

IDENTIFICATION

Product Code: DEC-08-A2D2-LA

Product Name: Listing of 8K SABR Assembler

Date Created: October 30, 1969

/SABR,815

PAL14

V135

23-SEP-69

11128

PAGE 1

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

/SABR,815

/SABR,V16

/DEC-28-A2D2-16

/COPYRIGHT 1969, DIGITAL EQUIPMENT CORP.,

/MAYNARD, MASS,

/

/NOTE: WHENEVER ABOVE VERSION NUMBER IS CHANGED

/BE SURE TO ALSO CHANGE VERSION NUMBER FOR TYPEOUT.

/THIS IS AT VERSN\*13 (ABOUT P, 83)

/

11 /THERE ARE TWO BASIC PHASES OF OPERATION WITHIN  
12 /SABR (A) COLLECTION AND (B) ASSEMBLY, IN PASS 1  
13 /SABR COLLECTS A FULL PAGE OF DATA AND THEN  
14 /ASSEMBLES THE FULL PAGE, IN PASS2 COLLECTION  
15 /AND ASSEMBLY ARE CARRIED OUT ON A LINE-  
16 /BY-LINE BASIS RATHER THAN PAGE-BY-  
17 /PAGE, FOLLOWING IS A DESCRIPTIVE FLOW CHART OF  
18 /THE PRINCIPAL METHODS OF OPERATION USED IN  
19 /THE PROGRAM DURING PASS1.  
20 / (1) BEGINNING AT START THERE ARE TWO ROUTINES NECESSARY  
21 /TO INITIALIZE THE ENTIRE PROGRAM, IOINIT CARRIES  
22 /OUT THE DIALOG WITH THE USER TO DETERMINE WHICH  
23 /I/O DEVICES WILL BE USED, INITA INITIALIZES ALL  
24 /THE FLAGS AND TABLES WHICH ARE USED CONTINUOUSLY  
25 /THROUGHOUT THE PROGRAM,  
26 / (2) THE DRIVER FOR THE FULL PAGE-BY-PAGE ASSEMBLER  
27 /IS CONTAINED IN THE LOOP THAT RUNS FROM RSTR1  
28 /TO RSTR6=1, THIS LOOP OPERATES AS FOLLOWS,  
29 /FIRST IT CALLS INILPT WHICH INITIALIZES ALL  
30 /THOSE FLAGS AND TABLES WHICH MUST BE  
31 /REFRESHED OR REBUILT FOR EACH PAGE OF CODE,  
32 /THEN IT CALLS THE MAIN LINE-BY-LINE  
33 /COLLECTION LOOP (WHICH IS DESCRIBED IN ITEM 3),  
34 /WHEN A FULL PAGE OF CODE HAS BEEN COLLECTED  
35 /THE DRIVER THEN CALLS L55 TO ASSEMBLE THE  
36 /PAGE (SEE ITEM 8),  
37 / (3) THE COLLECTION LOOP RUNS FROM RSTR1 THROUGH  
38 /THE CODE AT RSTR6, THIS LOOP FIRST CALLS  
39 /INCPT WHICH PREPARES FLAGS AND INCREMENTS  
40 /TABLE POINTERS FOR EACH LINE OF CODE, IT  
41 /THEN CALLS THE LINE DECODER DCIL (SEE  
42 /ITEM 4) FOLLOWED BY SETCT, THE ROUTINE WHICH  
43 /INCREMENTS THE PAGE COUNTERS AS REQUIRED  
44 /FOR THE GIVEN LINE (SEE ITEM 6), THEN  
45 /THE COLLECTIONS LOOP PROCEEDS TO INTERPRET  
46 /THE DATA LEFT BY DCIL AND STORE IT, PROPERLY  
47 /CODED, ON THE PAGE TABLE, IF THERE WAS  
48 /A TAG ("LFS" FOR "LOCATION FIELD SYMBOL") IT  
49 /IS NECESSARY TO CALL RECT FOR A PAGE RECOUNT,  
50 / (SEE ITEM 7), THEN THE SIZE OF THE PAGE SO  
51 /FAR COLLECTED IS TALLYED UP BY CPGES, IF  
52 /IT IS STILL ,LE, 200, EVERYTHING IS FINE AND WE  
53 /RUN THROUGH THE LOOP AGAIN, IF NOT WE  
54 /FIRST SAVE (USING PUSHIN)  
55 /ALL THE KEY INFORMATION ABOUT THE LINE WHICH  
56 /CAUSED THE OVERFLOW AND THEN EXIT FROM THE  
57 /COLLECTION LOOP TO ASSEMBLE THE PAGE,  
58 / (4) CERTAIN NOTES ABOUT DCIL MAY BE HELPFUL, THIS  
59 /ROUTINE CONTROLS INPUT OF THE SOURCE, INDEV  
60 / (SET BY IOINIT) POINTS TO THE PROPER INPUT  
61 /ROUTINE, HSR OR ASR, THESE ROUTINES  
62 /READ ONE CHARACTER AT A  
63 /TIME FROM THE INPUT DEVICE, THE ROUTINE CALLED  
64 /R DRAWS CHARACTERS ONE AT A TIME FROM THE  
65 /INPUT BUFFER (DATA), WHEN THIS BUFFER IS

/SABR,815

PAL10

V135

23-SEP-69

11128

PAGE 2-1

66  
67  
68  
69  
70

/EMPTY R REFILLS IT USING \*INDEV, FETCH  
/USES R TO EXTRACT ONE CHARACTER AT A TIME FROM  
/THE INPUT BUFFER AND DOES SOME PRELIMINARY  
/SCREENING. RLN USES FETCH TO READ A  
/FULL LINE OF CODE INTO THE LINE BUFFER.

