A	BSC	L	XY2?	MOVE TO NEXT STMNT
1166	*				
1167	*				
1168	W11	S		WEQU	IS IT EQUIVALENCE STMNT
1169		BSC	L	XY27,7	BRANCH IF NOT
1170	*				
1171	*				
1172		LDX		3 68	PUT ERROR NO. 68
1173		STX	L3	ERNO	IN ERNO
1174	*				
1175	*				
1176	W12	LD		1 0	LOAD WORD
1177		S		WLP	IS CHAR LEFT PARENTHESIS
1178		BSC	L	XY7,7	BRANCH IF NOT
1179	*				
1180	*				INITIALIZE LIST COUNT
1181		STO		LOT	TO ZERO
1182	*				
1183	*				
1184	W13	BSI	L	MOVE1	MOVE POINTER
1185	*				
1186	*				
1187		LD		1 0	LOAD WORD
1188		BSI	L	GETID	GET SYM TBL ID WORD
1189		AND		W83DE	IS VALID TO BE EQUIVALENCE
1190		BSC	L	XY7,7	BRANCH IF NOT
1191	*				
1192	*				
1193		BSI	L	NAME	CHECK NAME AND SUBSCRIPTS
1194		MDX	L	LOT,1	INCREMENT LIST COUNT
1195	*				
1196	*				
1197		LD	L	SW1	IS DIMENSION NAME SUBSCRIP
1198		BSC	L	XY7,2	BRANCH IF NOT
1199	*				

```

1200 *
1201 LD L SW7 IS IT DIMENSIONED NAME
1202 BSC L W14,7 BRANCH IF NO
1203 *
1204 *
1205 LD 2 -3 LOAD WORD
1206 S W18 IS IT LITERAL OPERATOR
1207 BSC L XY7,7 BRANCH IF NOT
1208 *
1209 *
1210 W14 LD 1 0 LOAD WORD
1211 S WDMA IS CHARACTER COMMA
1212 BSC L W13, - BRANCH IF YES
1213 *
1214 *
1215 S WRP IS CHAR RIGHT PARENTHESIS
1216 BSC L XY7,7 BRANCH IF NO
1217 *
1218 *
1219 LD LCT LOAD LIST COUNT
1220 S CNE LESS THAN TWO ARRAYS
1221 BSC L XY7, BRANCH IF YES
1222 *
1223 *
1224 BSI L MOVE1 OUTPUT WORD, MOVE POINTER
1225 *
1226 *
1227 LD 1 0 LOAD WORD
1228 S WDMA IS IT COMMA
1229 BSC L XY13,7 BRANCH IF NO
1230 *
1231 *
1232 BSI L MOVE1 OUTPUT WORD, MOVE PT
1233 MDX W12 CHECK CHARACTER
1234 *
1235 *
1236 LCT DC 0 LIST COUNT
1237 ONE DC 1 CNE
1238 WEQU DC /15-/0E EQUAL SIGN
1239 WLP DC /10 LEFT PARENTHESIS
1240 W830E DC /87DE EQUIVALENCE MASK
1241 W18 DC /18 LITERAL OPERATOR
1242 WDMA DC /12 COMMA
1243 WRP DC /02-/12 RIGHT PARENTHESIS
1244 *
1245 *
1246 BSS OVERL-**320*3 PHASE-17 PATCH AREA
1247 END START

```