

```

1 *   GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2 *****
3 *
4 *   PROGRAM NAME   FPH-14
5 *
6 *   MODEL NUMBER   BF014
7 *
8 *   PURPOSE        FORTRAN PHASE-14
9 *
10 *  PROGRAMMER     DICK WALLMANN
11 *
12 *****      REVISION LIST      *****
13 *
14 *   RV DATE       SOC   BY   REASON FOR CHANGE
15 *   -----
16 *
17 *   01 11/16/70  NONE   RPH  INITIAL RELEASE
18 *
19 *****
20 *****
21         HDNG      MPX FORTRAN ** DO CONTINUE
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 *   * CHECKS FOR VALID SYNTAX IN DO STATEMENTS
27 *     AND IN NESTED DO-LOOPS
28 *   * GENERATES THE CODING NEEDED TO PERFORM THE
29 *     DO-TEST.
30 *   * CHECKS THE SYNTAX OF DO, CONTINUE, BACK-
31 *     SPACE, REWIND, END FILE, STOP, PAUSE, AND
32 *     END STATEMENTS
33 *   * CHECKS FOR A GO TO, IF, STOP, CALL LINK,
34 *     CALL EXIT, OR RETURN STATEMENT AS THE LAST
35 *     EXECUTABLE STATEMENT OF THE SOURCE PROGRAM.
36 *
37 *ENTRY POINTS-
38 *   * START-PHASE 14 IS READ INTO COPE BY PHASE
39 *     13 VIA ROLRX.  EXECUTION IS BEGUN AT
40 *     LOCATION START.
41 *
42 *INPUT-
43 *   * THE STATEMENT STRING
44 *   * THE SYMBOL TABLE
45 *   * THE FORTRAN COMMUNICATION AREA
46 *
47 *OUTPUT-
48 *   * THE STATEMENT STRING
49 *   * THE SYMBOL TABLE
50 *   * THE FORTRAN COMMUNICATION AREA
51 *
52 *EXTERNAL REFERENCES-
53 *   * SUBROUTINES-
54 *     ROLRX
55 *   * OTHER FORTRAN PHASES
56 *     NONE
57 *
58 *EXITS-
59 *   * NORMAL-

```

```

60 *          PHASE 15 IS CALLED VIA ROLFX AND
61 *          CONTROL IS PASSED TO IT.
62 * * ERRORS-
63 *   OVERLAP-
64 *     PROCESSING IS HALTED, THE ERROR WORD IS
65 *     SET IN FCOM, PHASE 15 IS CALLED VIA
66 *     ROLFX, CONTROL IS PASSED TO IT.
67 *   SYNTAX-
68 *     THE ERRONEOUS STATEMENT IS REPLACED
69 *     WITH AN ERROR NUMBER, PROCESSING
70 *     CONTINUES, PHASE 15 IS CALLED VIA ROLFX,
71 *     AND CONTROL PASSED TO IT.
72 *     THE ERRORS DETECTED ARE NUMBERS 51,52,
73 *     53,54,55,56,57,58,59,60 AND 61.
74 *
75 *TABLES/WORK AREAS-
76 * * THE STATEMENT STRING
77 * * THE SYMBOL TABLE
78 * * THE FORTRAN COMMUNICATION AREA
79 *
80 *ATTRIBUTES-N/A
81 *
82 *NOTES-
83 *   THE SWITCHES USED IN PHASE 14 FOLLOW. IF NON-
84 *   ZERO, THE SWITCH IS TRANSFER=T. IF ZERO, THE
85 *   SWITCH IS NORMAL=N.
86 *     NORM2-PAUSE OR STOP STATEMENT
87 *           T PAUSE OR STOP FOUND
88 ******
89 *   HDNG      MPX FORTRAN ** DO CONTINUE
90 *   ABS      REF CORE
91 *
92 *   SYSTEM AND FORTRAN EQUATES
93 *
94 *   MEMRY EQU      FFFF CORE      MAXIMUM CORE SIZE
95 *   PHSIZ EQU      4*320                MAXIMUM PHASE SIZE
96 *   OVERL EQU      MEMRY-PHSI7          PHASES 2-29 START
97 *   FCOM EQU       OVERL-22             FORTRAN COMM. TABLE
98 *   PHNTB EQU      FCOM-56             PHASE TABLE
99 *   ROLRX EQU      PHNTR-50            INTERPHASE CALL
100 *
101 *   FORTRAN COMMUNICATION AREA
102 *
103 *   ORG          FCOM          FORTRAN COMM AREA
104 *   SCFS BSS     1            START OF STRING
105 *   EOFS BSS     1            END OF STRING
106 *   SCFST BSS    1            START OF SYMBOL TABLE
107 *   SOFNS BSS    1            START OF NON-STMNT NOS.
108 *   SOFXT BSS    1            START OF SUBSC TEMPS
109 *   SCFGT BSS    1            START OF GENERATED TEMPS
110 *   ECFST BSS    1            END OF SYMBOL TABLE
111 *   COMON BSS    1            NEXT AVAILABLE COMMON
112 *   CSIZE BSS    1            SIZE OF COMMON
113 *   ERROR BSS    1            OVERLAP ERROR
114 *   FNAME BSS    1            PROGRAM NAME WD 1
115 *           BSS    1            PROGRAM NAME WD 2
116 *   SCRF BSS     1            SUBR (-) OR FUNC (+)
117 *   CCWD BSS     1            CONTROL CARD WD
118 *           PIT 15 TRANSFER TRACE
119 *           PIT 14 ARITHMETIC TRACE

```

```

120 *          BIT 13 EXTENDED PRECISION
121 *          BIT 12 LIST SYMBOL TABLE
122 *          BIT 11 LIST SUBPROGRAM NAMES
123 *          BIT 10 LIST SOURCE PROGRAM
124 *          BIT 9 ONE WORD INTEGERS
125 IOCS BSS 1          IOCS CONTROL CARD WORD
126 *
127 *          SEE PHASE ONE FOR BIT PATTERNS
128 *
129 DFCNT BSS 1          DEFINED FILE COUNT
130 *
131 LCOMN BSS 2          SIZE OF INSKEL COMMON
132 *
133 ICCER BSS 2          IOCS CONTROL CARD ERROR
134 *
135          BSS 2          SYSTEM LOADER USE
136 *
137 *          END OF FORTRAN COMMUNICATION
138 *          AREA
139 *
140 *
141          HDNG MPX FORTRAN ** DO CONTINUE
142          ORG OVERL PHASE ENTRY
143 START LD ERROR TEST FOR OVERLAP ERROR
144          BSC L OUT,Z BRANCH IF ERROR
145 *
146 *          INITIALIZE PHASE
147          LDX I1 SOFS INITIALIZE INPUT POINTER
148          LDX L3 TABLE-4 INITIALIZE DO TABLE PT
149          SLA 16 CLEAR FIRST WORD
150          STO 3 4 OF DO TABLE
151 *
152 *
153 *          CHECKS FOR END, DO, STOP, AND
154 *          PAUSE STATEMENTS
155 ABEL LD 1 0 LOAD STMT ID WORD
156          STX L1 PAVE SAVE ADDRESS OF ID WORD
157          SRA 2
158          AND IDTPE GET STMT ID TYPE
159          S ENDC IS IT END STMT
160          BSC L DOTPL, - BRANCH IF YES
161 *
162 *
163          S DOC IS IT DO STMT
164          BSC L FULL, - BRANCH IF YES
165 *
166 *
167          S BACKC IS IT BACKSPACE
168          BSC L CALL1, - BRANCH IF YES
169 *
170 *
171          S WINDC IS IT REWIND
172          BSC L CALL2, - BRANCH IF YES
173 *
174 *
175          S FILEC IS IT END FILE
176          BSC L CALL3, - BRANCH IF YES
177 *
178 *
179          S STCFC IS IT STOP