71  
72 /L65 READS ONE CHARACTER AT A TIME FROM  
73 /THE LINE BUFFER, GTSYM READS THE LINE  
74 /ITEM-BY-ITEM, IF THE ITEM IS A SYMBOL, GTSYM  
75 /CALLS SRSYM TO LOOK UP THE ITEM IN THE  
76 /MAIN SYMBOL TABLE OR ENTER IT IF IT IS NOT  
77 /ALREADY THERE(SEE ITEM 5).  
78 /\*\*IMPORTANT\*\*  
79 /WHEN A SYMBOL HAS BEEN PLACED ON THE SYMBOL  
80 /TABLE THE ADDRESS OF THE FIRST WORD OF THE  
81 /ENTRY IS RETURNED AT "SYMBOL," THIS ADDRESS  
82 /IS UNIQUE FOR EACH SYMBOL AND IN THE  
83 /RANGE 2000-7575, THIS NUMBER IS USED  
84 /THROUGHOUT THE PROGRAM AS THE IDENTIFIER FOR  
85 /THIS SYMBOL, \*\* DCIL CONSIDERS EACH ITEM  
86 /OF THE LINE AND ACTS APPROPRIATELY, FOR  
87 /STANDARD INSTRUCTIONS A STRING OF KEY DATA ABOUT  
88 /THE LINE IS LEFT, IF THE LINE IS A PSUEDO-OP  
89 /DCIL WILL IMMEDIATELY CALL THE APPROPRIATE PSUEDO-OP  
90 /HANDLER TO TAKE ALL NECESSARY ACTION, MOST  
91 /OF THE PSUEDO-OP HANDLERS RETURN TO THE  
92 /BEGINNING OF DCIL WHERE THE NEXT LINE CAN BE  
93 /PROCESSED AS IF NOTHING UNUSUAL HAPPENED, THE  
94 /EXCEPTIONS TO THIS ARE THOSE PSUEDO-OPS WHICH  
95 /CAUSE A PREMATURE PAGE ASSEMBLY,  
96 /THE ROUTINE SKIPL IS ACTUALLY A SMALL PART OF  
97 /DCIL, IT HAS TWO PURPOSES, ONE, IT WATCHES  
98 /FOR LINES WHICH SHOULD BE IGNORED BECAUSE THE  
99 /FORTR PSUEDO-OP IS IN EFFECT, SECONGLY IT  
100 /MUST WATCH FOR SEMI-COLONS SO THAT  
101 /IF ONE IS ENCOUNTERED(OUTSIDE A COMMENT)  
102 /THE REMAINDER OF THE LINE CAN BE SAVED FOR  
103 /PROCESSING AS THE "NEXT" LINE.  
104 /5) ONLY TWO MAIN ROUTINES SRSYM AND  
105 /OBSYM, TOGETHER WITH THEIR SUBSIDIARYS RUSVL AND SUSVL  
106 /MAY DIRECTLY CONTACT THE MAIN SYMBOL TABLE,  
107 /THESE ROUTINES COMMUNICATE WITH THE REST OF  
108 /THE PROGRAM THROUGH FOUR IMPORTANT  
109 /CELLS IN PAGE 0:  
110 /USE CONTAINS THE CODE WORD FOR THE SYMBOL ENTRY,  
111 /VAL CONTAINS THE VALUE OF THE SYMBOL,  
112 /SYMBOL CONTAINS THE ADDRESS OF THE FIRST WORD OF THE  
113 /ENTRY(NAMELY THE CODE WORD),  
114 /VALPTR CONTAINS THE ADDRESS OF THE VALUE WORD  
115 /OF THE ENTRY,  
116 /SRSYM, AFTER LOCATING A GIVEN SYMBOL IN THE TABLE  
117 / (OR ENTERING IT IF NECESSARY)  
118 /CALLS SUSVL TO FILL THE FOUR CELLS WITH THE  
119 /PROPER INFORMATION ABOUT THE SYMBOL.  
120

121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169

/OBSYM USES A SYMBOL IDENTIFIER TO GET  
/THE FOUR ESSENTIAL BITS OF INFORMATION, AGAIN  
/CALLING SUSVL TO DO THE WORK, HOWEVER  
/BEFORE EITHER SRSYM OR OBSYM DO ANYTHING  
/THEY BOTH MAKE USE OF RUSVL. RUSVL IS A  
/VERY IMPORTANT ROUTINE. HERE IS HOW IT WORKS.  
/LET US SUPPOSE THAT THE PROGRAM HAS OBTAINED  
/USE, VAL, SYMBOL AND VALPTR FOR A GIVEN  
/SYMBOL (USING SRSYM OR OBSYM), FURTHER, LET  
/US SUPPOSE THAT THE PROGRAM WISHES TO  
/MODIFY BOTH OR EITHER OF THE CODE AND VALUE  
/WORDS FOR THIS SYMBOL IN THE SYMBOL TABLE.  
/THE PROGRAM DOES NOT DIRECTLY ACCESS THESE  
/WORDS IN THE SYMBOL TABLE, INSTEAD THE  
/PROGRAM MERELY MAKES THE DESIRED MODIFICATIONS  
/TO USE AND VAL, NOW SYMBOL AND VALPTR  
/ARE THE POINTERS FOR STORING THIS NEW INFORMATION  
/BACK IN THE TABLE. IT IS VERY IMPORTANT THAT  
/NO PART OF THE PROGRAM EXCEPT SRSYM AND OBSYM  
/EVER MODIFY SYMBOL OR VALPTR, AND BEFORE  
/EITHER OF THESE MODIFY THEM THEY ALWAYS CALL  
/RUSVL. RUSVL TAKES USE AND VAL  
/INCLUDING ANY MODIFICATIONS THAT HAVE BEEN  
/MADE TO THEM AND STORE THESE WORDS BACK  
/IN THE TABLE USING THE STILL UNCHANGED POINTERS  
/SYMBOL AND VALPTR. IN THIS WAY MODIFICATIONS  
/TO THE SYMBOL TABLE ARE MADE IN TWO STAGES.  
/THE FIRST STAGE CONSISTS OF A SIMPLE REFERENCE  
/TO ONE OF TWO PAGE # LOCATIONS, AND THE  
/SECOND STAGE IS TAKEN CARE OF AUTOMATICALLY  
/DURING FURTHER OPERATION OF THE PROGRAM.  
/6)SETCT AND CPGES DEAL WITH FIVE SEPARATE PAGE  
/COUNTERS, THE SUM OF THESE IS THE NUMBER  
/OF WORDS OF CORE NECESSARY TO ASSEMBLE THE CURRENT  
/COLLECTED DATA. PTSZE (PAGE TABLE SIZE) IS THE  
/NUMBER OF ITEMS - CONSTANTS, ADDRESS PARAMETERS  
/AND INSTRUCTIONS - WHICH HAVE BEEN SO FAR  
/COLLECTED. LTSZE IS THE NUMBER OF DISTINCT LITERALS  
/WHEN ARE REQUIRED ON THE CURRENT PAGE. PGEESC  
/WILL BE EITHER 2 OR 4. IT IS  
/2 IF THE LAST COLLECTED INSTRUCTION WAS NOT A  
/SKIP INSTRUCTION, 4 OTHERWISE. THESE ARE THE  
/NUMBER OF WORDS REQUIRED FOR THE PAGE  
/ESCAPE, THIS ITEM IS IGNORED WHEN THE AUTO-  
/MATIC PAGING SWITCH IS NON-ZERO. OPSCTR  
/IS THE NUMBER OF POINTERS TO OFF-PAGE SYMBOLS  
/WHICH ARE REQUIRED ON THE CURRENT PAGE.

