

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
/COPYRIGHT (C) 1975

XVIM

/DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

/THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY
/ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
/THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
/SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PRO-
/VIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
/EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO
/THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE
/SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

/THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
/WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COM-
/MITMENT BY DIGITAL EQUIPMENT CORPORATION.

/DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
/OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
.EJECT

```

21 /
22 /
23 / ASSEMBLY PARAMETER ARBVEC, NONX/M
24 /
25 / WHEN HARDWARE STROKE VECTOR IS AVAILABLE ON THE VT-15
26 / DEFINE ARBVEC=0 AT ASSEMBLY TIME TO SAVE SPACE
27 /
28 / TO RUN ON A NON-X/M SYSTEM IT IS NECESSARY
29 / TO DEFINE NONX/M=0 AS ASSEMBLY TIME
30 /
31 /
32 /SYMBOLS USED TO FORM INSTRUCTIONS.
33 /
34 140000 A PY=140000 /POSITION, Y DIRECTION
35 144000 A PX=144000 /POSITION, X DIRECTION
36 100000 A STRK=100000 /STROKE VECTOR, X OR Y DIRECTION
37 600000 A DJMP=600000 /DISPLAY JUMP
38 620000 A DJMPI=620000 /DISPLAY JUMP INDIRECT
39 660000 A DJMSI=660000 /DISPLAY JMS*
40 402000 A V1=402000 /BASIC VECTOR, DIRECTION 1
41 410000 A V4=410000 /BASIC VECTOR, DIRECTION 4
42 414000 A V6=414000 /BASIC VECTOR, DIRECTION 6
43 060000 A CHARSI=060000 /CHARACTER STRING INSTRUCTION
44 000020 A SC=20 /SCALE ENABLE
45 002000 A INT=2000 /INTENSITY ENABLE
46 000010 A LP=10 /LIGHT PEN ENABLE
47 001000 A BK=1000 /BLINK ENABLE
48 000004 A DSH=4 /DASH ENABLE
49 000002 A OFST=2 /OFFSET ENABLE
50 000040 A ROT=40 /ROTATE ENABLE
51 000400 A LODN=400 /LOAD NAME REGISTER ENABLE
52 200000 A PRM1=200000 /PARAMETER WORD 1 OP CODE
53 200000 A DNOP=200000 /DISPLAY NOP
54 200001 A DNOPB=200001 /BEGINNING OF PARAM GROUP MARKER
55 200004 A DNOPE=200004 /END OF GROUP MARKER
56 210000 A PRM2=210000 /PARAMETER WORD 2 OP CODE
57 220000 A PRM3=220000 /PARAMETER WORD 3 OP CODE
58 234000 A SKP2=234000 /SKIP2 OP CODE
59 235000 A DSKP=235000 /UNCONDITIONAL SKIP
60 240000 A SAVE=240000 /SAVE INSTRUCTION
61 260000 A REST=260000 /RESTORE INSTRUCTION
62 162000 A GY=162000 /GRAPHPLOT, Y DIRECTION
63 703142 A RS3=703142 /READ STATUS 3 IOT
64 000010 A DATVT=10 /VT15 USES DAT SLOT 10
65 703044 A STPD=703044 /STOP DISPLAY IOT
66 703004 A LSD=703004 /START DISPLAY
67 000013 A ENTRES=13 /NUMBER OF TABLE ENTRIES
68 /
69 /
70 .GLOBL LINE,TEXT,COPY,GRAPH,PRAMTR,BLANK,UNBLNK
71 .GLOBL DINIT,SETPT,PLOT,REPLOT,RSETPT,DCLOSE,DELETE
72 /

```

```

73      / INFO ENTRIES FOR OTHER GRAPHICS MODULES
74      /
75      / .GLOBAL MF.101,MF.102,LNSIZE
76      / .IODEV DATVT
77      /
78      /
79      / TABLE ORDER
80      /
81      / 0 COPY
82      / 1 LINE
83      / 2 PARAMETER
84      / 3 TEXT
85      / 4 SETPT
86      / 5 GRAPH
87      / 6 DELETE
88      / 7 DINIT
89      / 8 BLANK
90      / 9 UNBLNK
91      / 10 LNSIZE
92      /
93      /
94      / OPTIONAL ARGUMENT FETCH TABLE
95      /
96      / FOR SETTING UP PNAME OR CNAME AS REQUIRED
97      /
98      / THIS TABLE IS XCT'ED
99      /
100      00000 R 201470 R OPT LAC ARG3 /COPY
101      00001 R 201471 R LAC ARG4 /LINE
102      00002 R 100430 R JMS OPTPAR /VARIES WITH # OF BITS
103      00003 R 201470 R LAC ARG3 /TEXT
104      00004 R 201470 R LAC ARG3 /POINT
105      00005 R 201471 R LAC ARG4 /GRAPH
106      00006 R 201466 R LAC ARG1 /FOR DELETE FETCH CNAME
107      00007 R 750000 A CLA /NONE FOR DINIT
108      00010 R 750000 A CLA /NONE FOR BLANK
109      00011 R 750000 A CLA /OR UNBLNK
110      00012 R 750000 A CLA /FOR LNSIZE
111      /
112      /
113      /
114      / CODE GENERATION TABLE
115      /
116      / AT VERY END, TO GENERATE CODE
117      /
118      / XCT WITH AC=-1, LINK=1
119      /
120      00013 R 100755 R CODE JMS CPY
121      00014 R 101137 R JMS VECTOR
122      00015 R 100606 R JMS PRM
123      00016 R 100522 R JMS CHR
124      00017 R 101063 R JMS PNT

```

```

125      00020 R 601010 R      JMP      GRPH      /SUBPIC ONLY NEED NOT JMS
126      00021 R 740000 A      LNOP     NOP        /DELETE GENERATES NO CODE
127      00022 R 601075 R      JMP      DNT       /DINIT DOESN'T ADD CODE TO MAIN
128      00023 R 601030 R      JMP      BLN
129      00024 R 601031 R      JMP      UBL
130      00025 R 601127 R      JMP      LNS
131      /
132      /   NEEDS TABLE
133      /
134      /   TO OBTAIN NEEDED * OF LOC'S FOR A REPLOT
135      /
136      /   EXECUTE WITH AC=0, LINK=0
137      /
138      00026 R 600445 R      NEEDS    JMP      NEDCOP  /3 OR 6 DEPENDING
139      00027 R 101137 R      JMS      VECTOR  /1 OR 2 OR MANY
140      00030 R 600451 R      JMP      NEDPAR  /1-6
141      00031 R 200044 R      LAC      L3       /TEXT IS ALWAYS 3
142      00032 R 200052 R      LAC      L2       /RSETPT IS ALWAYS 2
143      00033 R 000743 R      LTEMP2   TEMP2    /CAN'T REPLOT GRAPH
144      00034 R 750000 A      CLA        /DELETE NEEDS 0
145      /
146      /   LITERALS AND LOCATIONS CORRESPONDING
147      /
148      00035 R 000013 A      LENT      ENTRES    /NUMBER OF ENTRIES
149      00036 R 000000 R      LOPT      OPT       /ADDR OF EXECUTE TABLE
150      /
151      .EJECT

```

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

/
/ **POINT**
/
/ ORDER IRRELEVANT
/
/ THE FORMAT OF THE DATA WORD FOLLOWING THE CALL TO GETARG
/
/ THE LOW TEN BITS OF THE WORD IS A GRAPHICS CODE DESCRIBING
/ WHICH ACTION ROUTINES TO CALL
/
/ BITS THREE THRU SEVEN MUST BE ZERO
/
/ BITS 0-2 ARE AS FOLLOWS
/
/ 0 (OCTAL) SUBPICTURE CALL
/ 1 MAIN FILE CALL
/ 2 EDIT CALL (CODE MODIFICATION DELETE, RSETPT)
/ 3 ILLEGAL
/ 4 ILLEGAL
/ 5 MAIN FILE CALL, BUT MOVE ARGUMENTS BY ONE (PLOT)
/ IN THIS CASE LOW TEN BITS ARE A NOP
/ REAL DATA COMING FROM THE FIRST ARGUMENT IN CALL
/ 6 EDIT CALL, MOVE ARGUMENTS (REPLT), HERE
/ ALSO, THE LOW TEN BITS ARE A NOP
/ 7 ILLEGAL
/
/

00037 R 000000 A	LINE	0	
00040 R 100116 R		JMS	GETARG
00041 R 000001 A	L1	1	
00042 R 000000 A	TEXT	0	/ALSO USE THESE AS LITERALS
00043 R 100116 R		JMS	GETARG
00044 R 000003 A	L3	3	
00045 R 000000 A	COPY	0	
00046 R 100116 R		JMS	GETARG
00047 R 000000 A	L0	0	
00050 R 000000 A	PRAMTR	0	
00051 R 100116 R		JMS	GETARG
00052 R 000002 A	L2	2	
00053 R 000000 A	GRAPH	0	
00054 R 100116 R		JMS	GETARG
00055 R 000005 A	L5	5	
00056 R 000000 A	SETPT	0	
00057 R 100116 R		JMS	GETARG
00060 R 100004 A		100004	/MAIN FILE, AND POINT
00061 R 000000 A	RSETPT	0	
00062 R 100116 R		JMS	GETARG
00063 R 200004 A		200004	/EDIT, AND POINT
00064 R 000000 A	DELETE	0	
00065 R 100116 R		JMS	GETARG
00066 R 200006 A		200006	/DELETE GENERATES A NULL EDIT

204	00067 R 000000 A	PLOT	0		
205	00070 R 100116 R		JMS	GETARG	
206	00071 R 500000 A		500000		/MAIN. BUT SHIFT ARGUMENTS
207	00072 R 000000 A	REPLOT	0		
208	00073 R 100116 R		JMS	GETARG	
209	00074 R 600000 A		600000		/EDIT. BUT SHIFT ARGUMENTS
210	00075 R 000000 A	BLANK	0		
211	00076 R 100116 R		JMS	GETARG	
212	00077 R 000010 A	L10	10		
213	00100 R 000000 A	UNBLNK	0		
214	00101 R 100116 R		JMS	GETARG	
215	00102 R 000011 A	L11	11		
216	00103 R 000000 A	DINIT	0		
217	00104 R 100116 R		JMS	GETARG	
218	00105 R 000007 A	L7	7		/DINIT IS A SUB, SINCE NO CODE ADDED
219		/			/THRU MAIN FILE POINTERS
220	00106 R 000000 A	LNSIZE	0		
221	00107 R 100116 R		JMS	GETARG	
222	00110 R 000012 A	L12	12		
223		/			
224	00111 R 000000 A	DCLOSE	0		
225	00112 R 141463 R		DZM	MFPTR	/KILL RUN SWITCH
226			.CLOSE	DAT/T	/DCLOSE SO SMALL HAS OWN CODE
	00113 R 000010 A *G		CAL	DAT/T&777	
	00114 R 000006 A *G		6		
227	00115 R 620111 R		JMP*	DCLOSE	
228		/			
229			.EJECT		

```

230 /
231 / ROUTINE GETARG
232 /
233 / THIS IS THE MAIN PROCESSING CODE
234 /
235 / OBTAIN ARGUMENTS, SET UP TABLE XCT'S, ESTABLISH
236 / SUB. MAIN, REPLOT ETC.
237 /
238 / CALLING SEQUENCE
239 /
240 / LINE 0
241 / JMS GETARG /MUST BE LOC FOLLOWING ENTRY POINT
242 / CONST /CONST FORM IS DESCRIBED ABOVE
243 /
244 / CONTROL NEVER RETURNS FOLLOWING THE JMS
245 /
246 00116 R 000000 A GETARG 0
247 00117 R 777776 A LAW -2 /SET UP TO ESTABLISH EXIT POINT
248 00120 R 040111 R DAC TYPE /SET UP LOC TO REMEMBER MAIN-SUB SPLIT
249 00121 R 340116 R TAD GETARG /POINTS TO ENTRY POINT
250 00122 R 040745 R DAC TEMP4 /FOR EXTRA INDIRECT
251 00123 R 220745 R LAC* TEMP4 /
252 00124 R 040050 R DAC RETAD2 /RETURN ADDR FOR EXIT
253 00125 R 040745 R DAC TEMP4 /FETCH ARGUMENT POINTER
254 00126 R 740001 A CMA
255 00127 R 360745 R TAD* TEMP4
256 00130 R 501512 R AND (7777 /IN CASE PAGE MODE
257 00131 R 740001 A CMA /-COUNT-1
258 00132 R 040744 R DAC TEMP3 /CONTROL FOR FETCH LOOP
259 /
260 00133 R 220116 R LAC* GETARG /GET DESCRIPTOR WORD
261 00134 R 751100 A SPA!CLA /SKIP IF ORDINARY ARG. FETCH, LEAVING AC=0
262 00135 R 777777 A LM1 LAW -1 /PLOT OR REPLOT, MOVE BACK ARG'S
263 00136 R 341464 R TAD LARG1 /ADDRESS OF ARG1 STORAGE
264 00137 R 040742 R DAC TEMP1 /FOR ARGUMENT PLACEMENT
265 00140 R 600150 R JMP GTRG05 /JOIN FETCH LOOP
266 /
267 00141 R 220745 R GTRG03 LAC* TEMP4 /AN ARGUMENT
268 00142 R 740100 A SMA /SKIP IF EXTRA INDIRECT
269 00143 R 600146 R JMP GTRG04 /NO
270 00144 R 040743 R DAC TEMP2
271 00145 R 220743 R LAC* TEMP2
272 00146 R 060742 R GTRG04 DAC* TEMP1 /TO OUR CORE
273 00147 R 440742 R ISZ TEMP1 /MOVE PUT
274 00150 R 440745 R GTRG05 ISZ TEMP4 /MOVE FETCH
275 00151 R 440744 R ISZ TEMP3 /MORE
276 00152 R 600141 R JMP GTRG03 /YUP
277 00153 R 160742 R DZM* TEMP1 /ZERO ANY POSSIBLE OPTIONAL ARGUMENT!
278 00154 R 220116 R LAC* GETARG /FETCH GRAPHICS CODE (IN LOW 10 BITS
279 00155 R 741100 A SPA /SKIP IF WE REALLY HAVE THE CODE
280 00156 R 221465 R LAC* ARG0 /WE DON'T, PLOT OR REPLOT GAVE IT TO US, GET IT
281 00157 R 501513 R AND (1777 /STRIP HIGH STUFF FROM GETARG POINTED ARG.

