

```

1 * GENERAL AUTOMATION, INC, ALL RIGHTS RESERVED
2 *****
3 *
4 * PROGRAM NAME FPH-16
5 *
6 * MODEL NUMBER 8F016
7 *
8 * PURPOSE FORTTRAN PHASE-16
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 FORTTRAN ** SCAN ** PHASE FR16
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 * * CONVERTS ALL READ, FIND, WRITE, IF, GO TO,
27 * CALL, STATEMENT FUNCTION, AND ARITHMETIC
28 * STATEMENTS INTO MODIFIED POLISH NOTATION,
29 * * ESTABLISHES THE ORDER OF ARITHMETIC OPERA-
30 * TIONAL PERFORMANCE.
31 * * SETS UP THE ARGUMENTS FOR SUBROUTINE CALLS
32 * TO BE GENERATED.
33 *
34 *ENTRY POINTS-
35 * START-PHASE 16 IS READ INTO CORE BY PHASE 15
36 * TRANSFER TO ROLRX, EXECUTION IS BEGUN
37 * AT LOCATION START,
38 *
39 *INPUT-THE STATEMENT STRING MAY BE CONSIDERED
40 * INPUT
41 *
42 *OUTPUT-THE STATEMENT STRING AND THE SYMBOL TABLE
43 * AS ALTERED BY THE PHASE MAY BE CONSIDERED
44 * OUTPUT
45 *
46 *EXTERNAL REFERENCES-N/A
47 *
48 *EXITS-
49 * NORMAL-
50 * AT LOCATION GOSS TRANSFER IS MADE TO THE
51 * INTERPHASE ROLLER ROUTINE WHICH LOADS
52 * PHASE 17,
53 * ERROR-
54 * THE FLOW OF COMPILER PROCESSING IS NOT
55 * INTERRUPTED WHEN AN ERROR IS ENCOUNTERED.
56 * AN ERROR STMT REPLACES THE FAULTY STMT
57 * ON THE STRING AND THE STMT SCANNING
58 * PROCESS CONTINUES, AN EXCEPTION OCCURS
59 * WHEN AN OVERLAP ERROR IS DETECTED. COMPIL

```

60 * ACTION IS THEN ABORTED AND CONTROL IS PASS-
 61 * ED FROM PHASE TO PHASE, EACH PHASE IMMED-
 62 * IATELY EXITING TO THE NEXT UNTIL PHASE 21
 63 * PRINTS THE OVERLAP ERROR MESSAGE.
 64 * THE ERROR DETECTED BY PHASE 16 IS NO. 64

65 *
 66 * TABLES/WORK AREAS-

67 * STRING
 68 * SYMBOL TABLE
 69 * FORTRAN COMMUNICATIONS AREA
 70 * FORCING TABLE-2 WORD ENTRIES
 71 * WORK 1-CONTAINS RT AND LFT FORCING VALUES
 72 * FOR EACH OPERATOR
 73 * WORD 2-PTR TO STRING OF OPS USED BY INTERP
 74 * PNT-OUTPUT BUFFER FOR MODIFIED EXPRESSION
 75 * PDL-PUSH DOWN LIST FOR STORING ARGS OF CALL

76 *
 77 * ATTRIBUTES-N/A

78 *
 79 * NOTES-

80 * THE SWITCHES USED IN PHASE 16 FOLLOW. IF NON-
 81 * ZERO, THE SWITCH IS TRANSFER=T. IF ZERO, THE
 82 * SWITCH IS NORMAL=N.
 83 * *SW1-T=GENERATE TEMPORARIES AS REQUIRED
 84 * *SW2-T=FAC IN OUTPUT EXPRESSION
 85 * *SW3-T=FAC ON STRING

86 * *****

87 * ABS *DEF CORE*

88 *
 89 * SYSTEM AND FORTRAN EQUATES

90 *
 91 * ~~MENRY EQU~~ *7FFF CORE* MAXIMUM CORE SIZE
 92 * PHSIZ EQU 4*320 MAXIMUM PHASE SIZE
 93 * OVERL EQU MEMRY-PHSIZ PHASES 2-29 START
 94 * FCOM EQU OVERL-22 FORTRAN COMM. TABLE
 95 * PHNTB EQU FCOM-56 PHASE TABLE
 96 * ROLRX EQU PHNTB-50 INTERPHASE CALL