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  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222

/THIS ITEM IS DETERMINED BY USE OF THE  
/PAGE SYMBOL TABLE, TWO TYPES OF SYMBOLS  
/ARE STORED ON THIS TABLE: TAGS(LFS'S) ON THE  
/CURRENT PAGE AND SYMBOLS WHICH ARE REFERENCED  
/BY MEMORY REFERENCE INSTRUCTIONS(AFS'S) ON THE  
/CURRENT PAGE, IN THIS TABLE SABR KEEPS TRACK OF  
/WHETHER THE SYMBOL IS ON-PAGE(I.E. IF IT OCCURS AS  
/A TAG ON THE PAGE) AND WHETHER IT HAS BEEN  
/REFERENCED EITHER SIMPLY OR WITH A NUMBER SIGN  
/(MEANING <SYM>\*1), IF THE SYMBOL IS OFF-PAGE  
/AND HAS BEEN REFERENCED ON THE PAGE, ONE POINTER  
/IS REQUIRED ON THE ASSEMBLED PAGE FOR  
/EACH TYPE OF REFERENCE USED (SIMPLE OR #),  
/IN ADDITION CERTAIN INFORMATION REGARDING OBACKR  
/IS KEPT IN THE P.S.T, OBACKR KEEPS COUNT OF  
/THE NUMBER OF EXTRA INSTRUCTIONS WHICH MUST  
/BE GENERATED ON THE CURRENT PAGE, THESE  
/INCLUDE COF'S TO CURRENT BANK (CODE#5 6201'S),  
/CDF 00'S FOR REFERENCE TO COMMON, /(\*)  
/JMS COFSK/SKP PAIRS FOR CDF CUR'S FOLLOWING  
/SKIP INSTRUCTIONS, AND JMS COZSK/SKP PAIRS FOR  
/CDF 00'S FOLLOWING SKIP INSTRUCTIONS, SUCH  
/CDF'S ARE NEEDED FOR OFF-PAGE REFERENCES WHENEVER  
/THE BANK REFERENCED IS NOT THE SAME AS PREVIOUSLY  
/0 INSTEAD OF CURRENT#1 OR VICE-VERSA) OR WHEN  
/THE BANK IS UNKNOWN (#=1) AS AFTER A TAG,  
/AT THE START OF A PAGE, OR FOR ALL JMS'S,  
/OBACKR ALSO KEEPS COUNT OF EXTRA INSTRUCTIONS  
/NEEDED TO GENERATE OFF-PAGE INDIRECT REFERENCES,  
/FOR THESE EITHER 2 OR 4 EXTRA INSTRUCTIONS  
/ARE NEEDED DEPENDING ON WHETHER OR NOT THE  
/PREVIOUS INSTRUCTION WAS A SKIP INSTRUCTION,  
/IN THE PST AN UP-TO-DATE RECORD IS KEPT OF THE  
/NUMBER OF INCREMENTS TO OBACKR SPECIFICALLY  
/DUE TO EACH OFF-PAGE SYMBOL, IN VIEW  
/OF THE RECOUNT PROCEDURE DESCRIBED IN ITEM 7  
/IT WOULD SEEM THAT THIS INFORMATION IS REDUNDANT  
/AND UNNECESSARY, HOWEVER, DURING THE DEBUGGING  
/STAGE OF THE PROGRAM WITH PASS 2 INCLUDED  
/I ENCOUNTERED SEVERAL SITUATIONS, WHICH I FIND  
/VERY DIFFICULT TO DESCRIBE, WHERE MORE IMMEDIATE  
/INFORMATION ABOUT OBACKR WAS NEEDED, I  
/AM NOT EVEN COMPLETELY SURE I UNDERSTAND WHY,  
/OBACKR MUST BE WATCHED CLOSELY, AT 6652 IN THE  
/PROGRAM THERE IS SOME CODE TO ASSIST IN  
/DEBUGGING THE PROGRAM IF PROBLEMS ARISE WITH  
/THE PAGE COUNT,  
/WHENEVER A NEW TAG IS ENCOUNTERED ON A  
/PAGE, SETCT USES CPLFS TO REDUCE  
/BOTH OPSCTR AND OBACKR APPROPRIATELY  
/SINCE WHAT PREVIOUSLY WERE OFF-PAGE REFERENCES  
/MAY NOW HAVE BECOME ON-PAGE REFERENCES.

```
223
224
225 / (7) WHENEVER A TAG IS ENCOUNTERED ON A GIVEN
226 / PAGE RECT IS CALLED TO GO THROUGH THE
227 / ENTIRE CURRENT PAGE TABLE AND RECOUNT
228 / THE PAGE, IT DOES THIS BY CALLING SETCT
229 / AGAIN, ONCE FOR EACH ITEM ON THE PAGE TABLE.
230 / THE ONLY THING REALLY ACCOMPLISHED HERE
231 / IS THAT OBACKR IS RESET, OPSCTR IS
232 / UNCHANGED AS WELL AS THE OTHER PAGE COUNTERS,
233 / THE NEED TO RECOUNT OBACKR IS SHOWN
234 / BY THE FOLLOWING EXAMPLE:
235 /
236 /   A, TAD B
237 /   TAD C
238 /   B, 0
239 /   <PAGE FILLS UP>
240 /   C, 0
241 / NOW BECAUSE BANK IS UNKNOWN AFTER "A,"
242 / AND B IS UNDEFINED AS YET, "TAD B"
243 / REQUIRES A CDF CUR, HENCE INCREMENT OBACKR.
244 / "TAD C" IS OFF PAGE TOO, BUT REQUIRES NO
245 / CDF SINCE IT IS IN THE SAME BANK.
246 / HOWEVER WHEN B IS DEFINED ON PAGE, THE
247 / CDF IT CAUSED IS NO LONGER NECESSARY,
248 / BUT NOW THE "TAD C" REQUIRES A CDF.
249 / (8) THE ROUTINE L55 CAUSES EACH PAGE TO BE ASSEMBLED,
250 / DURING PASS 1 THERE ARE TWO SUB-PHASES TO THE ASSEMBLY.
251 / FOR THE MOST PART BOTH PHASES RUN THROUGH THE
252 / ENTIRE PAGE OF COLLECTED CODE USING THE
253 / SAME ASSEMBLY ROUTINES, THE DIFFERENCE IS
254 / THAT DURING PHASE 1 (ACTR=0)
255 / ACTUAL OUTPUT IS SUPPRESSED, THE KEY
256 / PURPOSE OF PHASE 1 IS TO DEFINE ALL THE
257 / TAGS THAT OCCUR ON THE PAGE, CLEARLY THE
258 / TAGS COULD NOT BE DEFINED DURING COLLECTION
259 / BECAUSE AT THAT POINT WE WERE NOT SURE
260 / WHAT SYMBOLS EVEN WERE ON PAGE, AND THUS
261 / NOT SURE HOW MANY EXTRA INSTRUCTIONS
262 / WOULD BE NECESSARY, THUS SUB-PHASE 1 OF THE
263 / ASSEMBLY IS REQUIRED SO THAT IN THE
264 / SECOND PHASE OF THE ASSEMBLY ON-PAGE
265 / FORWARD REFERENCES CAN BE RESOLVED, HENCE
266 / L55 CALLS THE ASSEMBLY ROUTINE ASMBL
267 / TWICE FOR EACH PAGE OF CODE.
268 / (9) AFTER INITIALIZING THE VARIOUS PAGE TABLE POINTERS
269 / ASMBL GOES INTO A LOOP WHEREIN THE
270 / LINE-BY-LINE ASSEMBLY ROUTINE ASM02 IS
271 / CALLED ONCE FOR EACH ITEM ON THE PAGE TABLE.
272 / ASM02 IS A HUGE ROUTINE OCCUPYING ABOUT
273 / THREE FULL PAGES OF CODE, ASM02 FIRST
```