```

282	00160 R 041465 R	DAC	ARG0	/SAVE GRAPHICS CODE FOR LATER CHECK
283	00161 R 340036 R	TAD	LOPT	/MAKE UP ADDR FOR XCT* OF TABLE
284	00162 R 040106 R	DAC	XOPT	/XCT OPTIONAL ARGUMENT TABLE
285	00163 R 340035 R	TAD	LENT	/
286	00164 R 040103 R	DAC	XCODE	/XCT CODE GENERATE TABLE
287	00165 R 340035 R	TAD	LENT	/
288	00166 R 040045 R	DAC	XNEED	/XCT NEED TABLE IN CASE EDIT
289				
290	00167 R 220116 R	LAC*	GETARG	/SPLIT ON SUB, MAIN, EDIT
291	00170 R 742010 A	RTL		/EDIT BIT INTO LINK
292	00171 R 745400 A	SZL!CLL		/SKIP IF NOT EDIT, ENTER ALL WITH LINK=0
293	00172 R 600265 R	JMP	EDIT	/GO TO EDIT PROCESSING (DON'T REJOIN)
294	00173 R 750100 A	SMA!CLA		/SKIP ON MAIN FILE
295	00174 R 600253 R	JMP	SUBPC	/SUBPICTURE PROCESSING, RETURN AT JOINSM
296				
297				

.EJECT

```

298
299 00175 R 541463 R MAINIT SAD MFPT /ARE WE RUNNING
300 00176 R 440111 R ISZ TYPE /NO, SO DON'T PLACE MFPT
301 00177 R 201462 R LAC MAINFL /PRESENT MAINFILE IS
302 00200 R 040042 R DAC FILE /THE ACTION FILE
303 00201 R 420106 R XCT* XOPT /FETCH OPTIONAL ARG (CNAME)
304 00202 R 740200 A SZL /SKIP IF NO CNAME
305 00203 R 440746 R ISZ PARBAK /SET UP FOR FOLLOWING DNOP IN CASE PRAM
306 00204 R 741200 A SNA /SKIP IF CNAME TO MAKE FILLING POINTER
307 00205 R 200033 R JOINSM LAC LTEMP2 /DISABLE CNAME RETURN
308 00206 R 040745 R DAC TEMP4 /POINTER TO CNAME OR TEMP2
309 00207 R 220042 R LAC* FILE /LENGTH POINTER FROM FILE
310 00210 R 745202 A SNA!CLL!CML /SKIP IF OLD FILE, LEAVING LINK1=OLD
311 00211 R 742010 A RTL /NEW FILE, LENGTH=2, LINK0=NEW
312 00212 R 340042 R TAD FILE /+FILE ADDR = SCENE OF ACTION
313 00213 R 040067 R DAC POINT /PLACEMENT POINTER
314 00214 R 040072 R DAC POINTS /STATIONARY POINTER
315 00215 R 060745 R DAC* TEMP4 /TO CNAME IF PRESENT, IF NOT, TEMP2
316 00216 R 340135 R TAD LM1 /MAKE A COPY AT -1. FLIP LINK!!
317 00217 R 040352 R DAC POINT1
318 00220 R 745402 A SZL!CLL!CML /SKIP IF OLD FILE, LINK NEEDED PRESENTLY
319 00221 R 600362 R JMP NEWFIL /GO SETUP NEW FILE; AC=FILE+1
320 00222 R 220352 R LAC* POINT1 /CHECK OUT RECLAIM OF DNOPE
321 00223 R 241507 R XOR LDNOPE /MAKE AC=0 IF A DNOPE ENDS THIS FILE
322 00224 R 743210 A SNA!RTL /IF NOT, SKIP, THEN TO MSTORE5, NO BACK-UP
323
324 00225 R 541465 R SAD ARG0 /DNOPE, NOW CHECK IF WE ARE A PARAMETER
325 00226 R 600234 R JMP MSTORE5 /NO BACK-UP
326 00227 R 220067 R LAC* POINT /NOT A PARAMETER, BACK UP END OF FILE
327 00230 R 060352 R DAC* POINT1 /JMP* OVER TOP OF DNOPE
328 00231 R 200352 R LAC POINT1 /NOW BACK UP THE POINTERS
329 00232 R 040067 R DAC POINT
330 00233 R 040072 R DAC POINTS
331 00234 R 754003 A MSTORE5 CLA!CMA!CLL!CML /REAL CODE GENERATE, CALL AC=-1, LIN=1
332 00235 R 040056 R DAC PUTCNT /SET UP PUT TO SAVE FIRST INSTR.
333 00236 R 420103 R XCT* XCODE /GENERATE CODE THRU POINT
334 00237 R 220072 R MRETUR LAC* POINTS /PREVIOUS END OF FILE
335 00240 R 100352 R JMS PUT /REGULAR PLACEMENT MECHANISM TO PRESENT END
336 00241 R 200037 R LAC HOLD /PLACE FIRST INST OVER OLD EOF
337 00242 R 060072 R DAC* POINTS
338 00243 R 200067 R LAC POINT /ADDR OF NEW END OF FILE
339 00244 R 440111 R SHUNT ISZ TYPE /SKIP UNLESS MAIN & RUNNING
340 00245 R 041463 R DAC MFPT /MAIN AND RUNNING, UPDATE POINTER
341 00246 R 740001 A CMA /COMPUTE TOP-OF-FILE LENGTH POINTER
342 00247 R 340042 R TAD FILE /ADDR OF TOP
343 00250 R 740001 A CMA
344 00251 R 060042 R DAC* FILE /LEGTH POINTR IS BOTT ADDR-TOP ADDR
345 00252 R 600350 R JMP EXX2 /OK -1 TO AC AND RETURN
346
347 / SUBPICTURE SETUP
348
349 00253 R 420106 R SUBPC XCT* XOPT /FETCH OPTIONAL ARG. IF ANY

```

350	00254	R	741200	A
351	00255	R	201502	R
352	00256	R	041502	R
353	00257	R	040042	R
354	00260	R	241462	R
355	00261	R	741200	A
356	00262	R	541463	R
357	00263	R	440111	R
358				
359	00264	R	600205	R
360				
361				

```

SNA          /SKIP ON REAL ONE
LAC          PNAME  /WASN'T, GET LAST PNAME
DAC          PNAME
DAC          FILE   /FILE USING, FOR MAIN AND SUB
XOR          MAINFL /IS SUBPIC SAME AS MAIN?!?
SNA          /SKIP IF NOT, DON'T CHANGE MFPTR.
SAD          MFPTR  /=, NOW SKIP IF MAINFILE RUNNING
ISZ          TYPE   /NASTY, MAKE TYPE=-1 SO IT WILL SKIP AT 'SHUNT'
JMP          JOINSM /MISS THE ISZ ONLY IF SUB=MAIN, AND RUNNING.

.EJECT

```

```

362
363
364
365
366 00265 R 420106 R EDIT XCT* XOPT /FETCH 'OPTIONAL' ARGUMENT
367 / /WHICH IS CNAME
368 00266 R 040742 R DAC TEMP1 /BUT WE WANT CONTENTS, NOT ADDR LIKE M+S
369 00267 R 220742 R LAC* TEMP1 /CNAME ADDR
370 00270 R 040352 R DAC POINT1 /ONCE FOR POINTER TO DNOP GROUP
371 00271 R 040061 R DAC POINT2 /ONCE FOR COUNT OF GROUP
372 00272 R 040072 R DAC POINTS /ONCE FOR CODE MAKERS
373 00273 R 340135 R TAD LM1 /BACK UP ONE FOR CODE PLACEMENT POINTER
374 00274 R 040067 R DAC POINT
375
376 / PART OF CODE TO DETERMINE SIZE OF PREVIOUS GROUP
377 /
378 00275 R 220072 R LAC* POINTS /FIRST DISPLAY INSTR. OF GROUP
379 00276 R 541511 R SAD LDSKP /CHECK SPECIAL CASE OF SKP. COUNT
380 00277 R 600366 R JMP SIZAPP /YES, OBTAIN COUNT FROM NEXT WORD
381 00300 R 540750 R SAD LDNOP /GROUP TO BE DELETED?
382 00301 R 744010 A CLL!RAL /NASTY, CALL IT A BASIC VECTOR
383 00302 R 500021 R AND LNOP /RETAIN 4 BITS OF OP (LNOP=740000)
384 00303 R 540750 R SAD LDNOP /IS IT A PARAMETER INSTR
385 00304 R 600370 R JMP SIZPAR /YES, SPECIAL CASE
386 00305 R 744010 A RAL!CLL /MOVE FOUR BITS TO RIGHT OF AC
387 00306 R 742010 A RTL
388 00307 R 742010 A RTL
389 00310 R 341514 R TAD (SIZETB /ADDR OF TABLE OF SIZES BY OP CODE
390 00311 R 040061 R DAC POINT2 /INIT VALUE OF POINT2 NOT NOW NEEDED
391 00312 R 220061 R SIZTST LAC* POINT2 /SIZE TO AC
392 00313 R 741200 A SNA /0 SIZE IS ILLEGAL, EXIT TO USER
393 00314 R 620050 R JMP* RETAD2
394 00315 R 340067 R TAD POINT /POINT TO LAST LOC IN GROUP
395 00316 R 040061 R SIZJ1 DAC POINT2 /SIZPAR REJOINS HERE
396 00317 R 740001 A CMA /MAKE COUNTER FOR DNOP OF GROUP
397 00320 R 340072 R TAD POINTS
398 00321 R 040056 R DAC PUTCNT /-# OF DNOP'S TO PLACE
399 00322 R 200750 R LAC LDNOP /NOW COUNT DNOP'S BEYOND GROUP
400
401 / FIND COUNT OF DNOP'S BEYOND GROUP
402 /
403 00323 R 440061 R SIZLOO ISZ POINT2 /MOVE POINTER TO CHECK NEXT LOC
404 00324 R 560061 R SAD* POINT2 /ANOTHER DNOP?
405 00325 R 600323 R JMP SIZLOO /YES
406
407 / POINT2 NOW POINTS 1 BEYOND TOTAL SPACE; FIND OUT NEED
408 /
409 00326 R 754000 A CLL!CLA /NEED ROUTINES ASSUME AC.LINK=0
410 00327 R 420045 R XCT* XNEED /RETURN # NEEDED IN AC
411 00330 R 340067 R TAD POINT /LAST ADDR NEEDED
412 00331 R 740001 A CMA /-LAST-1 NEEDED
413 00332 R 340061 R TAD POINT2 /COMPARE

```

```

414 00333 R 751100 A SPA!CLA /SKIP IF ENUF
415 00334 R 620050 R JMP* RETAD2 /NO, EXIT WITH AC=0
416 00335 R 703044 A STPD /STOP DISPLAY FOR EDIT
417 00336 R 200750 R LAC LDNOP /CLEAR OUT PREVIOUS GROUP
418 00337 R 060352 R EDICLR DAC* POINT1 /
419 00340 R 440352 R ISZ POINT1
420 00341 R 440056 R ISZ PUTCNT /DONE
421 00342 R 600337 R JMP EDICLR /NO
422 /
423 / FALL THRU LEAVING PUTCNT=0, WHICH IS CORRECT FOR PUT!
424 /
425 00343 R 754003 A CLA!CMA!CLL!CML /SAY WE ARE REALLY DOING CODE THIS TIME
426 00344 R 420103 R XCT* XCODE /MAKE CODE
427 00345 R 201463 R EXX0 LAC MFPTR /GET ON-OFF SWITCH
428 00346 R 740200 A SZA /SKIP IF OFF
429 00347 R 703004 A EXX1 LSD /RESTART DISPLAY IF ON
430 00350 R 777777 A EXX2 LAW -1 /OK EXIT FLAG IN AC
431 00351 R 620050 R JMP* RETAD2 / TO CALLER
432 /
433 .EJECT