97 *
 98 * FORTRAN COMMUNICATION AREA

99 *
 100 * ORG FCOM
 101 * SOFS BSS 1 START OF STRING
 102 * EOFS BSS 1 END OF STRING
 103 * SOFST BSS 1 START OF SYMBOL TABLE
 104 * SOFNS BSS 1 START OF NON-STATEMENT NUMBERS
 105 * SOFXT BSS 1 START OF SUBSCRIPT TEMPORARIES
 106 * SOFGT BSS 1 START OF GENERATED TEMPORARIES
 107 * EOFST BSS 1 END OF SYMBOL TABLE
 108 * COMON BSS 1 NEXT AVAILABLE COMMON
 109 * CSIZE BSS 1 SIZE OF COMMON
 110 * ERROR BSS 1 OVERLAP ERROR
 111 * FNAME BSS 1 PROGRAM
 112 * BSS 1 NAME
 113 * SORF BSS 1 SUBROUTINE - OR FUNCTION
 114 * CCWD BSS 1 CONTROL CARD WORD
 115 * BIT 15 TRANSFER TRACE
 116 * BIT 14 ARITHMETIC TRACE
 117 * BIT 13 EXTENDED PRECISION
 118 * BIT 12 LIST SYMBOL TABLE
 119 * BIT 11 LIST SUBPROGRAM NAMES

```

120 *          BIT 10 LISA SOURCE PROGRAM
121 *          BIT 9 ONE WORD INTEGERS
122 IOCS  BSS      1      IOCS CONTROL CARD WORD
123 *
124 *          SEE PHASE ONE FOR BIT PATTERNS
125 *
126 DFONT BSS      1      DEFINE FILE COUNT
127 LCOMM BSS      2      SIZE OF INSKEL COMMON
128 ICCER BSS      2      IOCS CONTROL CARD ERROR
129      BSS      2      SYSTEM LOADER USE
130 *          END OF FORTRAN COMM AREA
131 *
132 *
133      ORG      OVERL
134 *
135 START LD        ERROR      CHECK FOR OVERLAP
136      BSC      L  GOSS,Z    GET NEXT PHASE IF OVERLAP
137 *
138 *          INITIALIZE PHASE
139 *
140      LDX      I1 SOFS      XR1 POINTS TO STRING START
141      LD        EOFST      INITIALIZE START OF
142      STO      SOFGT      GENERATED TEMPORARIES
143 *
144 RST  LD        EOFST      RESET GI ENTRIES
145      STO      ROFGT
146 *
147 *          CHECK FOR READ, FIND, WRITE, IF,
148 *          GO TO, ASF, AND ARITH STMENTS
149 *
150 SETUP LD        1 0      STRING ID WORD
151      SRA      11      GET ID TYPE
152      STO      XTMP1      SAVE
153      LDX      3 TYTB-1  XRS=TYPE TABLE SIZE
154      LD        XTMP1      GET ID
155      S        L3 TYTB-1  CHECK IF TABLE TYPE
156      BSC      L  SCANI, - TO SCANI IF ONE TBL TYPES
157      NDJ      3 -1      CONTINUE
158      NDJ      **7      CHECK
159 *
160      LD        XTMP1      GET ID
161      S        ENST      CHECK FOR END STATEMENT
162      BSC      L  NEXTS,Z  GET NEXT STMT IF NOT END
163 *
164 *          EXIT TO PHASE 17
165 *
166 GOSS  BSI      L  ROLRX    XFER TO FORTRAN ROLLER
167      DC        17      NEXT PHASE NUMBER
168 *
169 *          MOVE TO NEXT STATEMENT
170 NEXTS LD        1 0      GET
171      AND      S07FC    STATEMENT
172      SRA      2      WORD
173      STO      * 1      COUNT
174      NDJ      L1 ***    AND MOVE STRING POINTER
175 *
176      LD        XTMP1      IF LAST STATEMENT
177      S        ASF      WAS ASF
178      BSC      L  RST, -  THEN REINITIALIZE FOR SCAN
179      MDX      SETUP    OTHERWISE GO CK NEXT TYPE