272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318

/EXAMINES THE CODED DATA PERTAINING TO THE GIVEN  
/ITEM ON THE PAGE TABLE TO DETERMINE WHAT  
/TYPE OF INSTRUCTION IS TO BE ASSEMBLED AND  
/WHAT THE CURRENT BANK AND SKIP SETTINGS  
/ARE. THEN DEPENDING ON THIS ANALYSIS THE ROUTINE  
/TRANSFERS TO THE PROPER SUBSECTION OF ITSELF  
/FOR HANDLING THIS TYPE OF INSTRUCTION. THERE  
/ARE A DOZEN OR MORE CASES WHICH MUST  
/BE DEALT WITH, THEN THE ROUTINE MOVES  
/TO ONE OF ITS VARIOUS EXIT STRINGS  
/TO COMPLETE THE ACTION AND SET THE BANK AND  
/SKIP CONDITIONS FOR THE NEXT LINE. AFTER ALL  
/ITEMS ON THE PAGE TABLE HAVE BEEN ASSEMBLED  
/IN THIS WAY ASMBL THEN CALLS THE ROUTINE  
/A2. A2 PRODUCES (IN PHASE 2) THE PAGE ESCAPE AND  
/THEN OUTPUTS THE ENTIRE LITERAL TABLE WITH  
/ALL THE OFF-PAGE POINTERS INTERMINGLED.  
/(10) SPECIAL CONSIDERATION SHOULD BE GIVEN TO OFF-  
/PAGE FORWARD REFERENCES SINCE THEY WILL REMAIN  
/UNRESOLVED WHEN THE CURRENT PAGE HAS BEEN  
/ASSEMBLED, DURING ASSEMBLY WHEN  
/A REFERENCE TO AN OFF-PAGE, OR AN AS YET  
/UNDEFINED SYMBOL IS ENCOUNTERED THE SYMBOL'S  
/IDENTIFIER IS STORED ON THE LITERAL/OFF-PAGE POINTER  
/TABLE. THEN WHEN A2 IS OUTPUTTING THE  
/LITERAL TABLE ANY STILL UNDEFINED SYMBOLS ARE  
/DEALT WITH AS FOLLOWS, THE SYMBOL'S IDENTIFIER  
/TOGETHER WITH THE LOCATION RESERVED IN THE CURRENT  
/PAGE FOR ITS VALUE ARE STORED ON THE OCCURRENCE  
/TABLE. THE LOCATION WHERE THE POINTER MUST BE  
/STORED IN THE CURRENT PAGE IS MERELY LEFT  
/BLANK AT THIS TIME, THEN LATER ON WHEN THIS SYMBOL IS  
/ENCOUNTERED AS A TAG THE ROUTINE LF5CK  
/WHICH PROCESSES TAGS DURING ASSEMBLY WILL  
/REMOVE THE ITEM FROM THE OCCURRENCE TABLE AND  
/OUTPUT IT PRECEDING THE POINTER BY AN ORIGIN  
/TO THE CORRECT LOCATION.  
/(11) DURING PASS2 (THE LISTING PASS) MOST OF  
/THE SAME CODE IS USED TO PRODUCE THE  
/ASSEMBLY LISTING, HOWEVER THE TIMING IS  
/DIFFERENT, NOW THE COLLECTION-ASSEMBLY  
/ALTERATION IS CARRIED OUT ON A LINE-BY-LINE  
/BASIS RATHER THAN ON A PAGE-BY-PAGE BASIS.  
/(HOWEVER ALL THE PAGE TABLES AND COUNTERS MUST  
/STILL BE MAINTAINED JUST AS IN PASS1.) THE  
/PASS2 OPERATION DIFFERS FROM PASS1 IN THE

319  
320 /FOLLOWING RESPECTS, EACH TIME A LINE HAS  
321 /BEEN COLLECTED AND ITS DATA ENTERED INTO THE  
322 /PAGE TABLE IN THE NORMAL FIRST PASS WAY,  
323 /A CALL IS ISSUED TO THE LINE-BY-LINE  
324 /ASSEMBLY ROUTINE ASM02, SINCE ALL SYMBOLS  
325 /ARE NOW DEFINED THERE WILL BE NO UNRESOLVED  
326 /FORWARD REFERENCES ON OR OFF-PAGE, ASM02 ACTS  
327 /DURING PASS2 EXACTLY AS IT DOES DURING PASS1  
328 /WITH ONE BIG EXCEPTION: THE BINARY OUTPUT ROUTINE  
329 /OUTBN IS SUPPRESSED AND IN ITS PLACE IS  
330 /SUBSTITUTED THE LISTING ROUTINE WRITE,  
331 /WHEN THE PAGE COUNTERS INDICATE THAT THE PAGE  
332 /IS FULL THE DRIVER ROUTINE WILL CALL L95 AS  
333 /USUAL. HOWEVER IN PASS2 THE TWO CALLS  
334 /TO ASMBL ARE BY-PASSED AND INSTEAD A SINGLE  
335 /CALL TO A2 IS ISSUED SO AS TO GET THE  
336 /LISTING TO THE PAGE ESCAPE, THE LITERALS AND  
337 /THE OFF-PAGE POINTERS.  
338 /((12) THE REASON FOR HAVING SEPARATE LITERAL TABLES FOR THE  
339 /COLLECTION AND THE ASSEMBLY PHASES OF  
340 /THE PROGRAM IS THAT DURING PASS2 BOTH  
341 /PHASES OF THE PROGRAM ARE OPERATING SIMULTANEOUSLY  
342 /AND BOTH ARE BUILDING LITERAL TABLES IN A  
343 /DIFFERENT WAY,  
344 /((13) THE PAGE ESCAPE TABLE, PERSE, IS NECESSARY  
345 /IS THAT DURING PASS2 LOCATIONS  
346 /CANNOT BE ASSIGNED FOR LITERALS AND OFF-PAGE  
347 /POINTERS UNLESS THE FINAL PAGE ESCAPE  
348 /FOR THE PAGE IS KNOWN, HENCE THESE NUMBERS  
349 /ARE SAVED DURING PASS1,  
350 /((14) THE PAGE OP TABLE IS ACTUALLY A PART OF  
351 /THE PAGE TABLE,  
352 /((15) EXTERNAL SYMBOLS ARISE IN TWO WAYS:  
353 /FROM ENTRY STATEMENTS AND FROM CALL STATEMENTS,  
354 /THEY ARE ENTERED IN THE E.S.T. IN ORDER OF  
355 /APPEARANCE IN THE PROGRAM AND NUMBERED  
356 /ACCORDINGLY, THESE ARE THE NUMBERS WHICH THE  
357 /LOADER REFERS TO AS "LOCAL EXTERNAL NUMBERS."  
358 /((16) EQUIVALENCING OF TAGS  
359 /IS TREATED AS A PSEUDO-OP AND IS  
360 /HANDLED BY THE ROUTINE PBSS2, (INCIDENTALLY  
361 /SOME OF THE ODD NAMES IN THE SOURCE WERE  
362 /PASSED ON TO ME FROM THE ORIGINAL ICS  
363 /PROGRAM, I DID NOT CHANGE THEM MERELY BECAUSE  
364 /THEY MADE NO SENSE,) THE OPERATION IS THIS:  
365 /ALL EXTRA TAGS TO BE DEFINED AT A GIVEN LOCATION ARE  
366 /ENTERED AS A GROUP IN THE EQUIVALENCE TABLE,  
367 /AND A CODE BIT IS SET ON THE PAGE TABLE TO  
368 /INDICATE THAT SUCH A GROUP IS TO BE DEFINED  
369 /WHEN THE LOCATION HAS BEEN DETERMINED DURING

```
370 /ASSEMBLY, ANUMCK DOES THE WORK OF DEFINITION./(*)
371 /V03 CHANGE NOTICE
372 / AS OF V03 THE SABR SYSTEM HAS
373 / BEEN CHANGED SUCH THAT COMMON
374 / WILL RESIDE IN FIELD 1 INSTEAD
375 / OF FIELD 0,
376 / THE ONLY CHANGES REQUIRED TO SABR
377 / ITSELF ARE AS FOLLOWS
378 / (1) HICOM=177 INSTEAD OF 777
379 / (2) PARG & ASMBL MUST NOW OUTPUT 6211'S
380 / INSTEAD OF 6201'S FOR CDF'S TO COMMON,
381 / NOTE
382 / THE COMMENTS HAVE NOT BEEN CHANGED TO
383 / REFLECT THIS CHANGE,
384 / ALSO, BANK = 0 IS STILL THE CONDITION
385 / FOR REFERENCES TO COMMON, (BANK=1
386 / STILL MEANS BANK KNOWN TO BE CURRENT
387 / AND BANK = =1 STILL MEANS BANK UNKNOWN.)
388
```

```
389
390
391      0000      /
392      /
393      /
394      /
395      /
396      0010      DISPL=10
397      0045      CDFSK=35*DISPL
398      0051      CDZSK=41*DISPL
399      0067      DUMS=57*DISPL
400      0033      LINK=23*DISPL
401      0055      OBIS=45*DISPL
402      0062      OPIS=52*DISPL
403      0040      RTN=30*DISPL
404
405
406      0001      *1
407      0001 0000      USE,      0
408      0002 0000      VAL,      0
409      0003 0000      SYMBOL, 0      /PTR TO CURRENT USE WORD IN MST
410      0004 7771      M7,      -7
411      0005 2150      AS0,      S0
412      0006 7577      OTP,      CORE4=1      /OCC. TAB. PTR (NEXT FREE WORD BELOW)
413      0007 2430      STT,      STTP      /PTR TO 1ST FREE WORD OF SYM, TAB.
414      / (KEEP STT AFTER OTP FOR INITA)
415
416      0010 0000      X0,      0      /LINE BUFFER INDEX
417      0011 0000      X1,      0      /TEMP AUTOS
418      0012 0000      X2,      0
419      0013 0000      X3,      0      /MSR BUFFER INDEX
420
421      0014 0002      K2,      2
422      0015 0004      K4,      4
423      0016 0003      K3,      3
424      0017 0130      K130, 130
425      0020 0030      K30,      30
426
```