```

```

434 /
435 / VARIOUS ROUTINES AND SUBROUTINES
436 /
437 / PUT SUBROUTINE
438 /
439 / ENTER WITH AC CONTAINING INSTRUCTION TO BE PLACED
440 /
441 / STANDARD OPERATION: MOVE 'POINT', PLACE INDIRECT THRU 'POINT'
442 /
443 / IF PUTCNT IS -1, THE AC IS PLACED IN 'HOLD' INSTEAD:
444 / IN THIS CASE, 'POINT' IS NOT MODIFIED.
445 / THIS LAST MODE IS FOR ADDITION TO MAIND+SUB FILES.
446 /
447 00352 R 000000 A PUT 0
448 00353 R 440056 R ISZ PUTCNT /SKIP IF SPECIAL CASE AC-TO-HOLD
449 00354 R 600357 R JMP PUTPUT /NOPE, JUST PLACE
450 00355 R 040037 R DAC HOLD /SAVE FIRST PLACE MAIN AND SUB
451 00356 R 620352 R JMP* PUT
452 /
453 00357 R 440067 R PUTPUT ISZ POINT /MOVE POINTER TO NEXT
454 00360 R 060067 R DAC* POINT /PLACE, LEAVING POINTER AT LAST ONE
455 00361 R 620352 R JMP* PUT /OUT
456 /
457 /
458 /
459 /
460 / NEW FILE SET-UP FOR MAIN AND SUBP.
461 /
462 00362 R 501515 R NEWFIL AND (17777 /AC HAS FILE+1, STRIP TO VT-15 ADDR FIELD
463 00363 R 341516 R TAD (DJMPI /OP CODE MAKING JMP* FILE+1
464 00364 R 060072 R DAC* POINTS /PLACE AT FILE+2
465 00365 R 600234 R JMP MSTORE5 /GO TO GENERATE CODE, CAN'T BACK UP POINTERS
466 /
467 /
468 / SIZE
469 /
470 / ADDITIONAL CODE TO DETERMINE SIZE OF PREVIOUS GROUP
471 /
472 00366 R 440061 R SIZAPP ISZ POINT2 /LINE APPROX SIZE LIVES IN GROUP+1
473 00367 R 600312 R JMP SIZTST /REJOIN MAIN LOOP
474 /
475 00370 R 041465 R SIZPAR DAC ARG0 /INDICATE PREVIOUS GROUP PARAM
476 00371 R 440061 R SIZPRL ISZ POINT2 /MOVE TO NEXT
477 00372 R 220061 R LAC* POINT2 /FIGURE OUT IF GROUP IS DONE
478 00373 R 540750 R SAD LDNOP /IT IS IF NEXT OF LDNOP
479 00374 R 600404 R JMP SIZND /GO BACK UP POINT2 AND REJOIN
480 00375 R 541510 R SAD LDNOPB /AND IF BEGINNING OF NEXT GROUP
481 00376 R 600404 R JMP SIZND
482 00377 R 541507 R SAD LDNOPE /AND IF END OF THIS GROUP
483 00400 R 600405 R JMP SIZND1 /FETCH POINT2, DON'T BACK UP
484 00401 R 500021 R AND LNOP /STRIP TO 4 BITS OF OP CODE
485 00402 R 540750 R SAD LDNOP /IS IT A PRAMETER INSTR.

```

```

486 00403 R 600371 R      JMP      SIZPRL /YES, KEEP GOING
487 00404 R 751001 A      SIZND   SKP!CLA!CMA /BACK UP POINT2 TO GROUP END
488 00405 R 750000 A      SIZND1  CLA      /KEEP POINT2 SET
489 00406 R 340061 R      TAD      POINT2 /RETURN POINT HAS POINT2 IN AC
490 00407 R 600316 R      JMP      SIZJ1
491 /
492 /
493 /
494 / TABLE OF GROUP SIZES BY OP CODE
495 /
496 00410 R 000000 A      SIZETB  0      /SINGLE CHAR ILLEGAL
497 00411 R 000003 A      3      /CHAR STRING (ASSUMED INDIRECT
498 00412 R 000002 A      2      /ARBITRARY VECTOR LINE
499 00413 R 000002 A      2      /POINT INSTRUCTION, SET POINT
500 00414 R 000000 A      0      /PARAMETER HANDLED AS A SPECIAL CASE
501 00415 R 000006 A      L6     6      /SAVE-RESTORE, COPY WITH SAVE RESTORE
502 00416 R 000000 A      0      /BUNCH OF ILLEGALS. DON'T REMEMBER OP CODE
503 00417 R 000000 A      0      /ORDER EXACTLY
504 00420 R 000001 A      1      /BASIC VECTOR
505 00421 R 000000 A      0
506 00422 R 000000 A      0
507 00423 R 000000 A      0
508 00424 R 000000 A      0
509 00425 R 000003 A      3      /DJMS (ASSUME INDIR) COPY
510 00426 R 000000 A      0
511 00427 R 000000 A      0
512 /
513 /
514 / SUBROUTINE OPTPAR
515 /
516 / FETCHES OPTIONAL ARGUMENT FOR A PARAMETER TYPE CALL
517 / RETURNS ARGUMENT IN AC
518 /
519 /
520 / ENTERS WITH LINK=0
521 /
522 00430 R 000000 A      OPTPAR  0
523 00431 R 140746 R      DZM      PARBAK /DEFAULT FOR LATER, NO TRAILING DNOPE
524 00432 R 140741 R      DZM      PARFRN /AND NO LEADING DNOPE
525 00433 R 201464 R      LAC      LARG1 /ADDR OF FIRST ARG
526 00434 R 040742 R      DAC      TEMP1
527 00435 R 221466 R      LAC*     ARG1  /GET THE BITS
528 00436 R 440742 R      OPTP1   ISZ   TEMP1 /MOVE ARG POINTER
529 00437 R 745400 A      OPTP2   SZL!CLL /SKIP IF NO BIT
530 00440 R 600436 R      JMP      OPTP1 /BIT, MOVE POINTER
531 00441 R 740220 A      SZA!RAR /SKIP IF ALL DONE
532 00442 R 600437 R      JMP      OPTP2 /NOPE, KEEP GOING
533 00443 R 220742 R      LAC*     TEMP1 /ARGUMNT
534 00444 R 620430 R      JMP*     OPTPAR
535 /
536 /
537 / NEED ROUTINES

```

```

538 /
539 / THESE ROUTINES RETURN THE NUMBER OF LOCATIONS
540 / REQUIRED TO EDIT IN THIS GROUP, IN THE AC.
541 / REJOIN THE MAIN-LINE A NRETUR.
542 /
543 /
544 / OBTAIN THE NUMBER FOR COPY CODE
545 / ENTER WITH AC=0
546 /
547 00445 R 561466 R NEDCOP SAD* ARG1 /WITH RESTORE?
548 00446 R 777775 A LAW -3 /NO, MAKE TOTAL =3
549 00447 R 340415 R TAD L6 /YES, MAKE TOTAL =6
550 00450 R 600330 R JMP NRETUR
551 /
552 /
553 /
554 / OBTAIN THE NUMBER FOR PARAMETER CODE
555 /
556 00451 R 220072 R NEDPAR LAC* POINTS /CHECK FOR SPECIAL CASE PARAM/PARAM AND
557 00452 R 541510 R SAD LDNOPB /ALSO NO LEADING DNOPB
558 00453 R 600457 R JMP NEDFRN /GOT A DNOPB, NOT THAT CASE
559 00454 R 201465 R LAC ARG0 /NOW CHECK PARAM/PARAM
560 00455 R 540750 R SAD LDNOP /SIZPAR PUTS DNOP IN ARG0 !!!!
561 00456 R 600467 R JMP NEDP1 /SPECIAL CASE, NO LEADING DNOPB NEEDED!!
562 00457 R 220067 R NEDFRN LAC* POINT /FETCH PRECEDING
563 00460 R 541507 R SAD LDNOPB /AN END PARAMETER
564 00461 R 600467 R JMP NEDP1 /IF SO, WE DON'T NEED A BEGINNING PARM
565 00462 R 540750 R SAD LDNOP /A REGULAR DNOP
566 00463 R 600467 R JMP NEDP1 /YES, ALSO WE DON'T NEED 1
567 00464 R 500021 R AND LNOP /STRIP TO OP: LNOP=740000
568 00465 R 540750 R SAD LDNOP /IS A PARAMTER
569 00466 R 440741 R ISZ PARFRN /YES, SO WE NEED A DILIMITER
570 00467 R 200741 R NEDP1 LAC PARFRN /KEEP THE COUNT GOING
571 00470 R 040745 R DAC TEMP4 /BUT SAVE PARFRN FOR GENERATION TIME
572 00471 R 201517 R LAC (MASK /POINTER TO MASK LIST
573 00472 R 040743 R DAC TEMP2
574 00473 R 777774 A LAW -4 /LOOP CONTROL
575 00474 R 040744 R DAC TEMP3
576 00475 R 221466 R NEDP2 LAC* ARG1 /BITS
577 00476 R 520743 R AND* TEMP2 /ANY FOR THIS PARAMETER WORD
578 00477 R 740200 A SZA /SKIP IF NOT
579 00500 R 440745 R ISZ TEMP4 /AND COUNT 1 IF SO
580 00501 R 440743 R ISZ TEMP2 /MOVE TO NEXT MASK
581 00502 R 440744 R ISZ TEMP3 /LOOP CONTROL
582 00503 R 600475 R JMP NEDP2
583 00504 R 200072 R LAC POINTS /MAKE POINTER TO INSTR AFTER GROUP
584 00505 R 340745 R TAD TEMP4
585 00506 R 040742 R DAC TEMP1
586 00507 R 220742 R LAC* TEMP1 /INSTRUCTION FOLLOWING GROUP
587 00510 R 541510 R SAD LDNOPB /DOES IS STRART WITH A NOP
588 00511 R 600517 R JMP NEDP3 /YES, WE DON'T NEED ONE
589 00512 R 540750 R SAD LDNOP /IS A REGULAR DNOP

```

590	00513 R 600517 R	JMP	NEDP3	/ALSO WE DONT NEED
591	00514 R 500021 R	AND	LNOP	/CHECK IF PARAMETER
592	00515 R 540750 R	SAD	LDNOP	/PARAMTER
593	00516 R 440746 R	ISZ	PARBAK	/YES, WE NEED DNOPE
594	00517 R 200746 R	NEDP3	LAC	PARBAK
595	00520 R 340745 R	TAD	TEMP4	/MAKING TOTAL COUNT
596	00521 R 600330 R	JMP	NRETUR	
597				
598				

/

.EJECT

```

599
600
601
602
603
604 00522 R 000000 A
605 00523 R 200072 R
606 00524 R 501515 R
607 00525 R 341520 R
608 00526 R 100352 R
609 00527 R 341521 R
610 00530 R 100352 R
611 00531 R 201466 R
612
613
614
615 00532 R 100352 R
616
617
618
619
620 00533 R 777776 A
621 00534 R 040745 R
622 00535 R 221467 R
623 00536 R 741301 A
624 00537 R 620522 R
625 00540 R 600543 R
626
627 00541 R 441466 R
628 00542 R 441466 R
629 00543 R 340055 R
630 00544 R 741100 A
631 00545 R 600541 R
632 00546 R 340533 R
633
634 00547 R 744110 A
635 00550 R 600553 R
636 00551 R 441466 R
637 00552 R 340110 R
638 00553 R 740201 A
639 00554 R 440745 R
640 00555 R 340605 R
641 00556 R 040744 R
642 00557 R 221466 R
643 00560 R 520744 R
644 00561 R 440744 R
645 00562 R 260744 R
646 00563 R 440744 R
647 00564 R 061466 R
648 00565 R 441466 R
649 00566 R 440745 R
650 00567 R 600557 R

/
/ CODE GENERATION SUBROUTINES
/
/ CHARACTER STRING
/
CHR 0
LAC POINTS /GROUP STARTING ADDR
AND (17777 /KEEP 13 BITS
TAD (CHARSI+2 /STRING INDIRECT TO .+2
JMS PUT /PLACE IT
TAD (DJMP-CHARSI+1 /ADD NEW INST-OLD INST
JMS PUT
LAC ARG1 /STRING ADDR
.IFDEF NONXLM
AND L77777 /MSK OFF ARRAY TYPE BITS FROM ADDR!!
.ENDC
JMS PUT

/
/TEST N AND INSERT ALT IF .NE. ZERO.
/
LM2 LAW -2 /MAX LOOP CONTROL FOR WORD MODIFY
DAC TEMP4 /FOR ALT-MODE INSERTION
LAC* N /GET NO. OF CHARS.
SNA!SPA!CMA /SKIP IF MUST INSERT ALT-MODE
JMP* CHR /NOPE
JMP TST2 /JOIN DIVIDE BY 5 LOOP

/
TST1 ISZ STR /MOVE IOPS ASCII POINTER BY 1 PAIR
ISZ STR
TST2 TAD L5 /ADD 5 UNTIL >=0
SPA /CHECK DONE
JMP TST1 /NOPE
TAD LM2 /CHAR POSITIONS 1,2,3,4,5 HAVE AC VALUE
/OF 2,1,0,-1,-2
/VALUES 4,2,0,-2,-4
SMA!RAL!CLL /VALUES 4,2,0 FOR POSITIONS 1,2,3
JMP TST3 /POSITIONS 4,5 ARE IN SECOND WORD
ISZ STR /VALUES 10,6 FOR POSITIONS 4,5
TAD L12 /VALUES -5,-3,-1,-11,-7
TST3 SZA!CMA /LOOP CONTROL -1 XCPT POSITION 3
ISZ TEMP4 /MAKE POINTER FOR CORRECT MASK
TAD ALTPNT
DAC TEMP3
TST4 LAC* STR /GET WORD OF IOPS PAIR FOR MODIFY
AND* TEMP3 /STRIP OUT CHAR
ISZ TEMP3 /MOVE POINTER TO CORRECT ALT-MODE
XOR* TEMP3 /INSERT ALT
ISZ TEMP3 /MOVE POINTER IN CASE SECOND PASS
DAC* STR /REPLACE
ISZ STR /IN CASE SECOND PASS
ISZ TEMP4 /LOOP CONTROL
JMP TST4 /DO A SECOND WORD MODIFY