```

180	*							
181	*							
182	*							
183	TYTB	DC	/00				ARITHMETIC	
184	ASF	DC	/1A				ARITH STATEMENT FUNCTION	
185		DC	/06				CALL	
186		DC	/0E				GO TO	
187		DC	/0F				IF	
188		DC	/1D				FIND	
189		DC	/11				WRITE	
190	TYTBE	DC	/12				READ	
191	*							
192	*						CONSTANTS	
193	XTMP1	DC	***				TEMPORARY	
194	ENST	DC	/02				END ID	
195	S07FC	DC	/07FC				MASK TO GET STMNT NORM	
196	ROFGT	DC	***					
197	*							
198	*						INITIALIZE FOR SCAN	
199	*							
200	SCANI	STX	L1	IDSV1			SAVE STRING PTR	
201		LD	1	0			ID WORD	
202		BSC		E			SKIP IF NO STMNT NUMBER	
203		MDX	1	1			MOVE STMNT PTR	
204		MDX	1	2			OVER STMNT NO	
205		STX	L1	SNDOP			SAVE STMNT PTR	
206		SLA		16			SET CARRY ON IF STMNT NO	
207		LD	I	IDSV1			GET ID WORD AGAIN	
208		BSC		C			SKIP IF NO STMNT NO	
209		S		TS4				
210		AND		S07FC			GET STATEMENT	
211		SRA		2			WORD COUNT AND	
212		STO		STNOM			SAVE	
213		STO		INORM			SAVE	
214		LDX	L3	PNT			INITIALIZE	
215		STX	3	OUTPT			EXPRESSION OUTPUT POINTER	
216		LDX	L3	PDL			INITIALIZE	
217		STX	3	PUSHD			PUSH DOWN LIST POINTER	
218		SLA		16			NORMALIZE	
219		STO	L	SW3			FACT ON STRING SWITCH	
220	*							
221		LD		ROFGT				
222		STO	L	XOFGT			RESET GIS	
223	*							
224	*						CHECK FOR NORM LESS THAN FOUR	
225	*							
226	CKNM	LDX	L3	***			GET MODIFIED NORM	
227	STNOM	EQU		CKNM 1			DYNAMIC STATEMENT NORM	
228		HDX	3	-3			DECREMENT BY 3	
229		HDX		ADV			GO ADVANCE STMNT PTR IF NO	
230		MDX		PTPOL			DONE. GO RECALCULATE NORM	
231	*							
232	ADV	NDX	1	1			ADVANCE STATEMENT POINTER	
233	*							
234	NANE	LD	1	0			ARE BOTH STRING SYMBOLS	
235		OR	1	-2			OPERATORS	
236		BSC	L	ADV, Z			GO ADV STRING PTR IF NOT	
237	*							
238	FORCE	LD	1	0			RIGHT OPERATOR	
239		STO		ROP 1				

```

240 LD 1 -2 LEFT OPERATOR
241 STO LOP 1
242 LDX L3 FTAB XR3 PTS TO FORCING TABLE
243 ROP LD L3 *** GET ENTRY ROP
244 AND SOOFF RIGHT FORCING VALUE
245 STO XTE1 SAVE RV
246 LOP LD L3 *** GET ENTRY LOP
247 SRA 8 LEFT FORCING VALUE
248 S XTE1 IS RV GI OR = TO LV
249 BSC L ADV,Z- GO ADV STMT PTR IF NOT
250 *
251 MDX I3 LOP 1 LEFT OP FORCING TBL ENTRY
252 LD 3 1 GET INTERPRETER STRING PTR
253 STO * 1
254 LDX L2 *** XR2 PTS TO STRING ADDRESS
255 MDX RETRR 1
256 *
257 * SELECT STRING OF OPERATIONS
258 * TO PERFORM
259 *
260 RETRR MDX 2 2 (NEXT OP PTR IN LIST)
261 LD 2 1 GET SUBSTR ADDR OR ARG
262 BSC I2 0 BR TO SELECTED STRING OPS
263 *
264 * CONSTANTS
265 XTE1 DC 0 TEMPORARY
266 XRR2 EQU XTE1
267 SOOFF DC /O0FF RH SIEVE
268 PUSHDC DC *** STACK POINTER
269 OUTPT DC *** EXPRESSION OUTPUT POINTER
270 ERID DC /A008 ERROR ID
271 FIVE DC 5 WOULD YOU BELIEVE A 5
272 ERNO DC 64 ERROR NUMBER
273 INORM DC *** NORM SAVE
274 TS4 DC 4 AN EVEN NUMBER
275 *
276 *
277 OVCHK LD PUSHDC CHECK FOR OVERLAP
278 S OUTPT IN OUTPUT AREA
279 BSC L RETRN,-Z FINISH UP LIST IF OK
280 *
281 * STATEMENT TOO LONG TO BE SCANNED
282 * PUT ERROR ON STRING
283 *
284 BADST LDX I1 IDSV1 STRING ID POINTER
285 LD 1 0 STATEMENT ID
286 BSC L * 3,E BR IF SIMNT IS NUMBERED
287 LD ERID
288 STO 1 0 PUT ERROR ID ON STRING
289 MDX * 6
290 LD ERID
291 A FIVE ACCOUNT FOR STMT NO
292 STO 1 0 PUT ERROR ID ON STRING
293 MDX 1 1
294 MDX L INORM,1 ADJUST NORM
295 LD ERNO
296 STO 1 1 PUT ERROR NO ON STRING
297 MDX 1 1 XR1 PTS TO NEXT AVAIL LOC
298 *
299 * CLOSE UP THE STRING