```

```

180          BSC L CALL4, -   BRANCH IF YES
181 *
182 *
183          S          PAUSE   IS IT PAUSE
184          BSC L CALL5, -   BRANCH IF YES
185 *
186 *
187          S          CONT0   IS IT CONTINUE
188          BSC L CNTST, -   BRANCH IF YES
189 *
190 *
191 *
192          BAKER LD      1 0   CHECKS FOR NUMBERED STMENTS
193          BSC L TINUE,E     DOES STMT HAVE STMT NO.
194 *
195 *
196          BSI L MOVE      MOVE TO NEXT STATEMENT
197          MDX ABEL        START STMT CHECK
198 *
199 *
200          IDTPE DC      /3E00  CONSTANTS
201          IDNRM DC      /01FF  STMT ID TYPE MASK
202          ENDC DC      /0400  NORM MASK
203          DCC DC      /1200  END
204          BACKC DC     /EC00  DO
205          WINDC DC     /0800  BACK SPACE
206          FILEC DC     /FC00  REWIND
207          STOPC DC     /2800  END FILE
208          PAUSE DC     /F800  STOP
209          CONT0 DC     /2000-2600 PAUSE
210          TWO DC      2      CONTINUE
211          ONE DC      1      TWO
212          TABLE BSS   126   ONE
213          ERR51 DC     51     DO TABLE
214          ERRID DC     /A008  ERROR NO. 51
215          ERSID DC     /A000  ERROR STMT ID WORD
216          CNTOK DC     /8009  ERROR STMT ID WITH NO.
217          ERQ53 DC     53     CONTINUE STMT ID WORD
218 *
219 *
220          CNTST LD      1 0   LOAD STMT ID WORD
221          S          CNTOK   IS ID WORD OK
222          BSC L BAKER, -   BRANCH IF YES
223 *
224 *
225          LD          ERQ53   SET UP ERROR NO. 53
226          MDX        CLOS1 1  PUT ERROR STMT ON STRING
227 *
228 *
229 *
230 *
231 *
232          DOTBL LD      3 4   CHECKS TO INSURE THAT THE DO
233          BSC L OUT1, -   TABLE IS EMPTY AT THE END
234 *
235 *
236          STO      1 1   OF PHASE
237          LD      1 0   IS DO TABLE EMPTY
238          STO      1 3   BRANCH IF YES
239          LD      ER51  STORE DO RANGE STMT NO.

```

```

240          STO      1 2          PUT ON STRING
241          LD       ERSID        LOAD ERROR STMT ID WORD
242          STO      1 0          PUT ON STRING
243          MDX      1 3          MOVE INPUT PT
244          MDX      3 -5         MOVE DO TABLE PT
245          STX     L1 EOFES       NEW END OF STRING ADDRESS
246          *
247          *
248          LD       L EOFEST      CHECK FOR OVERLAP
249          A        L TWO         BETWEEN STRING AND
250          S        L EOFES       SYMBOL TABLE
251          BSC     L DOTPL,-      BRANCH IF NO OVERLAP
252          *
253          *
254          JIMY    LD       L ONE   SET UP
255          STO     L ERROR        OVERLAP ERROR
256          *
257          *          TRANSFERS TO THE FOL ROUTINE
258          *          TO LOAD THE NEXT PHASE
259          OUT     BSI    L ROLPX   CALL DOWN PHASE 15
260          DC      15           NEXT PHASE NUMBER
261          *
262          *          CONSTANTS
263          NRMSV   DC      0        TEMPORARY STORGE FOR NORM
264          ERRNO   DC      0        ERROR NUMBER
265          CONTC   DC     /2000     CONTINUE ID TYPE
266          DIP     DC      0        TEMPORARY STORAGE
267          OCTID   DC     /C027     DO STMT ID WORD - NORM 9
268          ERR52   DC      52       ERROR NO. 52
269          TBCN    DC     TABLE 121 END OF TABLE
270          *
271          *
272          *          CHECKS FOR MORE THAN 25 ENTRIES
273          *          IN THE DO TABLE
274          FULL    STX     3 DIP     STORE DO TABLE POINTER
275          LD      DIP     IS DO TABLE
276          S       TBCN    FULL
277          BSC     L CEE,7   BRANCH NOT FULL
278          *
279          *
280          CLOS1   LD      ERR52    SET UP
281          STO     ERRNC    ERROR NO. 52
282          STX     3 SAVE3 1   SAVE DO TABLE PTR
283          *
284          *          REPLACES THE ERRONEOUS STMT
285          *          WITH AN ERROR MESSAGE
286          *          CLOSES UP THE STRING
287          CLOSE   LDX     I1 PAVE   LOAD STMT ID WORD ADDR
288          LD      1 0        LOAD STMT ID WORD
289          SRA     2          GET STMT
290          AND     L IDNRM     NORM
291          STO     NRMSV      SAVE NORM
292          A       L PAVE      GET ADDRESS OF NEXT STMT
293          STO     GET 1      ID WORD
294          GET     LDX     L2 0      INITIALIZE LOOP INPUT PT
295          *
296          LD      1 0        LOAD STMT ID WORD
297          BSC     L CALD,E     BRANCH IF HAVE STMT NO.
298          MDX     CALD 4      GO GET ERROR ID
299          CALD   LD      ERIID    LOAD ERROR ID WORD