427		/	INDIRECT REFERENCES	
428		/		
429	0021	5310	ICPLFS, CPLFS	/CHECK FOR AND PROCESS COLLECTION LFS
430	0022	2215	CPGESI, CPGES	
431	0023	1323	CTYPE, L61	/CHARACTER TYPEOUT ROUTINE
432	0024	4726	CRLF, L73	
433	0025	3726	DUMMY, DUM	/DUMMY ROUTINE
434	0026	4725	GETCHR, L65	/ROUTINE TO READ NEXT CHAR
435	0027	0600	GETSYM, GTSYM	/ROUTINE TO INPUT AND DECODE NEXT SYMBOL
436	0030	1000	INI, INILPT	
437	0031	3000	LFSCBK, LFSCK	/CHECK FOR A LFS
438	0032	4507	OBSYM, OBNSYM	/OBTAIN SYMBOL FROM MST
439	0033	2001	DCIL1, RDL1	
440	0034	4764	NULLP, NULL	
441	0035	3664	OTYPE, L62	/OCTAL TYPEOUT ROUTINE
442	0036	3400	OUTBIN, OUTBN	/ROUTINE TO OUTPUT COMP WORD AND REL BITS
443	0037	6261	OUTSKP, OUTSK	/ROUTINE TO OUTPUT A SKIP INSTRUCTION
444	0040	6443	POPEXP, POPEX	
445	0041	4000	PRSYMP, PRSYM	
446	0042	3164	PUNCH, L63	/BINARY PUNCH ROUTINE
447	0043	2000	RDIL, DCIL	/READ AND DECODE ONE INPUT LINE
448	0044	0443	RECTI, RECT	
449	0045	0400	L55I, L55	
450	0046	2200	SKIPL, L72	/SKIPS UNTIL A RETURN OR SEMICOLON
451	0047	2400	SLITAB, SLTAB	/SEARCH LITERAL TABLE
452	0050	2226	SPSTAB, SPSTB	/SEARCH PAGE SYMBOL TABLE
453	0051	3713	SREST, L66	/ROUTINE TO SEARCH EXTERNAL SYMBOL TABLE
454	0052	5000	STCE, SETCT	
455	0053	1347	TEST, TSCHR	/ROUTINE TO TEST CHARACTERS FOR EQUALITY
456	0054	4772	TYPE, L64	/TTY TYPE ROUTINE
457	0055	1060	WLNP, WLN	
458	0056	2532	WRITEP, WRITE	

```

459 /
460 /
461 /
462 0057 0000 ACTR, 0 /ASSEMBLY COUNTER
463 0060 0000 BSSSW, 0 /BSS 0 IN PROCESS SWITCH
464 0061 0000 CHR, 0 /LOC TO HOLD CURRENT CHARACTER
465 0062 0000 CSUM, 0 /BINARY CHECK SUM
466 0063 0000 EQVOPR, EQUTB /EQUIVALENCE TABLE OUTPUT POINTER
467 0064 0000 EQVIPR, EQUTB /EQ. TB, INPUT PTR.
468 0065 0000 ILC, 0 /CURRENT LOCATION
469 0066 0000 LFSPTR, 0 /POINTER TO LFS TABLE ENTRY
470 0067 0000 LINE, 0 /NO OF LINES SINCE LAST LFS
471 0070 0000 LITSZE, 0 /SIZE OF LIT TAB (ASM PHASE)
472 0071 0000 LTSZE, 0 /SAME FOR COLL. PHASE
473 0072 0000 LSTSKP, 0 /LAST INSTRUCTION SKIP INDICATOR
474 0073 0000 LSTBNK, 0 /LAST INSTRUCTION BANK INDICATOR
475 0074 0000 OBACTR, 0 /OFF BANK INSTRUCTION ADDITION COUNTER
476 0075 0000 OPSCTR, 0 /OFF PAGE SYMBOL COUNTER
477 /***** KEEP ITEMS SO INCLOSED IN THE GIVEN ORDER FOR INITA
478 0076 0177 HICOM, 0177
479 0077 0200 PAG, 0200 /CURRENT PAGE BITS
480 0100 0000 ESTSIZ, 0 /HOLDS SIZE OF EXTERNAL SYMBOL TABLE
481 0101 0000 EQVBIT, 0
482 0102 0000 APMSW, 0 /AUTOMATIC PAGING MODE SWITCH
483 0103 0001 TEM7, 1 /SPECIAL VARIABLE USED BY ASME5
484 0104 0001 CPSW, 1
485 0105 0000 DSW, 0
486 0106 0000 FORFLG, 0 /FORTR PSUEDO=OP FLAG
487 /POS NON=0 MEANS IGNORE DATA
488 0107 0000 SCOLON, 0
489 /*****
490 0110 0000 PASS, 0
491 0111 0000 PGEESC, 0 /HOLDS SIZE OF PAGE ESCAPE REQUIRED FOR CUR PAGE
492 0112 0000 PUPGE, 0
493 0113 0000 PHASE, 0 /PHASE SWITCH
494 0114 0000 PSTCPR, 0 /PAGE SYMBOL TABLE CODE POINTER
495 0115 0000 PSTSPR, 0 /PAGE SYMBOL TABLE SYMBOL POINTER
496 0116 0000 PSTSZE, 0 /SIZE OF PST
497 0117 0000 PTCPR, 0 /PAGE TABLE CODE POINTER
498 0120 0000 PTOPR, 0 /PAGE TABLE OP CODE POINTER
499 0121 0000 PTSPR, 0 /PAGE TABLE SYMBOL POINTER
500 0122 0000 PTSZE, 0 /SIZE OF PT
501 0123 0000 TEM1, 0
502 0124 0000 TEM2, 0
503 0125 0000 TEM3, 0
504 0126 0000 TEM4, 0
505 0127 0000 TEM5, 0
506 0122 0000 PTSIZ=PTSZE /KEYPUNCHING ERROR
507 0070 0000 LITSIZ=LITSZE /KEYPUNCHING ERROR
508 /
509 /LISTING VARIABLES
510 0130 0000 LFLG, 0 /0 IF NULL LINE
511 0131 0000 EFLG, 0 /ERROR FLAG, 6BIT CHAR. IN LEFT HALF
512 0132 0000 VFLG, 0 /0 IF NO VALUE TO OUTPUT
513 0133 0000 AFLG, 0 /DITTO FOR ADDRESS