```

```

300 LDX I2 IDSV1 STRING ID WORD
301 MDX I2 INORM POINT TO NEXT STATEMENT
302 STX 2 XRR2 SAVE POINTER
303 LD L EOFS SET
304 S XRR2 XR3
305 STO * 1 TO NUMBER
306 LDX L3 *** OF WORDS TO BE MOVED
307 MDX 3 1 THAT DOES IT
308 LD 2 0 LOOP
309 STO 1 1 TO
310 MDX 1 1 CLOSE
311 MDX 2 1 THE
312 MDX 3 -1 STRING
313 MDX *-6 CONTINUE
314 STX L1 EOFS RESET END OF STRING PTR
315 LDX I1 IDSV1 OLD STRING ID AGAIN
316 MDX QLZQ GO PROCESS NEXT STATEMENT
317 *
318 *
319 PTPOL MDX L INORM,-2 CALCULATE STATEMENT NORM
320 *
321 LD CPNT DETERMINE IF
322 S OUTPT STRING MOVEMENT
323 STO CNT IS NECESSARY
324 A INORM
325 STO AMOUT
326 BSC L OPEN, Z OPEN
327 BSC L CLOSE, Z CLOSE
328 *
329 * PLACE POLISH TABLE ON STRING
330 *
331 MOVE LDX L2 PNT XR2 PTS TO OUTPUT EXPRESS
332 LDX I1 SNDOP STRING REPLACE PTR 1
333 LDX L3 *** COUNT
334 CNT EQU *-1
335 LD 2 0 LOOP TO PLACE
336 STO 1 -1 OUTPUT BUFFER
337 MDX 1 1 ON STRING
338 MDX 2 1 REPLACING
339 MDX 3 1 FAC
340 MDX *-6 CONTINUE
341 LDX L1 *** RESET XR1 TO STMT ID
342 IDSV1 EQU *-1 (ID SAVE)
343 *
344 QLZQ BSC L NEXTS GO PROCESS NEXT STATEMENT
345 *
346 AMOUT DC *** AMOUNT TO MOVE
347 CPNT DC PNT ADCON
348 *
349 * OPEN STRING
350 *
351 OPEN LD L EOFST FIRST CHECK
352 S L EOFS TO SEE IF STRING CAN
353 A AMOUT BE OPENED
354 BSC L OVERS, Z IF OVERLAP POSSIBLE GO OUT
355 LDX I1 EOFS XR1 WILL POINT TO OLD EOFS
356 LD L EOFS CALCULATE
357 S SNDOP NUMBER OF
358 STO * 1 WORDS TO BE MOVED
359 LDX L3 *** XR3=NO OF WDS

```

```

360 LD L EOF5 CALCULATE
361 S AMOUT NEW
362 STO L EOF5 EOF5
363 LDX I2 EOF5 XR2 POINTS TO NEW EOF5
364 LD 1 0 LOOP
365 STO 2 0 TO
366 MDX 1 -1 OPEN
367 MDX 2 -1 THE
368 MDX 3 -1 STRING
369 MDX *-6 CONTINUE
370 *
371 * CORRECT NORM
372 *
373 CRTNM LD I IDSV1 STRING ID
374 SRT 2
375 S AMOUT ADJUST BY AMOUT
376 SLT 2
377 STO I IDSV1 RESET
378 MDX MOVE
379 *
380 * CLOSE THE SIRING
381 *
382 CLOSE LDX L2 *** DO SOME
383 SNDOP EQU *-1 ARITHMETIC
384 LD SNDOP AND
385 A AMOUT SO
386 STO * 1 FORTH
387 LDX L1 *** TO SET UP
388 LD L EOF5 COUNIS
389 S *-4 AND
390 STO * 1 POINERS
391 LDX L3 *** FOR CLOSE
392 LD 1 1 LOOP
393 STO 2 1 TO
394 MDX 1 1 CLOSE
395 MDX 2 1 THE
396 MDX 3 -1 STRING
397 MDX *-6 CONTINUE
398 STX L2 EOF5 RESET EOF5
399 MDX CRTNM
400 *
401 *
402 OVERS MDX L ERROR,1 SET OVERLAP ERROR INDICTR
403 BSC L GOSS EXIT TO NEXT PHASE
404 *
405 *
406 *
407 *****
408 *
409 * INTERPRETER
410 *
411 *****
412 *
413 * PLACE OPERAND ON STRING AT XR1-1
414 *
415 PLACE STO 1 -1 STORE ON ON STRING
416 S FACNY
417 BSC L RETRN,Z RETURN IF OPERAND NOT FAC
418 STX L0 SW3 SET FAC ON STRING SWITCH
419 STX L1 FACSV

```

```

420      MDX  L  FACSV,-1  SET STRING ADDRESS OF FAC
421      MDX  RETRN      RETURN
422      *
423  FACNY DC      /8000   FAC
424      *
425      *              MOVE THE STRING POINTER
426      *
427  MVPT  STO      * 1
428      MDX  L1 ***      MOVE PTR BY OPERAND
429      MDX  RETRN      RETURN
430      *
431      *              INSERT STOP CODE ON PDL
432      *
433  STOPI  MDX  L  PUSHD,-1  PUSH DOWN
434      SLA      16      STOP
435      STO  I  PUSHD      ON STACK
436      MDX  2  -1
437      BSC  L  OVCHK      GO CHECK FOR OVERLAP
438      *
439      *              DELETE STOP CODE FROM PDL
440      *
441  STOPD  MDX  L  PUSHD,1  POP UP STACK
442      MDX  2  -1
443      MDX  RETRN      GO PROCESS REST OF STMNT
444      *
445      *              EMPTY PDL TO OUTPUT BUFFER
446      *              UNTIL STOP CODE IS ENCOUNTERED
447      *
448  PTO    MDX  2  -1
449      LD  I  PUSHD      HEAD
450      BSC  L  RETRN, -  RETURN IF STOP OUT
451      STO  I  OUTPT     EMPTY TO EXPRESSION BUFFER
452      MDX  L  PUSHD,1  POP UP STACK
453      MDX  L  OUTPT,1  MOVE OUTPT PTR
454      MDX  PTO 1      CONTINUE
455      *
456      *              DELETE NUMBER OF SYMBOLS
457      *              SPECIFIED BY OPERAND FROM
458      *              THE STRING
459      *
460  DELET  STO      * 4      NUMBER TO DELETE(-OFFSET)
461      STX  1 * 1
462      LDX  L3 ***      GET STRING PTR IN XR3
463      MDX  L3 ***      MODIFY BY NUMBER TO DELETE
464      STX  3 DEND 1
465  DELOP  LD  1 0      LOOP
466      STO  3 0      TO
467      HDX  1 1      CLOSE
468      HDX  3 1      STATEMENT
469      BSC  L  DELOP,2  CONTINUE TO END OF STMNT
470      LD  L  FACSV
471      S    DELET 3  WAS FAC ON STRING MOVED
472      BSC  L  DEND, 2  TO DEND IF NOT
473      LD  L  FACSV
474      A    DELET 5  ADJUST ADDRESS OF
475      STO  L  FACSV  FAC ON STRING
476  DEND  LDX  L1 ***      RESET XR1 TO STRING PTR
477      LD  L  STNOM
478      A    DELOP-2  CORRECT SPECIAL
479      STO  L  STNOM  STRING NORM