```

300		STO	1 0	PUT ON STRING
301		MDX	1 1	MOVE PT OVER STMT NO.
302		MDX	CALD7	PUT ERROR NO. ON STRING
303		LD	L ERRID	LOAD ERROR ID WORD
304		STO	1 0	PUT ON STRING
305	CALD7	LD	ERRNC	LOAD ERROR NO.
306		STO	1 1	PUT ON STRING
307		MDX	1 1	MOVE POINTER
308		LD	L EOFS	FIND RANGE
309		S	L PAVE	OF STRING
310		S	NRMSV	TO BE
311		STO	RANGE 1	MOVED
312	RANGE	LDX	L3 0	INITIALIZE RANGE CTR
313	*			*MODIFIABLE
314		MDX	3 1	
315	LOOP1	LD	2 0	MOVE WORD DOWN
316		STO	1 1	IN STRING
317		MDX	1 1	MOVE POINTERS
318		MDX	2 1	
319		MDX	3 -1	DECREMENT RANGE COUNTER
320		MDX	LOOP1	CONTINUE LOOP
321		STX	L1 EOFS	NEW END OF STRING ADDRESS
322		LDX	I1 PAVE	RESTORE INPUT PT
323	SAVE3	LDX	L3 0	RESTORE DO TPL PT
324	*			*MODIFIABLE
325		BSC	L BAKER	CHECK FOR STMT NO.
326	*			
327	*			CONSTANTS
328	WHY	DC	0	TEMPORARY STORAGE FOR PT
329	LOADI	DC	/0800	LOAD INSTRUCTION
330	ADDI	DC	/1800	ADD INSTRUCTION
331	STOI	DC	/1000	STORE INSTRUCTION
332	SUBTI	DC	/2000	SUBTRACT INSTRUCTION
333	BSCI	DC	/5008	BSC I INSTRUCTION
334	DOXID	DC	/3000	DO STMT ID
335	ERIID	DC	/A000	ERROR STMT ID WORD
336	FIDWD	DC	/00-706	FORMAT
337	NO51E	DC	51	ERROR NO 51
338	*			
339	*			
340	*			CHECKS FOR AN EQUIVALENCE STMT
341	*			NO. IN THE DO TABLE
342	*			CHECKS FOR A TRANSFER STMT
343	TINUE	LD	1 1	LOAD WORD
344		S	3 4	IS IT DO RANGE STMT NO.
345		BSC	L BAKER 3,7	BRANCH IF NOT
346	*			
347	*			
348		BSI	L TRCKS	CHECK IF TRANSFER STMT
349		DC	* 1	BRANCH IF TRANSFER
350		DC	PFEM	BRANCH IF NO TRANSFER
351	*			
352	*			
353	ER51	LD	NO51E	SET UP
354		MDX	CLOS1 1	ERROR NO. 51
355	*			
356	*			
357	PFEM	S	FIDWD	IS IT FORMAT STMT
358	*			CHECKS THE STMT ID WORD FOR TH
359	*			REFERENCED INDICATOR

```

350 *                CHECKS FOR A CONTINUE STMT
361 BSC L ER51, -    BRANCH IF YES
362 *
363 *
364 LD 1 0           PUT REFERENCED STMT PT
365 OR L TWO        IN STMT ID
366 STO 1 0        PUT BACK ON STRING
367 *
368 *
369 STX 1 WHY      SAVE INPUT PT
370 *
371 *
372 LD 1 0         LOAD STMT ID WORD
373 SRA 2         GET STMT
374 AND L IDTPE   ID TYPE
375 S          CONTC IS IT CONTINUE
376 BSC L MOVE1, - BRANCH IF YES
377 *
378 *
379 LIZ1 BSI L MOVE MOVE PT TO NEXT STMT
380 MDX LI7      OPEN UP STRING
381 *
382 *
383 *                CHANGES THE ID WORD OF CONTINUE
384 *                STMTS DEFINING THE DO RANGE
385 *                TO THE DO TEST TYPE
386 *                OPENS THE STRING SIX MORE WORDS
387 MOVE1 LD DOTID CHANGE ID WORD
388 STO 1 0       TO DO TEST
389 *
390 *
391 MDX 1 2       MOVE POINTER
392 *
393 *                OPEN STRING 6 OR 7 WORDS
394 STX 1 TEM     SAVE INPUT POINTER
395 LD L EOF5    GET SIZE OF STRING
396 S          TEM TO BE MOVED
397 STO BOUND 1  STORE SIZE TO BE MOVED
398 RBOUND LDX L1 0 INITIALIZE SIZE CTR
399 MDX 1 1
400 LDX I2 EOF5  INITIALIZE LOOP INPUT PT
401 LD 3 3      LOAD DO INCREMENT
402 BSC L TRYMR,Z BRANCH IF NOT ZERO
403 MDX 2 7     MODIFY LOOP PT
404 LD LD7     LOAD INPUT INSTRUCTION
405 MDX CNWTH  MOVE STRING UP
406 TRYMR LD I PAVE LOAD STMT ID WORD
407 S FRO     SUBTRACT ONE FROM NORM
408 STO I PAVE PUT BACK IN STRING
409 LD LD6    LOAD INPUT INSTRUCTION
410 MDX 2 6   MODIFY LOOP POINTER
411 ONWTH STO LOOZ PUT INSTRUCTION IN LOOP
412 STX L2 EOF5 NEW END OF STRING ADDRESS
413 LCOZ LD 2 -6 LOAD WD ON STRING, THIS
414 *          *INSTRUCTION IS MODIFIAPL
415 STO 2 0   PUT IN NEW LOCATION
416 MDX 2 -1  MOVE PT
417 MDX 1 -1  DECREMENT CTR
418 MDX LOOZ  CONTINUE LOOP
419 LDX I1 TEM RESTORE IN PUT PT

```

```

420 *
421 *
422 *          CHECKS FOR A SYM TBL OVERLAP
423 *          INSERTS THE DO TEST CODING
424 *          INTO THE STMT STRING
425 OVLAP LD   L   EOFST      IS THERE
426         A   L   TWO        OVERLAP BETWEEN
427         S   L   EOF5       STRING AND SYMBOL TABLE
428         BSC L   JIMY, 7    BRANCH IF YES
429 *
430 *
431         LD   3 3          LOAD DO INCREMENT
432         BSC L   NASTY, -   BRANCH IF ZERO
433 *
434 *          INSERT DO TEST
435         LD   3 0          LOAD DO INDEX - COMBINE
436         AND   MASK5       WITH LOAD INSTRUCTION -
437         OR    LOADI       PUT ON STRING
438         STO  1 0          ***** LD      INDEX
439         LD   3 3          LOAD INCREMENT - COMBINE
440         AND   MASK5       WITH ADD INSTRUCTION - PU
441         OR    ADDI        ON STRING
442         STO  1 1          ***** A      INCREMENT
443         LD   3 0          LOAD INDEX - COMBINE WITH
444         AND   MASK5       STORE INSTRUCTION - PUT
445         OR    STOI        ON STRING
446         STO  1 2          ***** STC     INDEX
447 XYZ     LD   3 2          LOAD TEST VALUE - COMBINE
448         AND   MASK5       WITH SUBTRACT INSTRUCTION
449         OR    SUBTI       - PUT ON STRING
450         STO  1 3          ***** S      TEST VALU
451         LD   BSCI        PUT BSC I ON STRING
452         STO  1 4          PUT TEST STMT NO.
453         LD   3 1          ON STRING
454         STO  1 5          ***** BSC   I *--*, -
455         MDX  3 -5        REMOVE DO TABLE ENTRY
456 *
457 *
458         LDX  I1 WHY      RESTORE INPUT PT
459         BSC L   BAKER    CHECK FOR STMT NO.
460 *
461 *          INSERTS THE SPECIAL DO TEST INT
462 *          THE STRING WHEN THE INCREMENT
463 *          IS IMPLIED
464 NASTY LD   MDXI        PUT MDX I ON STRING
465         STO  1 0
466         LD   3 0          LOAD INDEX - PUT ON STRIN
467         STO  1 1          ***** MDX   I INDEX
468         LD   NOPI        LOAD NOP - PUT ON STRING
469         STO  1 2          ***** NOP
470         MDX  1 1          MOVE INPUT POINTER
471         LD   3 0          LOAD INDEX - COMBINE WITH
472         AND   MASK5       LOAD INSTRUCTION - PUT CN
473         OR    LOADI       STRING
474         STO  1 2          ***** LD      INDEX
475         MDX  XY7         INSERT REST OF TEST
476 *
477 *          CONSTANTS
478 TEM     DC   *--*      TEMPORARY STORAGE
479 FRO     DC   4         ONE COUNT FOR NOEM