```

/SABR.815	PAL10	V135	23=SEP=69	11128	PAGE 12-1
514	0134	0000	CODE, 0	/RELOCATION CODE	
515	0135	0000	ADDRESS, 0	/INSTR. ADDRESS	
516	0136	0000	VALUE, 0	/INSTR. VALUE	

517

518

519 0137 0000

520 0140 0000

521 0141 0000

522 0142 0000

523 0143 0000

524 0144 0000

525 0145 0000

526 0146 0000

527 0146 0000

528 0147 0000

529 0150 0000

530 0151 0000

531 0152 0000

532 0153 0000

533

534

535

536

537 0154 0005

538 0155 0007

539 0156 0010

540 0157 0020

541 0160 0040

542 0161 0077

543 0162 0100

544 0163 0177

545 0164 0200

546 0165 0240

547 0045

548 0027

549 0031

550 0030

551 0041

552 0166 7600

553 0166

554 0167 7524

555 0170 1645

556 0171 7776

557 0172 7775

558 0043

559 0052

560 0164

## /LINE INFO

LFS, 0

OP, 0

IB, 0

AFS, 0

UMIC, 0

NSGN, 0

EXP, 0

SK, 0

CURSKP=SK

BANK, 0

S0, 0

S1, 0

S2, 0

S3, 0

/KEEP THIS LIST ORDERED AS GIVEN  
/TO AGREE WITH TLFS LIST

## / FREQUENTLY USED CONSTANTS

/

K5, 5

K7, 0007

K10, 0010

K20, 0020

K40, 0040

K77, 0077

K100, 100

K177, 0177

K200, 0200

K240, 240

K400=L55I

K600=GETSYM

K3000=LFSCHK

K1000=INI

K4000=PRSYMP

K7600, 7600

M200=K7600

M254, =254

LINAX, LINBUF=1

M2, =2

M3, =3

K2000=RDIL

M3000=STCE

M7600=K200



```
561 /
562 / CORE LAYOUT POINTERS
563 /
564 0164 PTOPTR=K200 /PAGE OP CODE TABLE 1 IN BANK 1
565 0162 RSEEST=K100 /BASE OF EXTERNAL SYMBOL TABLE IN BANK 1
566 0043 MST=K2000 /BASE MAIN SYM, TAB IN BANK1
567 0027 LFSRSE=K600 /BASE OF LOCATION FIELD SYMBOL TABLE IN BANK 1
568 0045 LITBSE=L55I /BASE OF ASSEMBLY PHASE LITERAL TABLE IN BANK 1
569 0173 6776 PSTBSE, PSTB /BASE OF PAGE SYMBOL TABLE IN BANK 0
570 0174 7176 PTBSE, PTB /BASE OF PAGE TABLE IN BANK 0
571 0030 LTBSE=K1200 /BASE OF COLL. PHASE LIT. TABLE IN BANK 1
572 7176 PTB=7176
573 6776 PSTB=6776
574
575
576
577 5575 IERROR=JMP I , /ERROR MESSAGES
578 0175 2710 ERRI
579 5576 CERROR=JMP I ,
580 0176 2711 ERRC
581 5577 SERROR=JMP I ,
582 0177 2702 ERRS
583
```

```

584      0200      *0200
585      /
586      /
587      /
588      0200      7200      START,  CLA
589      0201      4777      JMS I   INITIO
590      0202      3110      DCA     PASS
591      0203      3726      DCA I   ICALSW
592      0204      4725      JMS I   INITAP
593      0205      4430      RSTRT,  JMS I   INI           /INITIALIZE PAGE TABLE POINTERS
594      0206      7410      SKP
595      0207      4727      RSTRT1, JMS I   INCPTI        /INCREMENT PAGE TABLE POINTERS
596      0210      4443      JMS I   RDIL          /INPUT AND DECODE ONE LINE
597      0211      4733      JMS I   CKCSWP        /CHK FOR MISSING ARG
598      0212      3060      DCA     BSSSW        /ALSO CLR BSS IS PROGRESS SW
599      0213      4452      JMS I   STCE          /SET COUNTERS FOR CURRENT LINE
600      0214      1140      TAD     OP           /OP CODE
601      0215      6211      CDF 10
602      0216      3520      DCA I   PTOPR        /TO PT OP CODE WORD
603      0217      6201      CDF 00
604      0220      1146      TAD     SK           /OR IN SKIP BIT
605      0221      7640      SZA  CLA
606      0222      1160      TAD     K40          /SKIP INST
607      0223      1517      TAD I   PTCPR        /IN CASE LFS BIT IN ALREADY
608      0224      3123      DCA     TEM1
609      0225      1141      TAD     IB           /OR IN INDIRECT BIT
610      0226      7640      SZA  CLA
611      0227      1045      TAD     K400         /YES
612      0230      1123      TAD     TEM1
613      0231      3123      DCA     TEM1          /FOR NEW PT CODE WORD
614      0232      1145      TAD     EXP          /DO WE HAVE A PAR?
615      0233      7640      SZA  CLA
616      0234      5242      JMP     RSTRT5        /YES
617      0235      1143      TAD     UMIC         /A MICRO INST?
618      0236      7650      SNA  CLA
619      0237      5245      JMP     RSTRT4        /NO AN MRI
620      0240      1015      TAD     K4           /OR IN OPERATE BIT
621      0241      5270      JMP     COMP          /EXIT TO COMPUTE PAGE SIZE
622      /
623      /
624      /
625      0242      1156      RSTRT5, TAD     K10          /PLACE PAR BIT ON PAGE TABLE
626      0243      1123      TAD     TEM1
627      0244      3123      DCA     TEM1
628      0245      1171      RSTRT4, TAD     M2
629      0246      1142      TAD     AFS          /IS AFS A CONSTANT
630      0247      7440      SZA
631      0250      5256      JMP     ,+6           /NO
632      0251      1157      TAD     K20          /YES ... CONSTANT BIT
633      0252      1123      RSTRT2, TAD  TEM1        /+PT CODE WORD
634      0253      3123      DCA     TEM1          /FOR NEW PT CODE WORD
635      0254      1150      TAD     S0           /ACTUAL BINARY CONSTANT
636      0255      5264      JMP     COMPGO        /EXIT TO COMPUTE PAGE SIZE
637      0256      7001      IAC
638      0257      7640      SZA  CLA           /IS AFS A LITERAL