```



```

480 *           PLACE STRING ELEMENTS SPECIFIE
481 *
482 *           NDIX      RETRNR      RETURN TO PROCESSING STMT
483 *           BY SUBSTRING ONTO PDL
484 *
485 PUSH  STO          * 1
486     LDX      I3    ***          XR3 = LENGTH OF SUBSTRING
487     STO          PUSH1 1
488     NDIX     L      PUSH1 1,1
489 PUSH1  LD          L      ***          NEXT SUBSTRING ARGUMENT
490     STO          * 1
491     LD          L1    ***          GET DISPLACED STRING ENTRY
492     NDIX     L      PUSHD,-1      PUSH DOWN
493 PUSH2  STO          I      PUSHD      PUT ON STACK
494     S          L      CONS3       WAS IT FAC
495     BSC      L      PUSH3, -      GO TO PUSH3 IF FAC ON STAC
496     NDIX     3      -1          DECREMENT COUNTER
497     NDIX          PUSH1-2        CONTINUE PUSH
498     NDIX          RETRNR         RETURN WHEN DONE
499 PUSH3  LD          L      CONS5       = OPERATOR
500     STO          I      OUTPT        PUT = IN OUTPUT
501     NDIX     L      OUTPT,1        INCREMENT OUTPUT POINTER
502     BSI          COMGT          GET COMPUTED GT
503     STO          I      OUTPT        STORE IN OUTPUT BUFFER
504     NDIX     L      OUTPT,1        INCREMENT OUTPUT POINTER
505     NDIX          PUSH2          GT REPLACES FAC ON STACK
506 *
507 *           COMPUTE GT FOR FAC
508 *
509 CONGT  DC          ***          RETURN ADDRESS
510     SLA          16
511     STO          SW3          NORMALIZE FAC ON STRING SW
512     LD          CONS4        GT ID (/0008)
513     STO          L      ***          STORE GT ID IN SYMBOL TABL
514 XOFGT  EQU          *-1        SYMBOL TABL GT PTR
515     NDIX     L      XOFGT,-3      EXTEND BY 3
516     LD          XOFGT
517     S          L      EOFST
518     BSC      L      * 2,-
519     NDIX     L      EOFST,-3      UPDATE END OF SYMBOL TABLE
520     LD          L      EOFST
521     S          XOFGT          COMPUTE
522     SRT          16          SYMBOL
523     D          CONS2          TABLE
524     OR          CONS3        ENTRY NUMBER
525     BSC      I      COMGT        RETURN TO CALLER
526 *
527 RETRNR BSC      L      RETRR        CENTRAL RETURN BRANCH
528 *           BULK OF INTERPRETER WORK DONE HERE
529 *           OUTFC- OUTPUT GTS AS REQUIRED
530 *           OUT  - DO NOT OUTPUT GTS
531 *
532 OUTFC  LDX          3 1          SET SW 1 TO OUTPUT GTS
533     NDIX     OUT 1
534 OUT    LDX          3 0          DO NOT OUTPUT GTS
535     STX     3 SW1          SET SW1 ON/OFF
536     LDX     3 0          NORMALIZE
537     STX     3 SW2          SW2 = FAC IN OUTPUT
538     STX     3 SWCAL        SWCAL = CALL OP DETECTED
539     STO          * 1