```


480	DOID	DC	/0024	DO ID WORD
481	COLON	DC	/32	COLON
482	MASK	DC	/07FF	SYMBOL TABLE POINTER MASK
483	THRE	DC	3	THREE
484	MASK4	DC	/0200	MASK TO CHECK IF STANT NO
485	NORM4	DC	0	CONSTANT
486	MASK5	DC	/87FF	INSTRUCTION MASK
487	MDXI	DC	/5101	MDX I INSTRUCTION
488	LD6	LD	2 -6	LOAD INSTRUCTION
489	LD7	LD	2 -7	LOAD INSTRUCTION
490	LD8	LD	2 -8	LOAD INSTRUCTION
491	NOPI	DC	/6000	NO-OP INSTRUCTION
492	M2S	DC	/C01C	BASIC DC TEST ID WORD
493	*			
494	*			
495	*			CHECKS FOR A DO STMT. IF NOT
496	*			FOUND, OPENS THE STRING SEVEN
497	*			WORDS - INSERTS THE DO TEST
498	*			ID WORD
499	LIZ	LD	1 0	LOAD STMT ID WORD
500		SRA	2	GET STMT
501		AND	L IOTPE	ID TYPE
502		S	L DOXID	IS IT DO STMT
503		BSC	L LIZ1, -	BRANCH IF YES
504	*			
505	*			OPEN STRING 7 OR 8 WORDS
506		STX	1 TEM	SAVE INPUT POINTER
507		LD	L EOF5	GET SIZE
508		S	TEM	OF STRING
509		A	L ONE	TO BE
510		STO	BCUN 1	MOVED
511	BCUN	LDX	L1 0	INITIALIZE STRING SIZE CT
512	*			*MODIFIABLE
513		LDX	I2 EOF5	INITIALIZE LOOP INPUT PT
514		LD	3 3	LOAD DO LOOP INCREMENT
515		BSC	L MRXY,7	BRANCH NOT ZERO
516		MDX	2 8	MODIFY LOOP POINTER
517		LD	LD8	LOAD INPUT INSTRUCTION
518		MDX	CNCN	GO STORE INPUT INSTRUCTIO
519	MRXY	LD	LD7	LOAD INPUT INSTRUCTION
520		MDX	2 7	MODIFY LOOP POINTER
521	ONON	STO	LOOT	SAVE INPUT INSTRUCTION
522		STX	L2 EOF5	NEW END OF STRING ADDRESS
523	LOOT	LD	2 -7	MOVE WD UP. THIS INST IS
524	*			*MODIFIABLE
525		STO	2 0	TO NEW LOCATION
526		MDX	2 -1	MOVE INPUT POINTER
527		MDX	1 -1	DECREMENT COUNTER
528		MDX	LOOT	CONTINUE LOOP
529	*			
530	*			INSERT DO TEST ID WORD
531		LDX	I1 TEM	RESET INPUT POINTER
532		LD	LD7	GET SIZE OF INSTRUCTION
533		S	LOOT	STRING TO BE OUTPUTTED
534		SLA	2	COMBINE THIS SIZE WITH
535		A	M2S	DO ID WORD
536		STO	1 0	PLACE ON STRING
537	*			
538	*			
539		MDX	1 1	MOVE INPUT POINTER

```

540          MDX      OVLAP      CHECK OVERLAP
541 *
542 *          CONSTANTS
543 ERR53 DC      53      ERROR NO. 53
544 PAVE DC      0      STORAGE FOR ID WORD ADDR
545 X32 DC      32      DEFINED VARIABLE BIT
546 *
547 *          INITIALIZES TO SCAN THE STMNT
548 *          GETS THE STMNT NO. FROM THE
549 *          SYM TBL
550 CEE MDX      3 5      MOVE TBL PTR TO NEXT POS
551 LD      1 0      LOAD STMNT ID WORD
552 BSC      E      SKIP IF NO STMNT NO.
553 MDX      1 1      MOVE PT OVER STMNT NO.
554 MDX      1 1      MOVE POINTER
555 *
556 *
557 LD      1 0      LOAD WORD
558 BSI L CHECK      GET SYMBOL TABLE ID WORD
559 AND      MASK4      IS IT
560 EOR      MASK4      STMNT NUMBER
561 BSC L HAF, -      BRANCH IF YES
562 *
563 *
564 ER53 DC      *-*      ENTRY POINT
565 EOR      H4080      CHECK IF SUPPROGRAM
566 BSC L R53, -      NO, GO INDICATE ERROR
567 LD L SORF      IS IT FUNCTION SUBPROGRAM
568 BSC L R53,      BRANCH IF NOT
569 LD L FNAME      IS IT THIS ONE
570 S      1 0
571 BSI I ER53, -      GO BACK IF OK
572 R53 LD      EPR53      SET UP
573 BSC L CLOS1 1      ERROR NO. 57
574 *
575 *
576 *          PUTS THE DO RANGE STMNT NO. INT
577 *          WORD 5 OF THE DO TBL ENTRY
578 *          CHECKS FOR VALID VARIABLES
579 *          INDICATES DEFINITION OF ALL
580 *          DEFINED VARIABLES
581 HAF LD      1 0      LOAD STMNT NUMBER
582 STO      4      PUT IN
583 *
584 MDX      1 1      MOVE POINTER
585 *
586 *
587 LD      1 0      LOAD WORD
588 S      COLCN      IS CHARACTER COLCN
589 BSC L R53,7      BRANCH IF NOT
590 *
591 *
592 MDX      1 1      MOVE STRING POINTER
593 *
594 *          LEGAL VARIABLE
595 LD      1 0      LOAD WORD
596 BSI L CHECK      GET TBL ID WORD
597 STO      SAVES      SAVE SYM TBL ID WORD
598 AND L HOBDE      IS IT LEGAL,
599 EOR      MASK1      INTEGER VARIABLE

```

600		BSI	L	ER53,Z	BRANCH IF NOT
601	*				
602	*				
603		LD		SAVES	LOAD STMT ID WORD
604		OR		X32	PUT IN DEFINED VAR BIT
605		STO	2	0	PUT BACK IN SYMBOL TABLE
606	*				
607	*				
608		LD	L	SORF	LOAD SORF
609		BSC			SKIP IF FUNCTION
610		MDX		PEND	PUT VARIABLE IN DO TABLE
611		LD		SAVES	LOAD SYM TBL ID WORD
612		AND		MASK7	IS VAR IN COMMON OR DUMMY
613		BSC	L	PEND, -	ARGUMENT - BRANCH IF NO
614	*				
615	*				
616		LD		ERR55	SET UP ERROR NO. 55
617		BSC	L	CLOS1 1	PUT ERROR STMT CN STRING
618	*				
619	*				
620	*				PLACES THE INDEX VARIABLE INTO
621	*				WORD 1 OF THE DO TBL ENTRY
622	PEND	LD	1	0	LOAD VARIABLE
623		STO	3	0	PUT IN DO TABLE
624	*				
625	*				
626		MDX	1	1	MOVE STRING POINTER
627	*				
628	*				
629		LD	1	0	LOAD WORD
630		S		EQAL	IS CHARACTER EQUAL SIGN
631		BSC	L	R53,7	BRANCH IF NOT
632	*				
633	*				
634		MDX	1	1	MOVE POINTER
635	*				
636	*				
637		LD	1	0	LOAD WORD
638		BSI	L	CHECK	GET SYMBOL TABLE ID WORD
639		AND		MASK3	IS IT A VALID CONSTANT OF
640		EOR		MASK1	VARIABLE INTEGER
641		BSI	L	ER53,Z	BRANCH IF NOT
642	*				
643	*				
644		LD	2	1	LOAD WORD
645		BSC		-	SKIP IF NON-ZERO
646		MDX		AA	ERROR IF ZERO
647	*				
648	*				SAVE M1 FOR DO INITIALIZE
649	AAA	LD	1	0	LOAD INITIAL VALUE
650		STO		M1	OF INDEX
651	*				
652	*				
653		MDX	1	1	MOVE POINTER
654		MDX		A	GET TEST VALUE
655	*				
656	*				
657	AA	LD		ERR54	SET UP ERROR NO. 54
658		BSC	L	CLOS1 1	PUT ERROR STMT CN STRING
659	*				