```

```

651      00570 R 620522 R      JMP*   CHR    /WHEW! DONE
652
653      /
654      / MASKS FOR INSERTION OF ALT-MODE IN EACH POSITION
655      /
656      00571 R 700377 A      700377    /POSITION 4
657      00572 R 076400 A      076400    /ITS ALT-MODE
658      00573 R 777401 A      777401    /POSITION 5
659      00574 R 000372 A      000372    /ITS ALTMODE
660      00575 R 003777 A      003777    /POSITION 1
661      00576 R 764000 A      764000    /ALT
662      00577 R 774017 A      774017    /POSITION 2
663      00600 R 003720 A      003720    /ALT
664      00601 R 777760 A      LM20  777760 /POSITION 3, FIRST PART
665      00602 R 000017 A      L17   000017 /PART OF ALT MODE
666      00603 R 077777 A      L77777 077777 /SECOND PART
667      00604 R 500000 A      LTS    500000 /END OF ALT MODE
668      00605 R 000602 R      ALTPNT L17   /ADD TO AC VALUE TO OBTAIN CORRECT POINTER
669      /
        .EJECT

```

```

670
671
672
673
674
675      00606 R 000000 A
676      00607 R 201464 R
677      00610 R 040064 R
678      00611 R 140742 R
679      00612 R 140743 R
680      00613 R 140744 R
681      00614 R 140745 R
682      00615 R 221466 R
683      00616 R 745220 A
684      00617 R 600633 R
685      00620 R 040053 R
686      00621 R 744400 A
687      00622 R 600630 R
688      00623 R 440064 R
689      00624 R 220064 R
690      00625 R 041466 R
691      00626 R 221466 R
692      00627 R 620656 R
693      00630 R 200053 R
694      00631 R 440627 R
695      00632 R 600616 R
696      00633 R 200655 R
697      00634 R 040627 R
698      00635 R 201522 R
699      00636 R 040053 R
700      00637 R 340415 R
701      00640 R 040522 R
702      00641 R 777772 A
703      00642 R 040064 R
704      00643 R 220053 R
705      00644 R 741200 A
706      00645 R 600650 R
707      00646 R 260522 R
708      00647 R 100352 R
709      00650 R 440053 R
710      00651 R 440522 R
711      00652 R 440064 R
712      00653 R 600643 R
713      00654 R 620606 R
714      00655 R 620656 R
715
716
717
718      00656 R 000667 R
719      00657 R 000674 R
720      00660 R 000703 R
721      00661 R 000711 R

```

/
 /
 /PRM - SETUP DISPLAY PARAMETER INSTRUCTIONS FOR BOTH
 / MAIN AND SUBPICTURE ENTRIES
 /
 /

PRM	0			
	LAC	LARG1		/ADDR OF ARG1
	DAC	REGT1		/FEATURES THAT NEED SETTING.
	DZM	TEMP1		/CLEAR WORDS TO BE
	DZM	TEMP2		/INSERTED IN DISPLAY FILE.
	DZM	TEMP3		
	DZM	TEMP4		
	LAC*	ARG1		/BITS INDICATING WHICH
PRM10	SNA!RCR			/DONE IF AC=0
	JMP	PRM25		/GO FINISH UP.
	DAC	REGT0		/SAVE REST.
	SNL!CLL			/TEST FEATR BIT IN L.
	JMP	PRM20		/NOT ON.
	ISZ	REGT1		/GET NEXT SETTING VALUE.
	LAC*	REGT1		
	DAC	ARG1		
	LAC*	ARG1		/ACTUAL VALUE IN AC.
PRM15	JMP*	PRMTBL		/DISPATCH.
PRM20	LAC	REGT0		/FETCH REMAINING BITS
	ISZ	PRM15		/POINT TO NEXT ROUTINE.
	JMP	PRM10		/RECYCLE.
PRM25	LAC	JINST		/RESET DISPATCHER.
	DAC	PRM15		
	LAC	(PARFRN	/POINTER TO DATA FOR SIX INSTRUCTIONS	
	DAC	REGT0		
	TAD	L6	/POINTER TO OP CODES FOR THOSE 6 INSTRUCTIONS	
	DAC	REGT2		
	LAW	-6	/LOOP CONTROL	
	DAC	REGT1		
PARLOP	LAC*	REGT0	/INSTRUCTION DATA	
	SNA		/SKIP IF THERE IS INDEED DATA FOR THIS ONE	
	JMP	PARL1	/NO DATA, DON'T PLACE IT	
	XOR*	REGT2	/OP CODE WITH DATA=FULL GRAPHICS INSTR.	
	JMS	PUT	/PLACE IN FILE	
PARL1	ISZ	REGT0	/MOVE DATA FETCH POINTER	
	ISZ	REGT2	/MOVE OP CODE FETCH POINTER	
	ISZ	REGT1	/CONTROL	
	JMP	PARLOP	/MORE	
	JMP*	PRM	/NO MORE	
JINST	JMP*	PRMTBL		

/
 /DISPATCH TABLE FOR PARAMETER SETTINGS
 /
 /

PRMTBL	SCALE
	INTENS
	LTPN
	BLINK

722	00662 R 000715 R	DASH		
723	00663 R 000721 R	OFFSET		
724	00664 R 000725 R	ROTAT		
725	00665 R 000731 R	NAME		
726	00666 R 000736 R	SYNC		
727		/		
728		/GENERATE SCALE SETTING FROM VALUE IN AC.		
729		/		
730	00667 R 500602 R	SCALE	AND L17	/MASK SETTING.
731	00670 R 241505 R		XOR L20	/ENABLE SCALE CHANGE.
732	00671 R 240742 R	T1R	XOR TEMP1	/FORM SETTING(S) FOR PARAM. WD. 1.
733	00672 R 040742 R		DAC TEMP1	/KEEP FOR DISPLAY FILE.
734	00673 R 600630 R		JMP PRM20	/RETURN TO DISPATCHER
735		/		
736		/GENERATE INTENSITY SETTING FROM VALUE IN AC.		
737		/		
738	00674 R 500105 R	INTENS	AND L7	/KEEP ONLY DATA BITS
739	00675 R 340077 R		TAD L10	/ENABLE BIT
740	00676 R 744010 A		CLL!RAL	
741	00677 R 742010 A		RTL; RTL; RTL	/MOVE SETTING INTO POSITION
	00700 R 742010 A			
	00701 R 742010 A			
742	00702 R 600671 R		JMP T1R	/INCLUDE IN PARAM. WD. 1.
743		/		
744		/GENERATE LIGHT PEN SETTING FROM VALUE IN AC.		
745		/		
746	00703 R 740200 A	LTPN	SZA	/1=ON, 0=OFF.
747	00704 R 201523 R		LAC (4	/ON BIT.
748	00705 R 340077 R		TAD L10	/ENABLE LIGHT PEN SETTING.
749	00706 R 240743 R	T2R	XOR TEMP2	/FORM SETTING(S) FOR PARAM. WD. 2.
750	00707 R 040743 R		DAC TEMP2	/KEEP FOR DISPLAY FILE
751	00710 R 600630 R		JMP PRM20	/RETURN TO DISPATCHER.
752		/		
753		/		
754		/GENERATE BLINK SETTING FROM VALUE IN AC.		
755		/		
756	00711 R 740200 A	BLINK	SZA	/1=ON, 0=OFF
757	00712 R 201524 R		LAC (400	/ON BIT.
758	00713 R 341525 R		TAD (BK	/ENABLE BLINK
759	00714 R 600706 R		JMP T2R	/INCLUDE IN PARAMETER WORD 2.
760		/		
761		/GENERATE DASH SETTING FROM VALUE IN AC		
762		/		
763	00715 R 500044 R	DASH	AND L3	/MASK SETTING.
764	00716 R 241523 R		XOR (DSH	/ENABLE DASH.
765	00717 R 040744 R		DAC TEMP3	/KEEP FOR DISPLAY FILE.
766	00720 R 600630 R		JMP PRM20	/RETURN TO DISPATCHER
767		/		
768		/GENERATE OFFSET SETTING FROM VALUE IN AC.		
769		/		
770	00721 R 740200 A	OFFSET	SZA	
771	00722 R 200041 R		LAC L1	/MASK SETTING.

```

772      00723 R 240052 R      XOR      L2      /ENABLE OFFSET
773      00724 R 600706 R      JMP      T2R      /INCLUDE IN PARAMETER WORD 2.
774
775      /GENERATE ROTATE SETTING FROM VALUE IN AC.
776
777      00725 R 740200 A      ROTAT     SZA      /1=ON, 0=OFF.
778      00726 R 201505 R      LAC      L20      /ON BIT.
779      00727 R 341526 R      TAD      (ROT      /ENABLE ROTATE.
780      00730 R 600706 R      JMP      T2R      /INCLUDE IN PARAMETER WORD 2.
781
782      /SET NAME REGISTER FROM VALUE IN AC.
783
784      00731 R 501527 R      NAME      AND      (177      /MASK NAME SETTING.
785      00732 R 241524 R      XOR      (LODN      /ENABLE LOAD NAME.
786      00733 R 240745 R      T4R      XOR      TEMP4      /FORM SETTING(S)FOR SKIP2 WORD.
787      00734 R 040745 R      DAC      TEMP4      /KEEP FOR DISPLAY FILE.
788      00735 R 600630 R      JMP      PRM20      /RETURN TO DISPATCHER.
789
790      /SET SYNC FROM VALUE IN AC
791
792      00736 R 740200 A      SYNC      SZA      /1=ON, 0=OFF.
793      00737 R 201530 R      LAC      (2000      /ON BIT.
794      00740 R 600733 R      JMP      T4R      /INCLUDE IN SKIP2 WORD.
795
796      /
797      / LIST OF DATA HOLDERS FOR MAX OF SIX INSTRU
798
799      / THE NEXT 12 LOCATIONS MUST STAY IN PRESENT ORDER!
800
801      00741 R 000000 A      PARFRN   0      /FRONT NOP, 1 IF NEEDED, 0 IF NOT
802      00742 R 000000 A      TEMP1     0      /DATA FOR PRM1
803      00743 R 000000 A      TEMP2     0      /PRM2
804      00744 R 000000 A      TEMP3     0      /PRM3
805      00745 R 000000 A      TEMP4     0      /SKP2 (TEMP1-4 SOMETIMES USED AS TEMP)
806      00746 R 000000 A      PARBAK    0      /BACK NOP, 1 IF NEEDED, 0 IF NOT
807
808      / LIST OF OP CODES FOR POSSIBLE PARAMTER INSTRUCTIONS
809
810      00747 R 200000 A      PAROPS    DNOP      /DNOP.XOR.1=DNOPB
811      00750 R 200000 A      LDNOP     PRM1      /REGULAR PARAMETER 1 INSTRU
812      00751 R 210000 A      PRM2
813      00752 R 220000 A      PRM3
814      00753 R 234000 A      SKP2
815      00754 R 200005 A      DNOPE\1      /XOR'ED INTO 1 GIVES DNOPE
816
817      /
818      .EJECT