```

540		LDX	I3	***	XR3	SUBSTRING LENGTH
541		STO		OUT1 1		STORE ADDR IN SUBSTR PTR
542		MDX	L	OUT1 1,1		INCREMENT SUBSTR PTR BY 1
543	OUT1	LD	L	***		GET NEXT ITEM ON SUBSTRING
544		STO		* 4		STORE
545		LD		SW1		SET CARRY
546		EOR		OUTFC		ON IF SW1 N
547		SLA		16		OFF IF SW1 T
548	*					SELECT ITEM ON STRING
549		LD	L1	***		DISPLACED FROM STRING PTR
550	*					USUALLY ROP BY SSTR AR
551		STO	I	OUTPT		AND OUTPUT IT
552		BSC	L	NOREV 2,C		BR IF SW1 NOT SET FOR GTS
553		S		CONS3		WAS FAC OUTPUT
554		BSC	L	NOREV 2,Z		BR IF NO
555		STX	0	SW2		YES- SET FAC O/P SW
556		MDX	L	OUTPT,-1		DECREMENT O/P PTR
557		LD	I	OUTPT		GET NEXT LAST O/P WORD
558		BSC	L	NOREV 2,-		BRANCH IF OPERATOR
559		NDX	L	OUTPT,-1		DECREMENT O/P PTR AGAIN
560		LD	I	OUTPT		GET 2ND TO LAST O/P WORD
561		S		CALOP		IS IT CALL OPERATOR
562		BSC		-		SKIP IF NOT
563		STX	0	SWCAL		SET CALL OP SWITCH
564		S		SREV1		IS
565		BSC	L	NOREV, Z		OPERATOR ONE OF
566		S		SREV2		- / **
567		BSC	L	NOREV,-Z		NO- GO TO NOREV
568		A		SREV3		YES- GET REVERSE OP
569		STO	I	OUTPT		AND REPLACE FAC IN O/P
570	NOREV	NDX	L	OUTPT,1		INCREMENT OUTPUT POINTER
571		NDX	L	OUTPT,1		INCREMENT OUTPUT POINTER
572		MDX	3	-1		DECREMENT SUBSTRING COUNT
573		NDX		OUT1-2		GO OUTPUT SOME MORE
574		LD		SW1		DONE WITH SUBSTR
575		BSC	L	OVCHK, -		TO OVCHR IF FAC NOT ON STR
576		LD		SW3		IS FAC ON STRING
577		BSC	L	OVCHK, -		NO- GO CHECK FOR OVERLAP
578		LD		SWCAL		CALL OP DETECTED
579		BSC	L	LOP2, -		BR IF NOT
580		BSI		COMGT		YES- ENTER GT IN SYMBL TBL
581		STO	I	OUTPT		AND PUT GT IN O/P BUFFER
582		MDX	L	OUTPT,1		INCREMENT O/P PTR
583		MDX	L	XOFGT,3		MOVE INTERNAL XOFGT BACK
584		MDX		LOP3		BRANCH
585	LOP2	LD		SW2		FAC IN O/P EXPRESSION
586		BSC	L	OVCHK,Z		YES- GO CHECK FOR OVERLAP
587	LOP3	MDX	L	OUTPT,2		MOVE OUTPUT EXPRESSION
588		LDX	I3	OUTPT		UP TWO PLACES UNTIL AN
589	LOP1	LD	3	-3		OPERATOR IS ENCOUNTERED
590		MDX	3	-1		MOVE OPERATOR AND
591		STO	3	0		LEAVE PTR AT LAST
592		BSC	L	LOP1, Z		MOVE
593		BSI		COMGT		COMPUTE GT AND ENTER IN ST
594		STO	3	-1		STORE GT IN O/P AT PTR=1
595		STO	L	***		SAVE
596	FACSV	EQU		*=1		GT STRING ADDRESS
597		LD		CONS5		OPERATOR
598		STO	3	-2		PUT OP IN O/P AT PTR=2
599		BSC	L	OVCHK		GO CHECK FOR OVERLAP

```

600 *
601 *
602 SW1 DC 0 SOME CONSTANTS
603 SW2 DC 0 GENERATE TEMPORARIES
604 SW3 DC 0 FAC IN OUTPUT EXPRESSION
605 SWCAL DC 0 FAC ON STRING
606 CALOP DC /2E CALL SWITCH
607 SREV1 DC /06-/2E CALL OPERATOR
608 SREV2 DC /0A-/06 SUBTRACT OPERATOR
609 SREV3 DC /0A /44-/06 EXPONENTIATION OPERATOR
610 CONS2 DC 3 CONST TO GET REVOP
611 CONS3 DC /8000 NOT TO BE CONFUSED WITH
612 CONS4 DC /0008 FAC = ALSO ST POINTER BAS
613 CONS5 DC /0E GT ID
614 XR1 DC 0 OPERATOR
615 SEND DC /24 XR1 SAVE
616 * S
617 *
618 *
619 ZEND NDX 1 -1 DECREMENT STRING POINTER
620 STX 1 XR1 AND SAVE
621 LD XR1 CHECK PIR
622 S L SNDOP AGAINST INITIAL
623 BSC L * 2,- LEAVE PTR IF GREATER
624 LDX I1 SNDOP RESET INITIAL STRING PTR
625 BSC L CKNM GO CHECK FOR NORM LT 4
626 *
627 *
628 *
629 SUBSC NDX 2 -8 POINT BACK TO OUTFC
630 LD 1 0 STRING PTR
631 S SEND SUBSCRIPT RT PARENTHESIS
632 BSC L RETRR,Z RETURN IF NOT
633 NDX 2 7 READY FOR MVPT
634 MDX RETRN RETURN
635 *
636 *
637 PNT BSS 401 *****
638 * POLISH OUTPUT *
639 * | *
640 * | *
641 * | *
642 * | *
643 * | *
644 * | *
645 * | *
646 * | *
647 * | *
648 * | *
649 * | *
650 PDL DC 0 * PUSH DOWN LIST *
651 * *****
652 *
653 *
654 *
655 FTAB DC /3C3C SEMI-COLON 00 TERMINATOR
656 DC BADST
657 DC /0132 02 RIGHT PARENTHESIS
658 DC BADST
659 DC /0A0A 04 ADD