				CONSTANTS	
660	*				
661	SAVES	DC	0		STORAGE FOR SYM TBL ID
662	H4080	DC	/4080		SUBR SYM TBL ID
663	H8000	DC	/8000		USEFUL CONSTANT
664	MASK1	DC	/4000		MASK TO TEST FOR INTEGER
665	MASK7	DC	/2400		COMMON OR DUMMY ARG MASK
666	M1	DC	0		INITIAL INDEX VALUE
667	EQAL	DC	/000E		EQUAL SIGN
668	MASK3	DC	/5BDE		CON OR VAR INTEGER MASK
669	COME	DC	/0012		COMMA
670	DOIID	DC	/580C		DO ID TYPE - NORM 3
671	CATOR	DC	/0220		SYM TBL ID FOR STMT NO.
672	SIGN	DC	/8001		SIGN BITS
673	ERR54	DC	54		ERROR NO. 54
674	ERR55	DC	55		ERROR NO. 55
675	*				
676	*				
677	*				PLACES A ZERO AS THE INCREMENT
678	*				INTO WORD 4 OF THE DO TBL ENTRY
679	*				PUTS THE TEST VALUE CONSTANT
680	*				INTO WORD 3 OF THE DO TBL ENTRY
681	DEMP	SLA	16		PUT ZERO IN DC TABLE AS
682		STO	3 3		VALUE OF INCREMENT
683		MDX	SEM		GO TO END OF STMT
684	*				
685	*				
686	A	LD	1 0		LOAD WORD
687		S	COME		IS CHARACTER COMMA
688		BSC	L R53,7		BRANCH IF NO
689	*				
690	*				
691		MDX	1 1		MOVE STRING POINTER
692	*				
693	*				
694		LD	1 0		LOAD WORD
695		BSI	L CHECK		GET SYM TBL ID WORD
696		AND	MASK3		IS IT VALID INTEGER
697		EOR	MASK1		CONSTANT OR VARIABLE
698		BSI	L ER53,7		BRANCH IF NO
699	*				
700	*				
701		LD	1 0		PUT INTEGER IN DC TABLE
702		STO	3 2		AS TEST VALUE
703	*				
704	*				
705		MDX	1 1		MOVE STRING POINTER
706	*				
707	*				
708		LD	1 0		LOAD CHARACTER
709		S	COME		IS CHARACTER COMMA
710		BSC	L DEMP,7		BRANCH IF NO
711	*				
712	*				
713		MDX	1 1		MOVE STRING POINTER
714	*				
715	*				
716		LD	1 0		LOAD INCREMENT
717		BSI	L CHECK		GET SYM TBL ID WORD
718		AND	MASK3		IS IT VALID CONSTANT OR
719		EOR	MASK1		VARIABLE INTEGER

720		BSI	L	ER53,7	BRANCH IF NOT
721	*				
722	*				
723		LD	1	0	PUT INTEGER IN DO TABLE
724		STO	3	3	AS INCREMENT
725	*				
726	*				
727		MDX	1	1	MOVE STRING POINTER
728	*				
729	*				
730	*				
731	*				PLACES THE DO INITIALIZE CODING
732	*				ON THE STRING
733	*				CLOSES UP THE STRING
734	SEM	LD	1	0	LOAD CHARACTER
735		BSC	L	R53,7	BRANCH IF NOT SEMI-COLON
736	*				
737	*				PUT OUT DO INITIALIZE
738		MDX	1	1	MOVE STRING PT TO NEXT
739		STX	1	VEN 1	STMNT - SAVE ADDRESS
740		LDX	I1	PAVE	LOAD PRESENT STMNT ID ADD
741		LD	1	0	LOAD STMNT ID WORD
742		SRA		2	GET STMNT NORM
743		AND	L	IDNRM	
744		STO	L	NRMSV	SAVE NORM
745		LD	1	0	LOAD STMNT ID WORD
746		BSC		E	SKIP IF NO STMNT NO.
747		MDX		XYZ1	PUT IN CORRECT ID WORD
748		LD		DOIID	LOAD DO STMNT ID WORD
749		STO	1	0	PUT ON STRING
750	XYZ2	LD		M1	LOAD INITIAL INDEX VALUE-
751		OR	L	LOADI	COMBINE WITH LOAD INST
752		STO	1	1	***** LD INITIAL
753		LD	3	0	LOAD INDEX - COMBINE WITH
754		OR	L	STCI	STORE INSTRUCTION - OUTPU
755		STO	1	2	***** STC INDEX
756		MDX	1	2	MOVE STRING POINTER
757	*				
758	*				CLOSE UP STRING
759		STX	3	HEN 1	SAVE DO TABLE POINTER
760	VEN	LDX	L2	0	LOAD ADDR OF NEXT STMNT
761	*				*MODIFIABLE
762		LD	L	EOFS	GET SIZE
763		S	L	PAVE	OF STRING
764		S	L	NRMSV	TO BE MOVED
765		STO		VEN1 1	PUT IN COUNTER
766	VEN1	LDX	L3	0	INITIALIZE COUNTER
767	*				*MODIFIABLE
768		MDX	3	1	
769	POOL	LD	2	0	MOVE WORD DOWN
770		STO	1	1	IN STRING
771		MDX	1	1	MOVE POINTERS
772		MDX	2	1	
773		MDX	3	-1	DECREMENT COUNTERS
774		MDX		POOL	CONTINUE LOOP
775		STX	L1	EOFS	NEW END OF STRING ADDRESS
776		LDX	I1	PAVE	RESTORE INPUT POINTER
777	HEN	LDX	L3	0	RESTORE DO TABLE POINTER
778	*				*MODIFIABLE
779	*				

```

780 *
781 *          CHECKS FOR A NUMBERED STMT
782 *          FOLLOWING THE DO STMT
783 *          IF NUMBERED STMT NOT FOUND,
784 *          GENERATES A LABEL AND INSERTS
785 *          IT IN THE SYM TBL
786 MUIT  BSI  L  MOVE          MOVE TO NEXT STMT
787 *
788 *
789          LD   1 0          DOES STMT HAVE STMT NO.
790          BSC  L  HOHO,E     BRANCH IF YES
791 *
792 *
793          A     F5          ADD STMT NO. BITS TO ID
794          STO  1 0          PUT ON STRING
795 *
796 *
797          MDX  1 1          MOVE STPING POINTER
798 *
799 *
800          BSI  L  ROUT          OPEN STRING ONE WORD
801 *
802 *
803          LDX  I2 EOFST      LOAD END OF SYM TBL ADDR
804          SLT          16     CLEAR NEXT TWO
805          STO  2 1          WORDS IN
806          STO  2 2          SYMBOL TABLE
807          LD   CATOR        LOAD STMT NO. SYM TBL ID
808          STO  2 0          PUT IN SYM TBL
809          LD   L  SOFST      GET SIZE OF
810          S     L  EOFST      SYMBOL TABLE
811          SRT          16     POSITION FOR DIVIDE
812          D     L  THRE      GET NO. OF
813          A     SIGN          SYMBOL TABLE ENTRY
814          STO  1 0          PUT IN STRING
815          MDX  1 -1          MOVE PT TO STMT ID WORD
816          MDX  L  EOFST,-3   MOVE SYMBOL TABLE POINTER
817          MDX  L  SOFXT,-3
818          MDX  L  SOFGT,-3
819 *
820 *
821          LD   L  EOFST      CHECK FOR
822          A     L  TWO        OVERLAP BETWEEN STRING
823          S     L  EOFS       AND SYMBOL TABLE
824          BSC  L  JIMY, Z     BRANCH IF OVERLAP
825 *
826 *
827 *          PLACES THE GENERATED LABEL OR
828 *          STMT NO. FOR THE DO TEST INTO
829 *          WORD 2 OF THE DO TABLE ENTRY
830 HAH    LD   1 1          LOAD STMT NO.
831          STO  3 1          PUT IN SYMBOL TABLE
832          BSC  L  ABEL        CHECK NEXT STMT.
833 *
834 *
835 XYZ1   LD   L  DOIIP        LOAD DO STMT ID
836 *          PLACES THE DO ID WORD INTO THE
837 *          STATEMENT STRING
838          A     F5          PUT IN STMT NO. BITS
839          STO  1 0          PUT ON STRING