```

```

817
818
819
820 00755 R 000000 A
821 00756 R 441467 R
822 00757 R 361466 R
823 00760 R 200072 R
824 00761 R 501515 R
825 00762 R 740400 A
826 00763 R 600773 R
827 00764 R 341531 R
828 00765 R 100352 R
829 00766 R 341532 R
830 00767 R 100352 R
831 00770 R 201467 R
832 00771 R 100352 R
833 00772 R 620755 R
834
835 00773 R 341533 R
836 00774 R 040742 R
837 00775 R 100352 R
838 00776 R 341534 R
839 00777 R 100352 R
840 01000 R 341535 R
841 01001 R 100352 R
842 01002 R 201467 R
843 01003 R 100352 R
844 01004 R 100352 R
845 01005 R 200742 R
846 01006 R 341536 R
847 01007 R 600771 R
848
849
850
851 01010 R 361467 R
852 01011 R 740001 A
853 01012 R 040743 R
854 01013 R 221470 R
855 01014 R 740200 A
856 01015 R 201537 R
857 01016 R 241540 R
858 01017 R 040742 R
859 01020 R 221466 R
860 01021 R 501513 R
861 01022 R 240742 R
862 01023 R 100352 R
863 01024 R 441466 R
864 01025 R 440743 R
865 01026 R 601020 R
866 01027 R 600237 R
867
868

/
/ GENERATE CODE FOR COPY
/
CPY 0
ISZ ARG2 /CALLED FILE, MAKE IT FILE+1
TAD* RST /AC=-1, CLEAR LINK IF RST NON0
LAC POINTS /SET UP 13 BITS OF ADDR
AND (17777 /FOR PLACEMENT IN ADDRESSES
SNL /SKIP IF RESTORE ARGUMENT WAS 0
JMP CPY1 /RESTORE ON, GO DO ITS CODE
TAD (DJMSI+2 /JMS THRU GROUP+2
JMS PUT /PLACE INSTRUCTIONS THRU PUT
TAD (DJMP-DJMSI+1 /ADD NEW INST-OLD INST
JMS PUT
LAC ARG2 /FETCH DISPLAY SUBROUTINE ADDR
CPY2 JMS PUT
JMP* CPY

/
CPY1 TAD (SAVE+4 /SAVE INSTRUCTION, GROUP+4
DAC TEMP1 /SAVE TO MAKE LATER RESTORE INSTRU
JMS PUT
TAD (DJMSI-SAVE-1 /NET DIFFERENCE BETWEEN INST'S
JMS PUT
TAD (DJMP-DJMSI+2
JMS PUT
LAC ARG2 /ADDR OF SUBPICTURE CALLED
JMS PUT
JMS PUT /LEAVE HOLE FOR RESTORED WORD
LAC TEMP1 /FETCH SAVE+4
TAD (REST-SAVE /MAKE RESTORE GROUP+4
JMP CPY2 /JOIN FOR LAST 2 INST'S

/
/ GENERATE CODE FOR GRAPH
/
GRPH TAD* ARG2 /AC=-1 GIVES COUNT-1
CMA /WHICH GIVES -COUNT
DAC TEMP2 /HOLD FOR LOOP CONTROL
LAC* ARG3 /GET AXIS SETTING
SZA
LAC (4000
XOR (GY
DAC TEMP1 /OP CDOE THRUOUT
GRPH5 LAC* ARG1 /GET A POINT
AND (1777 /MAKE SURE NO XTRA BITS
XOR TEMP1 /OP CODE
JMS PUT /PLACE POINT
ISZ ARG1 /MOVE INPUT POINTER
ISZ TEMP2 /CHECK COUNT
JMP GRPH5 /KEEP GOING
JMP MRETUR /RETURN TO MAIN-SUB CODE
/
/

```

CODE FOR BLANK AND UNBLANK

```

869
870
871 01030 R 754000 A BLN CLLIC /INDICATE BLN BY LINK OF
872 01031 R 040745 R UBL DAC TEMP4 /SAVE AC (SAME AS LINK)
873 / /UBL CALL HAVING AC, LINK ALREADY -1
874 01032 R 340033 R TAD LTEMP2 /BLN POINTS TEMP2;UBL POINTS TEMP1
875 01033 R 040744 R DAC TEMP3 /USED TO PLACE EOF POINTER
876 01034 R 201466 R LAC ARG1 /FILE ADDR
877 01035 R 340052 R TAD L2 /MAKE POINTER TO FILE+2
878 01036 R 040742 R DAC TEMP1 /PLACE TWICE; ONCE OVERWRITE
879 01037 R 040743 R DAC TEMP2
880 01040 R 221466 R LAC* ARG1 /LENGTH POINTER
881 01041 R 741200 A SNA /BETTER NOT BE ZERO
882 01042 R 620050 R JMP* RETAD2 /JUST RETURN TO CALLER
883 01043 R 341466 R TAD ARG1 /POINT TO END OF FILE
884 01044 R 060744 R DAC* TEMP3 /PLACE EOF IN ONE OR OTHER
885 01045 R 220742 R LAC* TEMP1 /GET A NON-JUMP
886 01046 R 040744 R DAC TEMP3 /SAVE IN TEMPORARY
887 01047 R 501541 R AND (760000 /CHECK IF NON-JUMP IS ALREADY A JUMP
888 01050 R 541516 R SAD (DJMP1 /IF SO, NO WORK TO DO
889 01051 R 620050 R JMP* RETAD2 /OUT
890 01052 R 220743 R LAC* TEMP2 /GET A JUMP
891 01053 R 060742 R DAC* TEMP1 /PLACE OVER THE OTHER END
892 /
893 / THIS GIVES A TRANSITION STATE WITH A JUMP AT EACH END
894 /
895 01054 R 740400 A SNL /SKIP IF UNBLANK
896 01055 R 703044 A STPD /STOP DISPLAY
897 01056 R 200744 R LAC TEMP3 /PLACE NON-JUMP OVER JUMP
898 01057 R 060743 R DAC* TEMP2
899 01060 R 440745 R ISZ TEMP4 /NEED TO TURN ON TUBE AGAIN?
900 01061 R 600345 R JMP EXX0 /YES
901 01062 R 600350 R JMP EXX2 /NO
902 /
903 / GENERATE POINT CODE
904 /
905 /
906 01063 R 000000 A PNT 0
907 01064 R 221467 R LAC* ARG2 /GET Y POINT
908 01065 R 501513 R AND (1777 /KEEP ONLY 10 BITS
909 01066 R 241542 R XOR (PY /MAKE UP OP CODE
910 01067 R 100352 R JMS PUT /PLACE IN DISPLAY LISE
911 01070 R 221466 R LAC* ARG1 /AND TH X POINT
912 01071 R 501513 R AND (1777
913 01072 R 241543 R XOR (PX
914 01073 R 100352 R JMS PUT
915 01074 R 621063 R JMP* PNT
916 /
917 / DINIT CODE TO START UP DISPLAY
918 /
919 01075 R 703044 A DNT STPD /STOP TUBE IN CASE RUNNING
920 .INIT DAT/VT.1,SICW /CLR FLAGS, ENABL LP,PB

```

```

01076 R 001010 A *G      CAL+1*1000 DAT/VT&777
01077 R 000001 A *G      1
01100 R 001126 R *G      SICW+0
01101 R 000000 A *G      0
921   01102 R 201466 R      LAC      ARG1      /NEW MAINFL ADDR
922   01103 R 041462 R      DAC      MAINFL    /SAVE FOR MAIN ROUTINE USE
923   01104 R 040067 R      DAC      POINT    /GRAB PUT MECHANISM, TOP OF MAIN
924   01105 R 100352 R      JMS      PUT      /DUMMY, SINCE SUBP STORES FIRST ARG
925   01106 R 340052 R      TAD      L2       /ADDR FILE+2
926   01107 R 100352 R      JMS      PUT      /PUT IN FILE+1
927   01110 R 221462 R      LAC*     MAINFL    /FIND OUT IF NEW MAIN FILE
928   01111 R 740200 A      SZA      /SKIP IF YES
929   01112 R 601123 R      JMP      MANCOM   /NOPE
930
931   /
932   / SET UP NEW FILE
933   /
933   01113 R 201544 R      LAC      (215052 /TURN OFF BLINK, OFFSET, ETC.
934   01114 R 100352 R      JMS      PUT      /FILE+2
935   01115 R 201462 R      LAC      MAINFL    /MAKE JMP* FILE+1
936   01116 R 501515 R      AND      (17777    /13 BIT VT-15 ADDRESSING
937   01117 R 341545 R      TAD      (DJMPI+1
938   01120 R 100352 R      JMS      PUT      /TO FILE+3
939   01121 R 200044 R      LAC      L3       /LENGTH POINTER OF THREE
940   01122 R 061462 R      DAC*     MAINFL    /PLACE AS LENGTH POINTER
941   01123 R 341462 R      MANCOM   TAD      MAINFL    /MAKE END POINTER
942   01124 R 041463 R      DAC      MFPTTR   /USED BY TRACKING ROUTINE
943   01125 R 600347 R      JMP      EXX1     /START, -1 TO AC, EXIT
944
945   01126 R 257740 A      /
946   / SICW      257740      /CLEAR FLAGS, ENABLE PB, LP INTERRUPTS
947   /
948   / ACTION FOR LINESIZE CALL
949   /
950   01127 R 777776 A      LNS      LAW      -2      /TOUGH. THIS MAKES CKNSDW NOT SKIP
951   /
952   /
953   01130 R 101137 R      JMS      VECTOR   /SO ON REGULAR EXIT FROM VECTOR
954   01131 R 061470 R      DAC*     ARG3     /SIZE IS RETURNED, BUT NO CODE GENER.
955   01132 R 620050 R      JMP*     RETAD2   /BUT ON 0,0 EXIT, SEE ZEROX
956   /
957   01133 R 440053 R      ZEROX    ISZ      CKNSDW  /LINE 0,0
958   01134 R 440053 R      ISZ      CKNSDW
959   01135 R 620050 R      JMP*     RETAD2   /ALL BUT LNS, EXIT TO CALLER
960   01136 R 621137 R      JMP*     VECTOR   /BUT LNS MUST RETURN, TO SEND
961   /
962   / CALLER THE ARGUMENT!!!
          .EJECT

```

```

963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012

```

01137 R 000000 A	VECTOR 0		
01140 R 040053 R	DAC	CKNSDW	/SO IT SKIPS IF REALLY TO MAKE LINES!
01141 R 141063 R	DZM	APPRSW	/CLEAR APPROXIMATION SWITCH.
01142 R 221467 R	LAC*	ARG1+1	/GET DELTA Y.
01143 R 101255 R	JMS	SIGNM	/CHECK SIGN AND MAG..
01144 R 501513 R	AND	(1777	/MASK TO TEN BITS.
01145 R 040522 R	DAC	DELTAY	/AND SAVE.
01146 R 751400 A	SZL!CLA		
01147 R 740001 A	CMA		/SIGN OF DELTAY; 0=PLUS, -1=MINUS.
01150 R 040606 R	DAC	SOY	
01151 R 221466 R	LAC*	ARG1	/GET DELTAX.
01152 R 101255 R	JMS	SIGNM	/CHECK SIGN AND MAG..
01153 R 501513 R	AND	(1777	/MASK TO TEN BITS.
01154 R 040064 R	DAC	DELTAX	/AND SAVE.
01155 R 751400 A	SZL!CLA		
01156 R 740001 A	CMA		/SIGN OF DELTAX; 0=PLUS, -1=MINUS.
01157 R 040755 R	DAC	SOX	
01160 R 200522 R	LAC	DELTAY	/START TESTS FOR BASIC VECTOR.
01161 R 741200 A	SNA		
01162 R 601215 R	JMP	YZERO	/DELTAY .EQ. ZERO
01163 R 200064 R	LAC	DELTAX	/DELTAY .NE. ZERO
01164 R 741200 A	SNA		
01165 R 601232 R	JMP	XZERO	/DELTAX .EQ. ZERO
01166 R 540522 R	SAD	DELTAY	/DELTAX .NE. ZERO
01167 R 601237 R	JMP	YEQX	/ABS (DELTAY) .EQ. ABS (DELTAX)
01170 R 750000 A	CLA		
01171 R 703142 A	RS3		/ABS (DELTAY) .NE. ABS (DELTAX)
01172 R 501546 R	AND	(10000	/CHECK RV OPTION AVAILABLE.
01173 R 741200 A	SNA		/RV AVAILABLE, INSERT COMMANDS.
01174 R 601273 R	JMP	APPROX	
01175 R 200052 R	LAC	L2	
01176 R 440053 R	ISZ	CKNSDW	/SKIP IF TO DO REAL WORK
01177 R 621137 R	JMP*	VECTOR	/ON, RETURN.
	.EJECT		