```

660	DC	CD1	-	06	SUBTRACT
661	DC	/0A0A			
662	DC	CD1	/	08	DIVIDE
663	DC	/0505			
664	DC	CD1	**	0A	EXPONENTIATE
665	DC	/0504			
666	DC	CD1	*	0C	MULTIPLY
667	DC	/0505			
668	DC	CD1		0E	ASSIGN
669	DC	/3B3C			
670	DC	CD1		10	LEFT PARENTHESIS
671	DC	/3101			
672	DC	CD2	,	12	COMMA
673	DC	/3130			
674	DC	CD3			
675	DC	/3101	IF	14	IF
676	DC	CD8			
677	DC	/3101	GOT	16	GO TO
678	DC	CD5			
679	DC	/2A01	0	18	LITERAL SUBSCRIPT
80	DC	CD6			
681	DC	/2A01	1	1A	1-DIM SUBSCRIPT
682	DC	CD6			
683	DC	/2A01	2	1C	2-DIM SUBSCRIPT
684	DC	CD6			
685	DC	/2A01	3	1E	3-DIM SUBSCRIPT
686	DC	CD6			
687	DC	/0501	U-	20	UNARY MINUS
688	DC	CD7			
689	DC	/3131	RG	22	IMPLIED DO END
690	DC	CD10			
691	DC	/3131	S	24	SUBSCRIPT RT PARE
692	DC	BADST			
693	DC	/3001	I/O	26	INPUT/OUTPUT
694	DC	CD11			
695	DC	/3030	IO	28	LIST
696	DC	CD12			
697	DC	/3101	DOA	2A	IMPLIED DO
698	DC	CD9			
99	DC	/3B01	ASF	2C	ARITH STMT FUNC
700	DC	CD13			
701	DC	/0201	C	2E	CALL
702	DC	CD4			
703	DC	/3001	DIO	30	DISK I/O OPERATOR
704	DC	CD11			
705	*				
706	*		CODE1	- / ** *	
707	CD1	DC	OUTFC		OUTPUT STRING SYMBOLS UNDE
708	DC	SUB1			CONTROL OF SUBSTRING 1
709	DC	DELET			DELETE
710	DC	-2			TWO SYMBOLS FROM STRING
711	DC	PLACE			PLACE
712	DC	/8000			FAC ON STRING AT PTR-1
713	DC	ZEND			END CODE1
714	*				
715	*		CODE2		
716	CD2	DC	NVPT		MOVE STRING POINTER
717	DC	-1			BACK ONE
718	DC	DELET			DELETE SYMBOL ON STRING AT
719	DC	-1			POINTER = 1

```

720      DC      MVPT      MOVE STRING POINTER
721      DC      2          UP 2
722      DC      DELET     DELETE SYMBOL ON STRING AT
723      DC      -1        POINTER # 1
724      DC      ZEND      END CODE2
725      *
726      *              CODE3
727      CD3     DC      PUSH      PUSH DOWN STACK UNDER
728      DC      SUB2      CONTROL OF SUBSTRING 2
729      DC      DELET     DELETE SYMBOL ON STRING AT
730      DC      -2        POINTER #2 AND PTR-1
731      DC      ZEND      END CODE3
732      *
733      *              CODE4 C
734      CD4     DC      OUTFC     OUTPUT STRING SYMBOLS UNDE
735      DC      SUB1      CONTROL OF SUBSTRING 1
736      DC      PTO       STACK TO O/P UNTIL STOP CD
737      DC      DELET     DELETE SYMBOL ON STRING AT
738      DC      -2        POINTER #1 AND PTR-2
739      DC      PLACE     PLACE
740      DC      /8000     FAC ON STRING AT PTR-1
741      DC      ZEND      END CODE4
742      *
743      *              CODE5 GOT
744      CD5     DC      OUT      OUTPUT STRING SYMBOLS UNDE
745      DC      SUB2      CONTROL OF SUBSTRING 2
746      DC      PTO       STACK TO O/P UNTIL STOP CD
747      DC      DELET     DELETE SYMBOL ON STRING AT
748      DC      -1        POINTER #1
749      DC      ZEND      END CODE5
750      *
751      *              CODE6 0 1 2 3
752      CD6     DC      OUT      OUTPUT STRING SYMBOLS UNDE
753      DC      SUB3      CONTROL OF SUBSTRING 3
754      DC      DELET     DELETE SYMBOL ON STRING AT
755      DC      -2        POINTER #2 AND PTR-1
756      DC      MVPT     MOVE STRING POINTER
757      DC      2          UP TWO
758      DC      OUT      OUTPUT STRING SYMBOLS UNDE
759      DC      SUB2      CONTROL OF SUBSTRING 2
760      DC      SUBSC     OUTPUT SUBSCRIPT ELEMENTS
761      DC      MVPT     MOVE STRING POINTER
762      DC      1          UP ONE
763      DC      DELET     DELETE SYMBOL ON STRING AT
764      DC      -3        POINTER-3, PTR-2, PTR=1
765      DC      ZEND      END CODE6
766      *
767      *              CODE7 U-
768      CD7     DC      OUTFC     OUTPUT STRING SYMBOLS UNDE
769      DC      SUB3      CONTROL OF SUBSTRING 3
770      DC      DELET     DELETE SYMBOL ON STRING AT
771      DC      -1        POINTER-1
772      DC      PLACE     PLACE
773      DC      /8000     FAC ON STRING AT PTR-1
774      DC      ZEND      END CODE7
775      *
776      *              CODE8 IF
777      CD8     DC      OUTFC     OUTPUT STRING SYMBOLS UNDE
778      DC      SUB3      CONTROL OF SUBSTRING 3
779      DC      PTO       STACK TO O/P UNTIL STOP CD