```

840		MDX	1 1	MOVE PT OVER STMT NO.
841		MDX	XYZZ	RETURN
842	*			
843	*			
844	*			CHECKS FOR FORMAT STATEMENTS
845	HOHO	SRA	11	GET STMT ID TYPE
846		S	HC	IS IT FORMAT STMT
847		BSC	L MUIT, -	BRANCH IF YES
848		MDX	HAH	PUT STMT NO. IN DO TBL
849	*			
850	*			CONSTANTS
851	HC	DC	/0	FORMAT STMT ID TYPE
852	F5	DC	5	BITS FOR STMT NO.
853	CALN1	DC	/1A80	BACKSPACE CALL
854	CALN2	DC	/1B00	REWIND CALL
855	CALN3	DC	/1A80	END FILE CALL
856	CALN4	DC	/1B80	STOP CALL
857	CALN5	DC	/1C00	PAUSE CALL
858	CALN0	DC	0	CALL STORAGE
859	NORM2	DC	0	PAUSE OR STOP SWITCH
860	ERR56	DC	56	ERROR NO. 56
861	ERR57	DC	57	ERROR NO. 57
862	ERR58	DC	58	ERROR NO. 58
863	ERR59	DC	59	ERROR NO. 59
864	ERR60	DC	60	ERROR NO. 60
865	*			
866	*			
867	CALL1	LD	CALN1	STORE BACKSPACE CALL
868		STO	CALNC	IN CALNO
869	*			
870	*			
871		LD	ERR56	SET UP
872		STO	L ERRNC	ERROR NO. 56
873		MDX	TED	PUT IN DEVICE NO.
874	*			
875	*			
876	CALL2	LD	CALN2	PUT REWIND CALL
877		STO	CALNC	IN CALNO
878	*			
879	*			
880		LD	ERR57	SET UP
881		STO	L ERRNC	ERROR NO. 57
882		MDX	TED	PUT IN DEVICE NO.
883	*			
884	*			
885	CALL3	LD	CALN3	PUT END FILE
886		STO	CALNC	CALL IN CALNO
887	*			
888	*			
889		LD	ERR58	SET UP
890		STO	L ERRNC	ERROR NO. 58
891		MDX	TED	PUT IN DEVICE NO.
892	*			
893	*			
894	CALL4	LD	CALN4	PUT STOP CALL
895		STO	CALNC	IN CALNO
896	*			
897	*			
898		LD	ERR59	SET UP
899		STO	L ERRNC	ERROR NO. 59

```

900 *
901 MDX FED CHECK VALID NO.
902 *
903 *
904 CALL5 LD CALN5 PUT PAUSE CALL
905 STO CALNO IN CALNO
906 *
907 *
908 LD ERR60 SET UP
909 STO L ERRNC ERROR NO. 60
910 *
911 *
912 FED MDX L NORM2,1 TAG PAUSE OR STOP SWITCH
913 *
914 * INITIALIZES TO SCAN THE STMNT
915 * CHECKS FOR THE SEMICOLON
916 * OPENS THE STRING ONE WORD
917 * ADJUSTS THE STMNT NORM
918 * PLACES THE SYM TBL ADDR OF THE
919 * CONSTANT ON THE STRING
920 TED LD 1 0 LOAD STMNT ID WORD
921 BSC E SKIP IF NO STMNT NO.
922 MDX 1 1
923 MDX 1 1 MOVE POINTER
924 LD 1 0 LOAD WORD
925 BSI L CHECK GET SYM TBL ID WORD
926 STO HORRS SAVE ID WORD
927 *
928 *
929 LD NORM2 PAUSE OR STOP STMNT
930 BSC L SHOT, - BRANCH IF NOT
931 *
932 *
933 LD 1 0 LOAD WORD
934 BSC L SHOE,Z BRANCH NOT SEMI-COLON
935 *
936 *
937 BSI L ROUT OPEN STRING ONE WORD
938 *
939 *
940 LD I PAVE LOAD STMNT ID WORD
941 A FOUR ADD ONE TO NORM
942 STO I PAVE PUT BACK IN STRING
943 *
944 *
945 BSI L BOXX PUT SYM TBL ADDR ON STRIN
946 MDX AMIN2 2 PUT CALL ON STRING
947 *
948 * LEGAL VARIABLE
949 SHOT LD HORRS LOAD SYM TBL ID WORD
950 AND HORRE IS IT VALID
951 EOR L MASK1 INTEGER VARIABLE
952 BSC L SHOE,Z BRANCH IF NOT
953 *
954 * PUT OUTPUT IN STRING
955 AMIN2 LD 1 0 LOAD DEVICE NO.
956 STO 1 1 PUT IN STRING
957 LD CALNO LOAD CALL
958 STO 1 0 PUT IN STRING
959 ZAMIN LOX I1 PAVE PLACE PT AT STMNT ID WORD