1013	01200 R 200755 R	LAC	SOX	
1014	01201 R 740200 A	SZA		/SET SIGN FOR Y VECTOR
1015	01202 R 201537 R	LAC	(4000	
1016	01203 R 240064 R	XOR	DELTAX	
1017	01204 R 241547 R	XOR	(STRK	/STROKE OP CODE
1018	01205 R 101263 R	JMS	LSTOR	/AND STORE IT.
1019	01206 R 200606 R	LAC	SOY	
1020	01207 R 740200 A	SZA		/SET SIGN FOR X VECTOR
1021	01210 R 201537 R	LAC	(4000	
1022	01211 R 240522 R	XOR	DELTAY	
1023	01212 R 241547 R	XOR	(STRK	/FORM SECOND INSTRUCTION
1024	01213 R 101263 R	JMS	LSTOR	/AND STORE IT.
1025	01214 R 621137 R	JMP*	VECTOR	
1026	01215 R 540064 R	YZERO	SAD	DELTAX
1027	01216 R 601133 R	JMP	ZEROX	/DELTAX .EQ. ZERO TOO?
1028	01217 R 540755 R	SAD	SOX	/GO FIND TO WHOM TO RETURN
1029	01220 R 341550 R	TAD	(-10000	/NO, CHECK SIGN OF DELTAX.
1030	01221 R 341551 R	TAD	(V4	/PLUS, VECTOR DIREC. 0
1031	01222 R 240064 R	YZER5	XOR	/MINUS, VECTOR DIREC. 4
1032	01223 R 040742 R	DAC	TEMP1	/FORM REST OF INSTRUCTION.
1033	01224 R 200041 R	LAC	L1	/SAVE IT.
1034	01225 R 440053 R	ISZ	CKND5W	/1 LOC NEEDED
1035	01226 R 621137 R	JMP*	VECTOR	/SKIP IF TO DO REAL WORK
1036	01227 R 200742 R	LAC	TEMP1	
1037	01230 R 101263 R	JMS	LSTOR	/STORE IN DISPLAY FILE.
1038	01231 R 621137 R	JMP*	VECTOR	
1039	01232 R 540606 R	XZERO	SAD	/X .EQ. 0, CHECK SIGN OF DELTAY.
1040	01233 R 341550 R	TAD	(-10000	/PLUS, VECTOR DIREC. 2
1041	01234 R 341552 R	TAD	(V6	/MINUS, VECTOR DIREC. 6
1042	01235 R 240522 R	XOR	DELTAY	/FORM REST OF INSTRUCTION.
1043	01236 R 601223 R	JMP	YZER5+1	/STORE IT AND RETURN.
1044	01237 R 750000 A	VEGX	CLA	/ABS DELTAY .EQ. ABS DELTAX
1045	01240 R 540755 R	SAD	SOX	/CHECK SIGN OF X.
1046	01241 R 601245 R	JMP	VEGX5	/PLUS.
1047	01242 R 540606 R	SAD	SOY	/MINUS, CHECK SIGN OF Y.
1048	01243 R 601251 R	JMP	DTAB+2	/DELTAX=MINUS, DELTAY=PLUS.
1049	01244 R 601250 R	JMP	DTAB+1	/DELTAX=MINUS, DELTAY=MINUS.
1050	01245 R 540606 R	VEGX5	SAD	/DELTAX=PLUS, CHECK DELTAY.
1051	01246 R 601252 R	JMP	DTAB+3	/PLUS, VECTOR DIREC. 1
1052	01247 R 341537 R	DTAB	TAD	/MINUS, VECTOR DIREC 7.
1053	01250 R 341537 R	TAD	(4000	/VECTOR DIREC. 5
1054	01251 R 341537 R	TAD	(4000	/VECTOR DIREC. 3
1055	01252 R 341553 R	TAD	(V1	/VECTOR DIREC. 1
1056	01253 R 601222 R	JMP	YZER5	/GO FORM INST. STORE, AND RETURN.
1057	01254 R 621137 R	JMP*	VECTOR	/RETURN
1058		.EJECT		

```

1059      /
1060      /SIGNM - SEPERATE SIGN AND MAGNITUDE.  ENTER WITH DELTA IN
1061      /AC, EXIT WITH ABS (DELTA) IN AC, L=0 FOR PLUS,
1062      /L=1 FOR MINUS.
1063      /
1064      01255 R 000000 A      SIGNM      0
1065      01256 R 744100 A      CLL!SMA
1066      01257 R 621255 R      JMP*      SIGNM      /CLEAR L AND EXIT IF POS.
1067      01260 R 744003 A      CMA!STL      /TWO'S COMPLEMENT AC
1068      01261 R 340041 R      TAD      L1      /AND SET L IF NEG.
1069      01262 R 621255 R      JMP*      SIGNM
1070      /
1071      /LSTOR - SET INTENSIFY BIT IF NECESSARY, AND STORE VECTOR INST.
1072      /
1073      01263 R 000000 A      LSTOR      0
1074      01264 R 040742 R      DAC      TEMP1      /SAVE TEMPORARILY
1075      01265 R 221470 R      LAC*      LT
1076      01266 R 750200 A      SZA!CLA      /CHECK INTENSITY ARG.
1077      01267 R 201536 R      LAC      (20000      /ON, SET INTENSIFY BIT.
1078      01270 R 240742 R      XOR      TEMP1
1079      01271 R 100352 R      JMS      PUT      /PLACE IT
1080      01272 R 621263 R      JMP*      LSTOR
1081      /
1082      /
1083      /
1084      /
1085      .EJECT

```

```

1086
1087
1088
1089
1090
1091 01273 R 200064 R
1092 01274 R 740001 A
1093 01275 R 340041 R
1094 01276 R 340522 R
1095 01277 R 741100 A
1096 01300 R 601313 R
1097 01301 R 200522 R
1098 01302 R 744020 A
1099 01303 R 041474 R
1100 01304 R 200064 R
1101 01305 R 744020 A
1102 01306 R 041475 R
1103 01307 R 200606 R
1104 01310 R 750200 A
1105 01311 R 601325 R
1106 01312 R 601327 R
1107 01313 R 200064 R
1108 01314 R 744020 A
1109 01315 R 041474 R
1110 01316 R 200522 R
1111 01317 R 744020 A
1112 01320 R 041475 R
1113 01321 R 200755 R
1114 01322 R 750200 A
1115 01323 R 601326 R
1116 01324 R 601330 R
1117
1118
1119
1120 01325 R 341537 R
1121 01326 R 341537 R
1122 01327 R 341537 R
1123 01330 R 341554 R
1124 01331 R 040075 R
1125 01332 R 200755 R
1126 01333 R 340606 R
1127 01334 R 741200 A
1128 01335 R 601346 R
1129 01336 R 340041 R
1130 01337 R 751100 A
1131 01340 R 601344 R
1132 01341 R 540606 R
1133 01342 R 601345 R
1134 01343 R 341537 R
1135 01344 R 341537 R
1136 01345 R 341537 R
1137 01346 R 341555 R

/
/ APPROX - APPROXIMATE RV TYPE LINE USING BASIC VECTORS AND
/ THE BRESENHAM ALGORITHM. (UNIT MOVE IS 2 RASTER UNITS)
/
/
/ APPROX LAC DELTAX /DETERMINE MAJOR DELTA
/ CMA
/ TAD L1
/ TAD DELTAY
/ SPA
/ JMP APPR5 /DELTAX IS MAJOR DELTA
/ LAC DELTAY /DELTAY IS MAJOR DELTA
/ CLL!RAR
/ DAC UVAR /SAVE DELTAY/2 AS MAJOR DELTA
/ LAC DELTAX
/ CLL!RAR
/ DAC VAR /SAVE DELTAX/2 AS MINOR DELTA
/ LAC SOY /TEST SIGN OF Y TO DETERMINE
/ SZA!CLA /DIRECTION FOR MOVE1
/ JMP MITAB /D6
/ JMP MITAB+2 /D2
/ LAC DELTAX /DELTAX IS MAJOR DELTA
/ CLL!RAR
/ DAC UVAR /SAVE DELTAX/2 AS MAJOR DELTA
/ LAC DELTAY
/ CLL!RAR
/ DAC VAR /SAVE DELTAY/2 AS MINOR DELTA
/ LAC SOX /TEST SIGN OF X TO DETERMINE
/ SZA!CLA /DIRECTION FOR MOVE1
/ JMP MITAB+1 /D4
/ JMP MITAB+3 /D0
/
/ MITAB, M2TAB - DETERMINE AND STORE VECTORS FOR MOVE1 AND MOVE2.
/
/ MITAB TAD (4000 /D6
/ TAD (4000 /D4
/ TAD (4000 /D2
/ TAD (400002 /D0 (UNIT VECTOR=2 RU)
/ DAC MOVE1 /SAVE MOVE1
/ LAC SOX
/ TAD SOY /TEST SIGN OF X AND Y TO
/ SNA /DETERMINE DIRECTION OF MOVE2.
/ JMP M2TAB+3 /((++)=D1
/ TAD L1
/ SPA!CLA
/ JMP M2TAB+1 /((-)=D5
/ SAD SOY /((+)=D7
/ JMP M2TAB+2 /((-+)=D3
/ M2TAB TAD (4000 /D7
/ TAD (4000 /D5
/ TAD (4000 /D3
/ TAD (402002 /D1 (UNIT VECTOR=2RU)

```

PAGE 29

VTPRIM 136

1138
1139

01347 R 040100 R

DAC
.EJECT

MOVE2

/SAVE MOVE2

```

1140
1141
1142
1143 01350 R 201475 R      LAC      VAR
1144 01351 R 740001 A      CMA
1145 01352 R 340041 R      TAD      L1
1146 01353 R 341474 R      TAD      U/AR      /U-V
1147 01354 R 744010 A      CLL!RAL      /2(U-V)
1148 01355 R 041476 R      DAC      U/AR
1149 01356 R 201475 R      LAC      VAR
1150 01357 R 744010 A      CLL!RAL
1151 01360 R 041477 R      DAC      B      /B=2V
1152 01361 R 740001 A      CMA
1153 01362 R 340041 R      TAD      L1
1154 01363 R 341474 R      TAD      U/AR
1155 01364 R 341474 R      TAD      U/AR
1156 01365 R 041500 R      DAC      A      /A=2U-B
1157 01366 R 201474 R      LAC      U/AR
1158 01367 R 740001 A      CMA
1159 01370 R 340041 R      TAD      L1
1160 01371 R 041474 R      DAC      U/AR      /SAVE -U FOR COUNT
1161 01372 R 341477 R      TAD      B
1162 01373 R 041501 R      DAC      D      /D=B-U
1163
1164
1165
1166
1167
1168 01374 R 740100 A      SMA
1169 01375 R 601402 R      JMP      APPR6
1170 01376 R 201475 R      LAC      VAR      /V IS LESSER.
1171 01377 R 744010 A      CLL!RAL      /2V
1172 01400 R 340044 R      TAD      L3      /2V+3
1173 01401 R 601406 R      JMP      APPR7
1174 01402 R 750200 A      APPR6  SZA!CLA      /(U-V) IS LESSER
1175 01403 R 200041 R      LAC      L1      /2(U-V)+3, IF D .NE. 0
1176 01404 R 341476 R      TAD      U/AR      /+2, IF D .EQ. 0
1177 01405 R 340052 R      TAD      L2
1178 01406 R 440053 R      APPR7  ISZ      CKND SW      /* IN AC, SKIP IF REAL WORK
1179 01407 R 621137 R      JMP*     /ECTOR      /0, RETURN.
1180 01410 R 040745 R      DAC      TEMP4      /SAVE COUNT
1181 01411 R 201511 R      LAC      LDSKP
1182 01412 R 100352 R      JMS      PUT      /PLACE SKIP OVER COUNT
1183 01413 R 200745 R      LAC      TEMP4      /GET BACK COUNT
1184 01414 R 100352 R      JMS      PUT      /PLACE IT IN LINE
1185
      .EJECT

```

```

1186
1187
1188
1189      01415 R 201501 R
1190      01416 R 741100 A
1191      01417 R 601435 R
1192      01420 R 201063 R
1193      01421 R 540041 R
1194      01422 R 601450 R
1195      01423 R 200100 R
1196      01424 R 101263 R
1197      01425 R 201500 R
1198      01426 R 740001 A
1199      01427 R 340041 R
1200      01430 R 341501 R
1201      01431 R 041501 R
1202      01432 R 200041 R
1203      01433 R 041063 R
1204      01434 R 601455 R
1205      01435 R 201063 R
1206      01436 R 540135 R
1207      01437 R 601450 R
1208      01440 R 200075 R
1209      01441 R 101263 R
1210      01442 R 201477 R
1211      01443 R 341501 R
1212      01444 R 041501 R
1213      01445 R 777777 A
1214      01446 R 041063 R
1215      01447 R 601455 R
1216      01450 R 460067 R
1217      01451 R 460067 R
1218      01452 R 741100 A
1219      01453 R 601442 R
1220      01454 R 601425 R
1221      01455 R 441474 R
1222      01456 R 601415 R
1223      01457 R 777777 A
1224      01460 R 041063 R
1225      01461 R 621137 R
1226
1227
1228
1229

```

```

/
/ APPROXIMATION LOOP
/
APPR20  LAC      D
        SPA
        JMP      APPR25
        LAC      SAMSW
        SAD      L1
        JMP      APPR30
        LAC      MOVE2
        JMS      LSTOR
APPR21  LAC      A
        CMA
        TAD      L1
        TAD      D
        DAC      D
        LAC      L1
        DAC      SAMSW
        JMP      APPRND
APPR25  LAC      SAMSW
        SAD      LM1
        JMP      APPR30
        LAC      MOVE1
        JMS      LSTOR
APPR26  LAC      B
        TAD      D
        DAC      D
        LAW      -1
        DAC      SAMSW
        JMP      APPRND
APPR30  ISZ*     POINT
        ISZ*     POINT
        SPA
        JMP      APPR26
        JMP      APPR21
APPRND  ISZ      UVAR
        JMP      APPR20
        LAW      -1
        DAC      APPRSW
        JMP*     VECTOR
/
        .ENDC
/
        .EJECT

```

```

/IS D LESS THAN ZERO?
/YES.
/GET SAME SWITCH.
/WAS LAST MOVE = MOVE2?
/YES.
/NO. GET MOVE2 VECTOR
/AND STORE IN DISPLAY FILE.

/D=D-A

/SET SAME SWITCH.
/GO CHECK IF DONE.
/D .LT.0, GET SAME SWITCH.
/WAS LAST MOVE = MOVE1?
/YES.
/NO. GET MOVE1 VECTOR.
/STORE MOVE1 VECTOR IN DISPLAY FILE.

/D=D+B

/SET SAME SWITCH
/GO CHECK IF DONE.
/THIS MOVE SAME AS LAST MOVE.
/BUMP EXISTING VECTOR BY 2.

/FINISHED?
/NO. GET NEXT MOVE.
/SET SWITCH TO
/INDICATE APPROXIMATION.

/END OF CONDITIONAL ASSEMBLY