```

```

639 0260 5263      JMP      ,+3          /NO
640 0261 1014      TAD      K2           /YES ... LITERAL BIT
641 0262 5252      JMP      RSTRT2      /SAVE AS CONSTANT FROM THIS POINT
642 0263 1142      TAD      AFS         /PLACE AFS ON PST
643 0264 3521      COMP60, DCA I      PTSPR
644 0265 1144      TAD      NSGN       /CK FOR # REF
645 0266 7640      SZA     CLA
646 0267 1043      TAD      K2000      /YES
647 0270 1123      COMP,  TAD      TEM1 /GET ALL THE BITS
648 0271 3517      DCA I      PTCPR    /TO THE CODE WORD
649
650 /
651 /
652 0272 1137      TAD      LFS         /IS THERE AN LFS
653 0273 7640      SZA     CLA
654 0274 4444      JMS I    RECTI      /YES ... EXIT TO RECOUNT PAGE
655 0275 4422      JMS I    CPGESI     /COMPUTE ACTUAL PAGE SIZE
656 0276 1166      TAD      M200       /SUBTRACT PHYSICAL PAGE SIZE
657 0277 7750      SPA     SNA CLA    /IS SIZE .GT. PHYSICAL SIZE
658 0300 5316      JMP      RSTRT6     /NO ... GET NEXT
659 0301 4731      JMS I    PSHINI     /YES ... PUSH CURRENT INPUT LINE
660 0302 1112      TAD      PUPGE      /RESTORE LAST PAGE ESCAPE
661 0303 3111      DCA     PGEESC
662 0304 7240      CLA     CMA        /DECREMENT PAGE TABLE SIZE
663 0305 1122      TAD      PTSZE
664 0306 3122      DCA     PTSZE
665 /
666 /
667 /
668 0307 4445      JMS I    L551       /ASSEMBLE CURRENT PAGE
669 0310 4776      JMS I    UDPG
670 0311 4732      JMS I    FIXI       /FIX ILC IF PASS 2
671 0312 4730      JMS I    POPINI     /POP LAST INPUT LINE
672 0313 4430      JMS I    INI        /INITIALIZE PT POINTERS
673 0314 3444      DCA I    RECTI      /CLR RECOUNT FLAG FOR CPLFS
674 0315 5211      JMP      RSTRT1+2   /EXIT TO PROCESS POPPED LINE
675 /
676 0316 1111      RSTRT6, TAD      PGEESC /SAVE CURRENT PAGE ESCAPE
677 0317 3112      DCA     PUPGE      /IN CASE NEXT LINE OVERFLOWS PAGE
678 0320 1110      TAD      PASS
679 0321 7640      SZA     CLA
680 0322 4724      JMS I    LASMP
681 0323 5207      JMP      RSTRT1
682 0324 5651      LASMP, ASM02
683 0325 4327      INITAP, INITA
684 0326 1963      ICALSW, CALLSW
685 0327 3551      INCPTI, INCPT
686 0330 4217      POPINI, POPIN
687 0331 4200      PSHINI, PUSHIN
688 0332 5561      FIXI,  FIXILC
689 0333 2762      CKCSWP, CKCSW

```

```

690 / PAGE PSEUDO OPERATION
691 /
692 0334 4446 PPAGE, JMS I SKIPL
693 0335 7240 CLA CMA /DECREMENT PAGE TABLE SIZE
694 0336 1122 TAD PTSZE
695 0337 7450 SNA /WATCH FOR ZERO
696 0340 5343 JMP ,+3
697 0341 3122 DCA PTSZE /FOR NEW PAGE TABLE SIZE
698 0342 4445 JMS I L551 /ASSEMBLE CURRENT PAGE
699 0343 4776 JMS I UDPG
700 0344 5370 JMP RORGX /INITIALIZE AND INPUT ANOTHER LINE
701 /
702 / REORG PSEUDO OPERATOIN
703 /
704 0345 4427 PRORG, JMS I GETSYM /GET NEXT INPUT ITEM
705 0346 7000 NOP /NOTHING THERE
706 0347 7410 SKP /SYMBOL
707 0350 7610 SKP CLA /CONSTANT
708 0351 5575 IERROR /LITERAL
709 0352 4446 JMS I SKIPL
710 0353 1150 TAD S0 /NEW RELOCATABLE ORIGIN
711 0354 0166 AND K7600 /MASK OFF PAGE DISPLACEMENT BITS
712 0355 7450 SNA /ARE WE TRYING TO REORIGIN BELOW 200
713 0356 5575 IERROR /YES ... NOT ALLOWED
714 0357 3374 DCA RORG1 /SAVE NEW ORIGIN
715 0360 7240 CLA CMA /DECREMENT PAGE TABLE SIZE
716 0361 1122 TAD PTSZE
717 0362 7450 SNA /IS THIS THE BEGINNING OF A PAGE
718 0363 5366 JMP ,+3 /YES
719 0364 3122 DCA PTSZE
720 0365 4445 JMS I L551
721 0366 1374 TAD RORG1 /NEW ORIGIN
722 0367 3077 DCA PAG /TO PROPER LOCATION
723 0370 4732 RORGX, JMS I FIXI
724 0371 1375 TAD RSTRTX /RETURN AT RSTRT INSTEAD OF RDL1
725 0372 3033 DCA DCIL1
726 0373 5434 JMP I NULLP /RE-INITIALIZE AND GO
727 0374 0000 RORG1, 0
728 0375 0205 RSTRTX, RSTRT
729 0376 3172 UDPG, UDPAGE
730 0377 6456 INITIO, IOINIT