```



```

960 *
961 *
962     SLA     16
963     STO L   DEC     CLEAR
964 *
965 *
966     STO     NORM2   CLEAR PAUSE OR STOP SWITC
967     BSC L   BAKER   CHECK FOR STMT NO.
968 *
969 *
970 *
970 MASK2 DC     /0000   INTEGER CONSTANT MASK
971 HOBRS DC     0       SYM TBL ID WORD
972 FCUR DC     4       CON TO ADD ONE TO NORM
973 DEVIL DC    0       TEMPORARY STORAGE
974 ZEMP DC     0       TEMPORARY STORAGE
975 C99 DC     9999   MAXIMUM VALUE OF PAUSE
976 ERR61 DC    61     ERROR NO. 61
977 CCN5 DC    /0014-70006 ERROR TYPE
978 CCN6 DC    /0000   FORMAT ID TYPE
979 ER62 DC    62     ERROR NO. 62
980 END DC     /1004   END STMT ID TYPE
981 SAVE2 DGE
982 SAVE1 DC     0       TEMPORARY STORAGE
983 SAVE DC     0       STORAGE FOR STRING POINTE
984 MASK6 DC    /01FF   NORM MASK
985 HDBDE DC    /DBDE   MASK FOR INTEGER VARIABLE
986 *
987 *
988 SHOE LD     HOBRS   LOAD SYM TBL ID WORD
989     EOR     MASK2   IS IT VALID INTEGER CON
990     BSC L   CLOS1 2,Z BRANCH IF NOT
991 *
992 *
993 LD     1 1       LOAD NEXT WORD
994 BSC L   CLOS1 2,Z BRANCH IF NOT SEMI-COLON
995 *
996 *
997 LD     2 1       LOAD CONSTANT
998 S     C99       IS IT GREATER THAN 9999
999 BSC L   ER61,7- BRANCH IF YES
1000 *
1001 *
1002 LD     NORM2   IS STMT PAUSE OR STOP
1003 BSC L   AMIN2, - BRANCH IF NOT
1004 *
1005 *
1006 LD     2 1       LOAD CONSTANT
1007 S     L TEN     IS IT GREATER THAN 10
1008 BSC L   AMIN2, Z BRANCH IF NOT
1009 *
1010 *
1011 LD     2 1       LOAD WORD
1012 MOX    BTOD    CONVERT TO DECIMAL
1013 *
1014 *
1015 *
1016 OUT1 STX    1 SAVE CHECK FOR TRANSFER STMT
1017 LD     L SOFS  PRIOR TO THE END STMT
1018 S     SAVE    SAVE ADDR OF END STMT
1019 BSC L   OUT, - IS END STMT THE
                    FIRST STMT IN STRING
                    BRANCH IF YES

```

1020		LDX	I2	SOFS	INITIALIZE CHECK POINTER
1021	OUT1A	STX	2	SAVE2	SAVE CHECK POINTER
1022		LD	2	0	LOAD STMT ID WORD
1023		SRA		2	GET NORM
1024		AND		MASK6	OF STATEMENT
1025		STO		OUT2 1	STORE NORM
1026	OUT2	MDX	L2	0	MOVE PT TO NEXT STMT
1027	*				*MODIFIABLE
1028		STX	2	SAVE1	SAVE NEW PT ADDRESS
1029		LD		SAVE1	IS NEW ADDRESS THE ADDRES
1030		S		SAVE	OF THE END STATEMENT
1031		BSC		Z	SKIP IF YES
1032		MDX		OUT1A	CHECK NXT STMT ON STRING
1033		LDX	I1	SAVE2	ADDR OF STMT BEFORE END
1034		LD	1	0	LOAD STMT ID WORD
1035		SRA		11	GET STMT ID TYPE
1036		S		CONF	IS IT FORMAT STMT--IF YE
1037		BSC	L	OUT1, -	CHECK PREVIOUS STMT
1038		BSI	L	TRCKS	CHECK FOR TRANSFER STMT
1039		DC		OUT	GO TO NEXT PHASE
1040		DC		*	GO TO NEXT INSTRUCTION
1041		S		CONF	IS IT ERROR STMT
1042		BSC	L	OUT, -	BRANCH IF YES
1043	*				
1044	*				
1045	TEST	LDX	I1	EOFS	LOAD PT WITH END OF STRIN
1046		LD	L	ERRID	PUT ERROR STMT ID WORD
1047		STO	1	0	AT END OF STRING
1048		LD		ER62	PUT ERROR NO. 62
1049		STO	1	1	ON STRING
1050		LD		END	PUT END STMT ID WORD
1051		STO	1	2	ON STRING
1052		MDX	1	2	MOVE STRING POINTER
1053		STX	L1	EOFS	NEW END OF STRING ADDRESS
1054		BSC	L	CUT	GO TO NEXT PHASE
1055	*				
1056	*				
1057	ER61	SLA		16	SET UP ERROR NO. 61
1058		STO	L	NORM2	TURN OFF PAUSE
1059		LD		ERR61	OR STOP SWITCH
1060		BSC	L	CLOS1 1	PUT ERROR ON STRING
1061	*				
1062	*				THIS IS A SUBROUTINE TO GET SYMBOL
1063	*				TABLE ADDRESS ID AND LOAD ACCUMULATO
1064	*				
1065	CHECK	DC		0	ENTRY POINT
1066		BSC	L	DEMPY 4,-	RETURN IF SPECIAL OPERATO
1067		LD	1	0	LOAD WORD
1068		AND	L	MASK	GET SYM TBL ENTRY NO.
1069		M	L	THRE	GET ADDRESS OF ENTRY
1070		SLT		16	RELATIVE TO START
1071		STO		DEVIL	OF SYMBOL TABLE
1072	*				
1073		LD	L	SOFS	GET ABSOLUTE ADDRESS
1074		S		DEVIL	OF SYMBOL TABLE ENTRY - 3
1075		STO		DEMPY 1	STORE ABSOLUTE ADDRESS
1076	DEMPY	LDX	L2	0	GET ADDRESS OF ENTRY ID
1077	*				*WORD. MODIFIABLE
1078		MDX	2	3	
1079		LD	2	0	LOAD SYM TBL ID WORD

```

1080          BSC  I  CHECK          RETURN
1081  *
1082  *          THIS IS A SUBROUTINE TO MOVE THE
1083  *          POINTER TO FOLLOWING STATEMENT
1084  MOVE  DC          0          ENTRY POINT
1085          LD    1  0          LOAD STMT ID WORD
1086          SRA   2          GET NORM
1087          AND  L  IDNRM        OF STATEMENT
1088          STO   NXID 1        GET ADDR OF NEXT
1089  NXID  MDX  L1          STMT ID WD. MODIFIABLE
1090          BSC  I  MOVE          RETURN
1091  *
1092  *
1093  *          OPENS THE STRING ONE WORD FOR
1094  *          INSERTION OF A GENERATED LABEL
1095  *
1096  ROUT  DC          0          ENTRY POINT
1097          STX   1  ZEMP        SAVE STRING POINTER
1098          LD    L  EOF5        GET SIZE OF
1099          S     L  ZEMP        STRING TO
1100          A     L  ONE         BE MOVED
1101  *
1102          STO   BOND 1        PUT IN COUNTER
1103  BOND  LDX  L1  0          LOAD COUNTER. MODIFIABLE
1104          LDX  I2  EOF5        GET NEW SIZE
1105          MDX  2  1          OF STRING
1106          STX  L2  EOF5        NEW END OF STRING ADDR
1107  PUZ   LD    2  -1         LOAD WORD AND MOVE
1108          STO   2  0          IT UP ONE WORD
1109          MDX  2  -1         MOVE POINTER
1110          MDX  1  -1         DECREMENT COUNTER
1111          MDX   PUZ          CONTINUE LOOP
1112          LDX  I1  ZEMP        RESET INPUT POINTER
1113          BSC  I  ROUT        RETURN
1114  *
1115  *          CONVERT BINARY TO DECIMAL
1116  BTOD  STX   1  BOX 1        SAVE STRING POINTER
1117          LDX   1  4          POSITION FOR DIVIDE
1118  REP   SRT   15          DIVIDE BY TEN
1119          D     TEN          DIVIDE BY TEN
1120          STO   WORK        SAVE QUOTIENT
1121          LD    DEC          LOAD DECIMAL NO.
1122          RTE   4          BRING REMAINDER TO ACC
1123          STO   DEC          SAVE DECIMAL NO.
1124          LD    WORK        LOAD QUOTIENT
1125          MDX  1  -1         DECREMENT COUNT OF DIVIDE
1126          MDX   REP          GET NEXT NO. IN CONVERSIO
1127  BOX   LOX  L1  0          RESTORE INPUT POINTER
1128  *          *MODIFIABLE
1129  *
1130  *
1131          BSI   BOXX        PUT SYM TBL ADDR ON STRIN
1132          BSC  L  AMIN2 2    PUT CALL ON STRING
1133  *
1134  *          PUTS THE POINTER TO A CONSTANT
1135  *          IN THE STRING - PLACES THE
1136  *          CONSTANT INTO THE SYMBOL TABLE
1137  *          CHECKS FOR A SYM TBL OVERLAP
1138  BOXX  DC          0          ENTRY POINT
1139          STX   3  ZEMP1 1    SAVE INDEX REGISTER 3