```

```

1230 /
1231 / .IFDEF ARBVEC
1232 /
1233 / SHORTER ROUTINE WHEN ARBVEC GAURANTEED
1234 /
1235 / CALL WITH AC=0 FOR # OF LOC'S RETURNED IN AC
1236 / CALL WITH AC=-1 FOR REAL CODE GENERATION
1237 / CALL WITH AC=-2 (LNS) LIKE AC=0 EXCEPT
1238 / ON 0,0 LINE RETURNS VIA VECTOR. NOT VIA RETAD2
1239 /
1240 VECTOR 0
1241 DAC CKNSW /EXIT CONTROLLER
1242 LAC* LT /INTENSIFIED?
1243 SZA /SKIP IF NOT
1244 LAC (20000 /INTENSITY BIT BOTH INSTR. TYPES
1245 DAC TEMP4 /HOLD
1246 LAC* ARG1 /GET DX
1247 JMS VECSUB /RETURN FULL STROKE VECT INST IN AC
1248 DAC TEMP1 /IN CASE WE DO DO STROKE VECTOR
1249 AND (1777 /KEEP MAGNITUDE
1250 DAC DELTAX
1251 LAC* ARG2 /GET DY
1252 JMS VECSUB /FULL STROKE Y
1253 DAC TEMP2 /AGAIN MAY NEED LATER
1254 AND (1777 /KEEP ONLY MAGNITUDE OF DY
1255 SAD DELTAX /DIAGONAL LINE??
1256 JMP VDIAG /YES, BASIC VECTOR, DIAGONAL
1257 SNA|CLA /SKIP IF DY NON-0
1258 JMP VY0 /BASIC VECTOR, DY=0
1259 SAD DELTAX /IS DELTAX MINUS?
1260 JMP VX0 /BASIC VECTOR, DX=0
1261 LAC L2 /* OF LOC'S IN CASE THAT ALL REQUIRED
1262 VECT2 ISZ CKNSW /SKIP IF REAL CODE TO BE DONE
1263 JMP* VECTOR /NOPE
1264 LAC TEMP1 /EITHER TYPE
1265 JMS PUT /PLACE AN INSTRUCTION
1266 LAC TEMP2 /SECOND STROKE, OR NOTHING
1267 SZA /SKIP ON NOTHING
1268 JMS PUT
1269 JMP* VECTOR /THAT'S IT
1270 /
1271 / VECSUB
1272 /
1273 / CALL WITH DX OR DY IN AC
1274 /
1275 / MAKES THE STROKE VECTOR INSTRUCTION RETURNS IN AC
1276 /
1277 VECSUB 0
1278 SMA|CMA|CLL /SKIP 1, COMPLEMENT, DEFAULT LINK OFF
1279 SKP|CMA|CML /* ORIGINAL, MAKE + AGAIN, 1 LINK BIT
1280 TAD L1 /MAKE TWO'S COMPLEMENT
1281 AND (1777 /KEEP TO 10 BITS

```

```

1282      SNL          /SKIP ON POSITIVE SIGN
1283      TAD          (4000 /SIGN BIT FOR NEGATIVE
1284      TAD          TEMP4 /INTENSITY
1285      TAD          (STRK /STROKE VECTOR OP CODE
1286      JMP*         VECSUB
1287  /
1288  / BASIC VECTOR SECTION
1289  /
1290  /VX0      LAC      TEMP1 /DY=0; BUILD FROM STROKE X
1291          AND      (4000 /SHIFT UP SIGN BIT BY ONE
1292          TAD      TEMP1 /LEAVING ALL ELSE IN PLACE
1293          JMP      BASCOM /COMMON FINISH UP
1294  /
1295  /VDIAG    SNA!CLA   /DIAG; CHECK FOR 0.0 LINE
1296          JMP      ZEROX /IT WAS, SPECIAL EXIT
1297          SNL      /SKIP IF Y POSITIVE
1298          TAD      (14000 /CONJURE UP THE DIRECTION CODE
1299          XOR      TEMP1 /BRING UP MAGNITUDE, INTES, ETC.
1300          TAD      (2000 /MAKE IT A DIAGONAL CODE
1301          JMP      BASCOM /REJOIN
1302  /
1303  /VX0      LAC      TEMP2 /DX=0; BUILD FROM STROKE Y
1304          AND      (4000 /SHIFT UP SIGN BIT BY ONE
1305          TAD      TEMP2 /ALL ELSE THE SAME
1306          TAD      (4000 /DIFFERS FROM DY=0 BY THIS AMOUNT
1307  BASCOM  XOR      LT5  /STROKE OP 100000 TO BASIC OP 400000
1308          DAC      TEMP1 /FULL INSTRUCTION
1309          DZM      TEMP2 /NO SECOND 1
1310          LAC      L1   /ONE INSTRUCTION FOR CKND SW EXIT
1311          JMP      VECT2 /COMMON FINISH UP
1312  /
1313  /      .ENDC
1314  /
1315      .EJECT

```

```

1316
1317
1318
1319
1320 01462 R 000000 A MF.101
1321 01463 R 000000 A MF.102
1322 001463 R MFPTR=MF.102
1323 001462 R MAINFL=MF.101
1324 01464 R 001466 R LARG1 ARG1
1325 01465 R 000000 A ARG0 0
1326 01466 R 000000 A ARG1 0
1327 01467 R 000000 A ARG2 0
1328 01470 R 000000 A ARG3 0
1329 01471 R 000000 A ARG4 0
1330 01472 R 000000 A ARG5 0
1331 01473 R 000000 A ARG6 0
1332 01474 R 000000 A ARG7 0
1333 01475 R 000000 A ARG10 0
1334 01476 R 000000 A ARG11 0
1335 01477 R 000000 A ARG12 0
1336 01500 R 000000 A ARG13 0
1337 01501 R 000000 A ARG14 0
1338 01502 R 000415 R PNAME L6
1339
1340 01503 R 000003 A MASK 3
1341 01504 R 000154 A 154
1342 01505 R 000020 A L20 20
1343 01506 R 000600 A 600
1344 01507 R 200004 A LDNOPE DNOPE
1345 01510 R 200001 A LDNOPB DNOPB
1346 01511 R 235000 A LDSKP DSKP
1347
1348
1349
1350 001466 R RST=ARG1
1351 001466 R STR=ARG1
1352 001466 R DTA=ARG1
1353 001470 R AX=ARG1+2
1354 001467 R N=ARG1+1
1355 001467 R PNAME1=ARG1+1
1356 001470 R LT=ARG1+2
1357
1358
1359
1360 001474 R UVAR=ARG1+6
1361 001475 R VAR=ARG1+7
1362 001476 R UMW=ARG1+10
1363 001477 R B=ARG1+11
1364 001500 R A=ARG1+12
1365 001501 R D=ARG1+13
1366
1367

```

/
 /
 /CONSTANTS AND STORAGE
 /
 /GLOBAL FOR FIRST LOC. OF CURRENT FILE.
 /GLOBL. END OF MAIN 0 IF NOT RUNNG
 /POINTER TO CURRENT END OF DISPLAY FILE.
 /POINTER TO BEGINNING OF CURRENT DISP.FILE.
 /PREVENT FORWARD LITERAL SPACE PROBLEM
 /S+M; GRAPHICS CODE:EDI; 200000 IF PREV. PARAM

 /ADDR OF SUBPIC NAME:L6 IN CASE DINIT
 /IS CALLED BEFORE ANY SUBPIC ROUTINES
 /FOUR MASKS FOR COMPUTING WORDS
 /NEEDED FOR PARAMTER GROUP
 /DASH
 /NAME AND SYNC

 / MNEMONIC EQUATES FOR ARGUMENT FETCH
 /
 /SAVE-RESTORE OPTION FLAG FOR COPY AND PLOT 0.
 /ADDR. OF 5/7 ASCII ARRAY FOR TEXT AND PLOT 3.
 /ADDR. OF DATA POINTS FOR GRAPH.
 /POINTED AXIS FOR GRAPH.
 /* OF DATA POINTS FOR GRAPH,ORCHARS. FOR TEXT.
 /ADDR. OF SUBPIC TO BE COPIED.
 /LINE INTENSITY ARGUMNT
 /
 / FOR LINE APPROX. USE HIGHER ARG'S FOR WORK SPACE
 /
 /MAJOR DELTA (VECTOR APPROX.)
 /MINOR DELTA (VECTOR APPROX.)
 /2(UVAR-VAR) (VECTOR APPROX.)
 /B VARIABLE FOR VECTOR APPROX.
 /A VARIABLE FOR VECTOR APPROX.
 /D VARIABLE FOR VECTOR APPROX.
 /
 .EJECT

```

1368 / ENTRY POINTS ARE ALSO AVAILABLE FOR STORAGE AND TEMP
1369 /
1370
1371 000075 R MOVE1=BLANK /MOVE1 VECTOR FOR APPROX.
1372 000100 R MOVE2=UNBLNK /MOVE2 VECTOR FOR APPROX.
1373 000755 R SOX=CPY /SIGN OF DELTA X (VECTOR SUBR).
1374 000606 R SOY=PRM /SIGN OF DELTA Y (VECTOR SUBR).
1375 000053 R CKNSDW=GRAPH /CHECK-NEED-SWITCH (VECTOR SUBR).
1376 000053 R REGT0=GRAPH /TEMPORARY (PRM SUBR)
1377 000064 R REGT1=DELETE /TEMPORARY (PRM SUBR)
1378 000522 R REGT2=CHR /TEMPORARY (PRM SUBR)
1379 001063 R SAMSW=PNT /SAME-SWITCH (VECTOR APPROX.)
1380 001063 R APPRSW=PNT /APPROX. SWITCH 8(VECTOR SUBR).
1381 000064 R DELTAX=DELETE /DELTA X (VECTOR SUBR).
1382 000522 R DELTAY=CHR /DELTA Y (VECTOR SUBR).
1383 000067 R POINT=PLT /RUNNING POINTER TO PLACE CODE IN FILE
1384 / /START: POINT=POINTS M+S:POINT=POINTS-1 E.
1385 000072 R POINTS=REPLT /ADDR. OF BEGINNING OF GROUP, UNCHANGED!
1386 000037 R HOLD=LINE /EDIT:GROUP DNOP COUNT
1387 000042 R FILE=TEXT /PRESENT FILE FOR SUB+MAIN
1388 000106 R XOPT=LNSIZE /XCT POINTER OPTIONAL ARGUMENT
1389 000045 R XNEED=COPY /XCT POINTER NEEDED # OF LOC'S
1390 000103 R XCODE=DINIT /XCT POINTER FOR CODE GENERATE
1391 000050 R RETAD2=PRAMTR /COMMON RETURN.
1392 000111 R TYPE=DCLOSE /SUB, MAIN, EDIT FLAG
1393 000056 R PUTCNT=SETPT /SET TO -1 TO FORCE CALL TO PUT
1394 / /TO PLACE INSTR. IN LOC. 'HOLD'
1395 000352 R POINT1=PUT /EDIT, ADDITIONAL GROUP POINTER
1396 000061 R POINT2=RSETPT /EDIT, ADDITIONAL GROUP POINTER
1397 /
1398 .EJECT

```

1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431

/
/ NOTES ON TREATMENT OF PARAMETER GROUPS
/
/ PARAMETER GROUPS HAVE A VARIABLE SIZE, SO THAT DELIMITERS
/ DNOPB (BEGIN) AND DNOPE (END) HAVE BEEN INVENTED TO HELP
/ COUNT PARAMETER GROUPS FOR EDITS (RELOTS).
/
/ ON CODE GENERATION (SUBPIC + MAIN (PLOT)) , NO DNOPB'S ARE PLACED.
/ A DNOPE IS PLACED ONLY IF CNAME IS SPECIFIED, I.E. THE GROUP
/ CAN BE SUBJECTED TO A FUTURE EDIT.
/
/ ON CODE EDITTING
/
/ ANYTHING OTHER THAN A PARAMETER OVER A PARAMETER WILL RECLAIM ANY
/ DNOPB OR DNOPE ASSOCIATED WITH THAT GROUP.
/
/ GENERAL PARAMETER OVER ANYTHING WILL REQUIRE A DNOPB IF THE
/ PREVIOUS GRAPHICS INSTRUCTION IS PART OF A PARAMETER GROUP, AND
/ NOT A DNOPE OR DNOP. SIMILARLY, IF THE FOLLOWING GRAPHICS INSTRUCC.
/ IS A PARAMETER AND NOT A DNOPB OR DNOP, AND FOLLOWING DNOPE
/ WILL BE REQUIRED. THE RELOT CAN FAIL BECAUSE OF INSUFFICIENT SPACE
/ FOR THESE DELIMITERS.
/
/ SPECIAL CASE PARAMETER OVER PARAMETER WHEN THERE IS NO LEADING DNOPB,
/ WILL REQUIRE NO LEADING DNOPB. THIS ALLOWS A EDITABLE PARAMETER GROUP
/ TO FOLLOW AN NON-EDITABLE PARAMETER GROUP WITHOUT COSTING A DNOP.
/ THIS WILL HOLD AS LONG AS THE PARAMETER GROUP IS NOT CHANGED TO
/ ANYTHING ELSE!
/
/ THESE CONVENTIONS MINIMIZE THE IMPACT ON THE DISPLAY FILE SIZE OF
/ EXISTING PROGRAMS.
/
/