```

```

780      DC      DELET  DELETE SYMBOL ON STRING AT
781      DC      -1     POINTER = 1
782      DC      ZEND   END CODE8
783      *
784      *           CODE9 DOA
785      CD9     DC      OUT    OUTPUT STRING SYMBOLS UNDE
786      DC      SUB1    CONTROL OF SUBSTRING 1
787      DC      PUSH   PUSH DOWN STACK UNDER
788      DC      SUB5    CONTROL OF SUBSTRING 5
789      DC      MVPT   MOVE STRING POINTER
790      DC      1      UP ONE
791      DC      DELET  DELETE SYMBOLS ON STRING A
792      DC      -4     P-4, P-3, P-2, P-1
793      DC      STOPI  PUSH DOWN STOP CODE
794      DC      ZEND   END CODE9
795      *
796      *           CODE10 RG
797      CD10    DC      PLACE  PLACE IMPLIED DO TEST
798      DC      /4A    DOT ON STRING AT PTR-1
799      DC      OUT    OUTPUT STRING SYMBOLS UNDE
800      DC      SUB2    CONTROL OF SUBSTRING TWO
801      DC      STOPD  DELETE STOP CODE FROM STAC
802      DC      PTO    STACK TO Q/P UNTIL STOP CD
803      DC      MVPT   MOVE STRING POINTER
804      DC      1      UP ONE
805      DC      DELET  DELETE SYMBOLS ON STRING A
806      DC      -3     PTR-3, PTR-2, PTR-1
807      DC      ZEND   END CODE10
808      *
809      *           CODE11 I/O DIO
810      CD11    DC      OUT    OUTPUT STRING SYMBOLS UNDE
811      DC      SUB3    CONTROL OF SUBSTRING 3
812      DC      PTO    STACK TO Q/P UNTIL STOP CD
813      DC      DELET  DELETE SYMBOLS ON STRING A
814      DC      -2     POINTER =2, PTR-1
815      DC      MVPT   MOVE STRING POINTER
816      DC      1      UP ONE
817      DC      PLACE  PLACE I/O LIST OP ON STRIN
818      DC      /28    AT POINTER = 1
819      DC      ZEND   END CODE11
820      *
821      *           CODE12 LIST
822      CD12    DC      OUT    OUTPUT STRING SYMBOLS UNDE
823      DC      SUB3    CONTROL OF SUBSTRING 3
824      DC      MVPT   MOVE STRING POINTER
825      DC      1      UP ONE
826      DC      DELET  DELETE SYMBOLS ON STRING A
827      DC      -2     POINTER =2, PTR-1
828      DC      ZEND   END CODE12
829      *
830      *           CODE13 ASF
831      CD13    DC      OUT    OUTPUT STRING SYMBOLS UNDE
832      DC      SUB1    CONTROL OF SUBSTRING 1
833      DC      PTO    STACK TO Q/P UNTIL STOP CD
834      DC      DELET  DELETE SYMBOLS ON STRING A
835      DC      -2     POINTER =2, PTR-1
836      DC      MVPT   MOVE STRING POINTER
837      DC      1      UP ONE
838      DC      ZEND   END CODE13
839      *

```

840	*				SUBSTRING 1
841	SUB1	DC	3		LENGTH
842		DC	-2		P-2
843		DC	-3		P-3
844		DC	-1		P-1
845	*				
846	*				SUBSTRING 2
847	SUB2	DC	1		LENGTH
848		DC	-1		P-1
849	*				
850	*				SUBSTRING 3
851	SUB3	DC	2		LENGTH
852		DC	-2		P-2
853		DC	-1		P-1
854	*				
855	*				SUBSTRING 5
856	SUB5	DC	1		LENGTH
857		DC	-3		P-3
858	*				
859		BSS		OVERL-***320*4	PHASE-16 PATCH AREA
860		END		START	