```

1140	LDX	J3	EOFST	LOAD END OF SYM TBL ADDR
1141	STX	3	WORK	PUT IN WORK
1142	LD	L	SOFST	GET SIZE OF
1143	S		WORK	SYMBOL TABLE
1144	SRT		16	POSITION FOR DIVIDE
1145	D	L	THRE	GET ENTRY NO.
1146	A		SIGN1	FOR NEW ENTRY
1147	STO	1	1	STORE IN STRING
1148	*			
1149	*			PUT CONSTANT ON SYMBOL TABLE
1150	LD		DEC	PUT DECIMAL CONSTANT IN
1151	STO	3	1	SYMBOL TABLE
1152	LD	L	MASK2	PUT SYM TBL ID WORD
1153	STO	3	0	IN SYMBOL TABLE
1154	SLA		16	PUT ZEROES AS SECOND WORD
1155	STO	3	2	OF SYM TBL NAME
1156	*			
1157	*			
1158	MDX	L	EOFST,-3	MODIFY SYMBOL
1159	MDX	L	SOFXT,-3	TABLE ADDRESSES
1160	MDX	L	SOFGT,-3	
1161	*			
1162	*			
1163	LD	L	EOFST	TEST FOR OVERLAP
1164	A	L	TWO	BETWEEN STRING AND
1165	S	L	EOFS	SYMBOL TABLE
1166	BSC	L	JIMY, Z	BRANCH IF OVERLAP
1167	ZEMP1 LDX	L	3 0	RESTORE XR3. MODIFIABLE
1168	BSC	I	BOXY	RETURN
1169	*			
1170	*			CONSTANTS
1171	DEC	DC	0	DECIMAL CONVERSION WORD
1172	WORK	DC	0	TEMPORARY STORAGE
1173	TEN	DC	10	TEN
1174	SIGN1	DC	/8001	SIGN FOR SYM TBL
1175	*			
1176	*			SUBROUTINE TO CHECK A STATEMENT
1177	*			ID WORD FOR A TRANSFER STATEMENT.
1178	*			ROUTINE BRANCHES INDIRECT THROUGH
1179	*			FIRST OPERAND IF TRANSFER, AND
1180	*			SECOND IF NOT.
1181	*			
1182	TRCKS	DC	0	ENTRY POINT
1183	LD		TRCKS	LOAD RETURN ADDRESS
1184	STO		WRT1 1	STORE IN TRANSFER EXIT
1185	A		WR111	ADD ONE
1186	STO		WRT2 1	STORE IN NON-TRANSFER EXI
1187	LD	1	0	LOAD WORD
1188	SRA		11	GET ID TYPE
1189	S		CON1	IS IT GO TO
1190	BSC	L	WRT1, -	BRANCH IF YES
1191	S		CON2	IS IT STOP
1192	BSC	L	WRT1, -	BRANCH IF YES
1193	S		CON3	IS IT RETURN
1194	BSC	L	WRT1, -	BRANCH IF YES
1195	S		NCON1	IS IT CALL LINK
1196	BSC	L	WRT1,-	BRANCH IF YES
1197	S		CON4	IS IT IF
1198	BSC	L	WRT1, -	BRANCH IF YES
1199	S		CON7	IS IT SUPR CALL

1200	BSC	L	WRT2,7	CHECK SUBR NAME IF SO
1201	LD	1	0	GET ID WORD OF STIPING STMN
1202	SLA		16	AND PUT STMT NO BIT IN
1203	LD	1	1	CARRY. GET SYMT POINTER TO
1204	BSC		0	STMT NO. SKIP ON C=OFF
1205	LD	1	2	GET SMYT PT. SHIFT
1206	SLA		6	PT - REMOVE EXTRANEIOUS BIT
1207	SRA		5	SOFST-3*N-3=POS OF 1ST
1208	STO		STEN	WD OF SYMBOL TABLE ENTRY
1209	SRA		1	*
1210	A		STEN	*
1211	S	L	SOFST	* COMPUTE THE ADDRESS OF
1212	EOR		HFFFF	* THE FIRST WORD OF THE
1213	A		FIVE	* NAME IN THE SYMBOL TABLE
1214	STO		LN1+1	*
1215	A		WON	*
1216	STO		LN2+1	*
1217	LN1	LD	L *-*	GET FIRST WORD
1218		STO	N1	SAVE IT
1219	LN2	LD	L *-*	GET SECOND WORD
1220		STO	N2	SAVE IT
1221		STX	2 SX2	SAVE XR2
1222		LDX	2 ENTAB-BNTAB	GET A WORD COUNT
1223	CNN	LDD	L2 BNTAB-2	GET A 2 WD NAME TBL ENTR
1224		SD	N1	COMPARE TO STRING NAME
1225		BSC	L WRT1X,+-	BRANCH ON COMPARE
1226		MDX	2 -2	ELSE MOVE XR2
1227		MDX	CNN	LOOP BACK
1228		LDX	I2 SX2	RESTORE XR2
1229	WRT2	BSC	I *-*	EXIT IF NOT TRANSFER STMT
1230	*			*MODIFIABLE
1231	WRT1X	LDX	L2 *-*	SET XR2
1232	SX2	EQU	*-1	MODIFIABLE
1233	WRT1	BSC	I *-*	EXIT IF TRANSFER STMT
1234	*			*MODIFIABLE
1235	*			
1236	*		CONSTANTS	
1237	WR111	DC	1	ONE
1238	CCN1	DC	/000E	GO TO
1239	CON2	DC	/0017-/000E	STOP
1240	CON3	DC	/0010-/0017	RETURN
1241	NCON1	DC	/0010-/0010	CALL LINK
1242	CON4	DC	/000F-/0010	IF
1243	CCN7	DC	/0006-/000F	CALL
1244	N1	BSS	E 1	CALLED SUBR NAME IN SYMBOL
1245	N2	DC	*-*	TABLE FORMAT
1246	TCO	DC	2	CONSTANT
1247	WON	DC	1	CONSTANT
1248	BNTAB	EQU	*	SPECIAL TRANSFER SUBRS.
1249		DC	/0408	.CALL SPECL.
1250		DC	/0003	
1251		DC	/8408	.CALL BACK.
1252		DC	/8480	
1253		DC	/8098	.CALL QLEVL.
1254		DC	/0253	
1255		DC	/8828	.CALL DEQUE.
1256		DC	/8905	
1257		DC	/8698	.CALL CLERO.
1258		DC	/0658	
1259	ENTAB	EQU	*	

1260	STEN	DC	*-*	2*SYMT ENTRY NUM
1261	HFFF	DC	/FFF	FOR-ING CCNSTANT
1262	FIVE	DC	/5	CONSTANT
1263		BSS	OVERL-**320*4	PHASE-14 PATCH AREA
1264		END	START	