.END

000000 A
01512 R 007777 A *L
01513 R 001777 A *L
01514 R 000410 R *L
01515 R 017777 A *L
01516 R 620000 A *L
01517 R 001503 R *L
01520 R 060002 A *L
01521 R 520001 A *L
01522 R 000741 R *L
01523 R 000004 A *L
01524 R 000400 A *L
01525 R 001000 A *L
01526 R 000040 A *L
01527 R 000177 A *L
01530 R 002000 A *L
01531 R 660002 A *L
01532 R 720001 A *L
01533 R 240004 A *L
01534 R 417777 A *L

01535 R 720002 A *L
01536 R 020000 A *L
01537 R 004000 A *L
01540 R 162000 A *L
01541 R 760000 A *L
01542 R 140000 A *L
01543 R 144000 A *L
01544 R 215052 A *L
01545 R 620001 A *L
01546 R 010000 A *L
01547 R 100000 A *L
01550 R 770000 A *L
01551 R 410000 A *L
01552 R 414000 A *L
01553 R 402000 A *L
01554 R 400002 A *L
01555 R 402002 A *L

SIZE=01556

NO ERROR LINES

A	001500	1156	1197	1364*					
ALTPNT	00605	640	667*						
APPRND	01455	1204	1215	1221*					
APPROX	01273	1008	1091*						
APPRSW	001063	981	1224	1380*					
APPR20	01415	1189*	1222						
APPR21	01425	1197*	1220						
APPR25	01435	1191	1205*						
APPR26	01442	1210*	1219						
APPR30	01450	1194	1207	1216*					
APPR5	01313	1096	1107*						
APPR6	01402	1169	1174*						
APPR7	01406	1173	1178*						
ARG0	01465	280	282	324	475	559	1325*		
ARG1	01466	106	527	547	576	611	682	690	691
		859	863	876	880	883	911	921	982
		989	1246	1324	1326*	1350	1351	1352	1353
		1354	1355	1356	1360	1361	1362	1363	1364
		1365							
ARG10	01475	1333*							
ARG11	01476	1334*							
ARG12	01477	1335*							
ARG13	01500	1336*							
ARG14	01501	1337*							
ARG2	01467	821	831	842	851	907	1251	1327*	
ARG3	01470	100	103	104	854	954	1328*		
ARG4	01471	101	105	1329*					
ARG5	01472	1330*							
ARG6	01473	1331*							
ARG7	01474	1332*							
AX	001470	1353*							
B	001477	1151	1161	1210	1363*				
BK	001000	47*	758						
BLANK	00075	70	210*	1371					
BLINK	00711	721	756*						
BLN	01030	128	871*						
CHARSI	060000	43*	607	609					
CHR	00522	123	604*	624	651	1378	1382		
CKND5W	000053	957	958	980	1010	1034	1178	1241	1262
		1375*							
CODE	00013	120*							
COPY	00045	70	186*	1389					
CPY	00755	120	820*	833	1373				
CPY1	00773	826	835*						
CPY2	00771	832*	847						
D	001501	1162	1189	1200	1201	1211	1212	1365*	
DASH	00715	722	763*						
DATVT	000010	64*	76	226	920				
DCLOSE	00111	71	224*	227	1392				
DELETE	00064	71	201*	1377	1381				
DELTAX	000064	992	999	1016	1026	1031	1091	1100	1107
		1250	1255	1259	1381*				
DELTAY	000522	985	996	1002	1022	1042	1094	1097	1110

DINIT	00103	71	216*	1390						
DJMP	600000	37*	609	829	840					
DJMPI	620000	38*	463	888	837					
DJMSI	660000	39*	827	829	838	840				
DNOP	200000	53*	809							
DNOPB	200001	54*	1345							
DNOPE	200004	55*	814	1344						
DNT	01075	127	919*							
DSH	000004	48*	764							
DSKP	235000	59*	1346							
DTA	001466	1352*								
DTAB	01247	1048	1049	1051	1052*					
EDICLR	00337	418*	421							
EDIT	00265	293	366*							
ENTRES	000013	67*	148							
EXX0	00345	427*	900							
EXX1	00347	429*	943							
EXX2	00350	345	430*	901						
FILE	000042	302	309	312	342	344	353	1387*		
GETARG	00116	181	184	187	190	193	196	199	202	
		205	208	211	214	217	221	246*	249	
		260	278	290						
GRAPH	00053	70	192*	1375	1376					
GRPH	01010	125	851*							
GRPH5	01020	859*	865							
GTRG03	00141	267*	276							
GTRG04	00146	269	272*							
GTRG05	00150	265	274*							
GY	162000	62*	857							
HOLD	000037	336	450	1386*						
INT	002000	45*								
INTENS	00674	719	738*							
JINST	00655	696	714*							
JOINSM	00205	307*	359							
LARG1	01464	263	525	676	1324*					
LDNOP	00750	381	384	399	417	478	485	560	565	
		568	589	592	810*					
LDNOPB	01510	480	557	587	1345*					
LDNOPE	01507	321	482	563	1344*					
LDSKP	01511	379	1181	1346*						
LENT	00035	148*	285	287						
LINE	00037	70	180*	1386						
LM1	00135	262*	316	373	1206					
LM2	00533	620*	632							
LM20	00601	663*								
LNOP	00021	126*	383	484	567	591				
LNS	01127	130	950*							
LNSIZE	00106	75	220*	1388						
LODN	000400	51*	785							
LOPT	00036	149*	283							
LP	000010	46*								
LSD	703004	66*	429							

LSTOR	01263	1018	1024	1037	1073*	1080	1196	1209	
LT	001470	1075	1242	1356*					
LTEMP2	00033	143*	307	874					
LTPN	00703	720	746*						
LT5	00604	666*	1307						
L0	00047	188*							
L1	00041	182*	771	1033	1068	1093	1129	1145	1153
		1159	1175	1193	1199	1202	1280	1310	
L10	00077	212*	739	748					
L11	00102	215*							
L12	00110	222*	637						
L17	00602	664*	667	730					
L2	00052	142	191*	772	877	925	1009	1177	1261
L20	01505	731	778	1342*					
L3	00044	141	185*	763	939	1172			
L5	00055	194*	629						
L6	00415	501*	549	700	1338				
L7	00105	218*	738						
L77777	00603	613	665*						
MAINFL	001462	301	354	922	927	935	940	941	1323*
MAINIT	00175	299*							
MANCOM	01123	929	941*						
MASK	01503	572	1340*						
MFPTR	001463	225	299	340	356	427	942	1322*	
MF.101	01462	75	1320*	1323					
MF.102	01463	75	1321*	1322					
MOVE1	000075	1124	1208	1371*					
MOVE2	000100	1138	1195	1372*					
MRETUR	00237	334*	866						
MSTOR5	00234	325	331*	465					
M1TAB	01325	1105	1106	1115	1116	1120*			
M2TAB	01343	1128	1131	1133	1134*				
N	001467	622	1354*						
NAME	00731	725	784*						
NEDCOP	00445	138	547*						
NEDFRN	00457	558	562*						
NEDPAR	00451	140	556*						
NEDP1	00467	561	564	566	570*				
NEDP2	00475	576*	582						
NEDP3	00517	588	590	594*					
NEEDS	00026	138*							
NEWFIL	00362	319	462*						
NRETUR	00330	411*	550	596					
OFSET	00721	723	770*						
OFST	000002	49*							
OPT	00000	100*	149						
OPTPAR	00430	102	522*	534					
OPTP1	00436	528*	530						
OPTP2	00437	529*	532						
PARBAK	00746	305	523	593	594	805*			
PARFRN	00741	524	569	570	698	800*			
PARLOP	00643	704*	712						
PARL1	00650	706	709*						

PAROPS	00747	809*							
PLOT	00067	71	204*	1383					
PNAME	01502	351	352	1338*					
PNAME1	001467	1355*							
PNT	01063	124	906*	915	1379	1380			
POINT	000067	313	326	329	338	374	394	411	453
		454	562	923	1216	1217	1383*		
POINTS	000072	314	330	334	337	372	378	397	464
		556	583	605	823	1385*			
POINT1	000352	317	320	327	328	370	418	419	1395*
POINT2	000061	371	390	391	395	403	404	413	472
		476	477	489	1396*				
PRAMTR	00050	70	189*	1391					
PRM	00606	122	675*	713	1374				
PRMTBL	00656	692	714	718*					
PRM1	200000	52*	810						
PRM10	00616	683*	695						
PRM15	00627	692*	694	697					
PRM2	210000	56*	811						
PRM20	00630	687	693*	734	751	766	788		
PRM25	00633	684	696*						
PRM3	220000	57*	812						
PUT	00352	335	447*	451	455	608	610	615	708
		828	830	832	837	839	841	843	844
		862	910	914	924	926	934	938	1079
		1182	1184	1265	1268	1395			
PUTCNT	000056	332	398	420	448	1393*			
PUTPUT	00357	449	453*						
PX	144000	35*	913						
PY	140000	34*	909						
REGT0	000053	685	693	699	704	709	1376*		
REGT1	000064	677	688	689	703	711	1377*		
REGT2	000522	701	707	710	1378*				
REPLOT	00072	71	207*	1385					
REST	260000	61*	846						
RETAD2	000050	252	393	415	431	882	889	955	959
		1391*							
ROT	000040	50*	779						
ROTAT	00725	724	777*						
RSETPT	00061	71	198*	1396					
RST	001466	822	1350*						
RS3	703142	63*	1005						
SAMSW	001063	1192	1203	1205	1214	1379*			
SAVE	240000	60*	835	838	846				
SC	000020	44*							
SCALE	00667	718	730*						
SETPT	00056	71	195*	1393					
SHUNT	00244	339*							
SICW	01126	920	945*						
SIGNM	01255	983	990	1064*	1066	1069			
SIZAPP	00366	380	472*						
SIZETB	00410	389	496*						
SIZJ1	00316	395*	490						

SIZLOO	00323	403*	405						
SIZND	00404	479	481	487*					
SIZND1	00405	483	488*						
SIZPAR	00370	385	475*						
SIZPRL	00371	476*	486						
SIZTST	00312	391*	473						
SKP2	234000	58*	813						
SOX	000755	995	1013	1028	1045	1113	1125	1373*	
SOY	000606	988	1019	1039	1047	1050	1103	1126	1132
		1374*							
STPD	703044	65*	416	896	919				
STR	001466	627	628	636	642	647	648	1351*	
STRK	100000	36*	1017	1023	1285				
SUBPC	00253	295	349*						
SYNC	00736	726	792*						
TEMP1	00742	264	272	273	277	368	369	526	528
		533	585	586	678	732	733	801*	836
		845	858	861	878	885	891	1032	1036
		1074	1078	1248	1264	1290	1292	1299	1308
TEMP2	00743	143	270	271	573	577	580	679	749
		750	802*	853	864	879	890	898	1253
		1266	1303	1305	1309				
TEMP3	00744	258	275	575	581	641	643	644	645
		646	680	765	803*	875	884	886	897
TEMP4	00745	250	251	253	255	267	274	308	315
		571	579	584	595	621	639	649	681
		786	787	804*	872	899	1180	1183	1245
		1284							
TEXT	00042	70	183*	1387					
TST1	00541	627*	631						
TST2	00543	625	629*						
TST3	00553	635	638*						
TST4	00557	642*	650						
TYPE	000111	248	300	339	357	1392*			
T1R	00671	732*	742						
T2R	00706	749*	759	773	780				
T4R	00733	786*	794						
UBL	01031	129	872*						
UMV	001476	1148	1176	1362*					
UNBLNK	00100	70	213*	1372					
UVAR	001474	1099	1109	1146	1154	1155	1157	1160	1221
		1360*							
VAR	001475	1102	1112	1143	1149	1170	1361*		
VECTOR	01137	121	139	953	960	979*	1011	1025	1035
		1038	1057	1179	1225	1240*	1263	1269	
V1	402000	40*	1055						
V4	410000	41*	1030						
V6	414000	42*	1041						
XCODE	000103	286	333	426	1390*				
XNEED	000045	288	410	1389*					
XOPT	000106	284	303	349	366	1388*			
XZERO	01232	1001	1039*						
YEQX	01237	1003	1044*						

YEQX5	01245	1046	1050*	
YZERO	01215	998	1026*	
YZER5	01222	1031*	1043	1056
ZEROX	01133	957*	1027	1296
.CLEAR	MACRO			
.CLOSE	MACRO	226		
.DELETE	MACRO			
.ENTER	MACRO			
.EXIT	MACRO			
.FSTAT	MACRO			
.GET	MACRO			
.GTBUF	MACRO			
.GVBUF	MACRO			
.INIT	MACRO	920		
.MTAPE	MACRO			
.MTRAN	MACRO			
.OVLRA	MACRO			
.PUT	MACRO			
.RAND	MACRO			
.READ	MACRO			
.RENAM	MACRO			
.RTRAN	MACRO			
.SEEK	MACRO			
.SETUP	MACRO			
.SYSID	MACRO			
.TIMER	MACRO			
.TRAN	MACRO			
.USER	MACRO			
.WAIT	MACRO			
.WAITR	MACRO			
.WRITE	MACRO			
.X/MOF	MACRO			
.X/MON	MACRO			