```



/SABR,815

PAL14

V135

23-SEP-69

11128 PAGE 17-1

786	0421	1231		TAD	L56+1	/RESTORE DUMMY ROUTINE
787	0422	3025		BCA	DUMMY	
788	0423	2057		ISZ	ACTR	/SET ASSEMBLY COUNTER
789	0424	1235		TAD	TEM55	/RESTORE AS BEFORE 1ST ASSEMBL
790	0425	3063		BCA	EQVOPR	
791	0426	4632		JMS I	ASSEMBL	/ASSEMBLE AND OUTPUT THIS TIME
792	0427	5600		JMP I	L55	/RETURN
793	0430	3400	L56,	OUTRN		
794	0431	3706		QUM		
795	0432	5400	ASSEMBL,	ASMBL		
796	0433	3537	L55A,	HCBPS		
797	0434	6725	L55B,	EQVFIX		
798	0435	0000	TEM55,	0		
799						
800						
801						
802	0436	4641	L55L,	JMS I	A2P	
803	0437	4642		JMS I	A1P	/INITIALIZE NEXT PAGE
804	0440	5600		JMP I	L55	
805	0441	5424	A2P,	A2		
806	0442	5526	A1P,	A1		
807						
808						

```

829
810
811 /
812 /COLLECTION PHASE ROUTINE
813 /RECOUNT THE CURRENT PAGE BECAUSE OF AN LFS
814 /CALL WITH AC=0, LEAVES AC=0
815 /FUNCTION:WHEN A NEW TAG IS DEFINED ON PAGE
816 / OPSCTR & OBACTR MAY NEED TO BE
817 / REDUCED, CPLFS TAKES CARE OF OPSCTR
818 / BUT OBACTR REQUIRES REVIEWING THE
819 / ENTIRE PAGE.
820 /OPERATION: (1) CALL CLNPST TO CLEAR BITS 1-9
821 / OF ALL PST CODE WORDS=WIPES OUT
822 / SHARE OF OBACTR DUE TO EACH SYM.
823 / (2) RE-INIT PAGE & CLR OBACTR
824 / (3) FETCH ITEM FROM PAGE TABLE
825 / (4) SET ALL INSTR. TYPE FLAGS ACCORDINGLY
826 / (5) CALL SETC
827 / (6) INC PAGE TABLE PTRS TO NEXT ITEM
828 / & LOOP BACK TO (3)
829 / CONTINUE THRU ENTIRE TABLE.
830 0443 7000 RECT, 0
831 0444 1116 TAD PSTSIZE /ANYTHING ON PST?
832 0445 7440 SZA
833 0446 5731 JMP I CLENUP /YES, CLEAN PST CODES
834
835 0447 4730 RECRET, JMS I INISS /DO INITS.
836 0450 3074 DCA OBACTR /ZERO OFF BANK ADDITION COUNTER
837 0451 1122 TAD PTSZE /SIZE OF PT
838 0452 7041 CIA
839 0453 3200 DCA RECT1 /TO INDEX LOCATION
840
841 /
842 / THIS IS THE RECOUNT LOOP
843 /
844 0454 6211 RECT2, CDF 10
845 0455 1520 TAD I PTOPR /OP CODE FROM PT
846 0456 6201 CDF 00
847 0457 3140 DCA OP
848 0460 1517 TAD I PTCPR /CK FOR SKIP INST
849 0461 2160 AND K40
850 0462 3146 DCA SK
851 0463 1517 TAD I PTCPR /CK FOR # REF
852 0464 0043 AND K200
853 0465 3144 DCA NSGN
854 0466 1517 TAD I PTCPR /PT CODE WORD
855 0467 0015 AND K4 /IS IT AN OPERATE INSTRUCTION
856 0470 3143 DCA UMIC
857 0471 1517 TAD I PTCPR /CK FOR PAR EXP
858 0472 2156 AND K10
859 0473 3145 DCA EXP
860 0474 1517 TAD I PTCPR /PAGE TABLE CODE WORD
861 0475 0045 AND K400 /MASK OFF INDIRECT BIT
862 0476 3141 DCA IB /PLACE IT IN PROPER LOCATION
863 0477 1517 TAD I PTCPR /PT CODE WORD
864 0520 0157 AND K20 /IS AFS A CONSTANT

```

864 0521 7112  
 865 0522 7440  
 866 0523 5326  
 867 0524 1517  
 868 0525 2014  
 869 0526 7110  
 870 0527 7450  
 871 0510 1521  
 872 0511 3142  
 873 0512 1521  
 874 0513 3150  
 875  
 876  
 877  
 878 0514 1517  
 879 0515 0332  
 880 0516 7650  
 881 0517 5322  
 882 0520 7240  
 883 0521 3147  
 884 0522 4452  
 885 0523 2200  
 886 0524 7410  
 887 0525 5643  
 888 0526 4733  
 889 0527 5254  
 890 0528 2400  
 891 0530 1041  
 892 0531 6424  
 893 0532 0201  
 894 0533 6701

CLL RTR  
 SZA  
 JMP ,+3 /YES  
 YAD I PTCPR /PT CODE WORD  
 AND K2 /IS AFS A LITERAL  
 CLL RAR  
 SNA  
 YAD I PTSPR /ADDRESS FIELD SYMBOL  
 DCA AFS  
 YAD I PTSPR /ACTUAL LITERAL  
 DCA S0 /TO LITERAL LOCATION  
 /  
 / AREA WHICH CALLS COUNT ROUTINE  
 /  
 RECT3, YAD I PTCPR /PT CODE WORD  
 AND K201 /IS THERE A TAG OR AN EQUIVALENCED TAG?  
 SNA CLA  
 JMP ,+3 /NO  
 CLA CMA /YES ... SET BANK UNKNOWN  
 DCA BANK  
 JMS I STCE /CALL COUNT ROUTINE  
 ISZ RECT1 /OVER YET  
 SKP /NO  
 JMP I RECT /EXIT  
 JMS I ISZPT1  
 JMP RECT2 /GO GET NEXT LINE  
 RECT1=L55  
 INISS, INISUB  
 CLENUM, CLNPST  
 K201, 201  
 ISZPT1, ISZPT



```

895 /
896 /
897 /
898 2534 1126 PENO, TAD FORFLG /IF FLAG ON, TURN OFF &
899 2535 7750 SPA SNA CLA /GO TO ROL1
900 2536 5342 JMP ,+4
901 2537 7040 CMA /TURN OFF FORTR P=OP
902 2540 3126 DCA FORFLG
903 2541 5433 JMP I RCIL1
904 2542 7240 CLA CMA /DECREMENT PT SIZE
905 2543 1122 TAD PTSZE
906 2544 7450 SNA /ARE WE AT THE BEGINNING OF A PAGE
907 2545 5351 JMP ,+4 /YES
908 2546 3122 DCA PTSZE /NO ... NEW PAGE TABLE SIZE
909 2547 2102 ISZ APMSW /LEAVE AUTO PAGING MODE FOR LAST PAGE
910 2550 4200 JMS L55 /ASSEMBLE CURRENT PAGE
911 2551 1062 TAD CSUM
912 2552 3123 DCA TEM1
913 2553 4436 JMS I OUTBIN /OUTPUT CKSUM
914 2554 2123 TEM1
915 2555 2010 10
916 2556 1110 TAD PASS
917 2557 7640 SZA CLA
918 2560 5370 JMP ENDEND
919 2561 4775 JMS I LEAD /OUTPUT TRAILER CODE
920 2562 4441 JMS I PRSYMP /TYPE OUT SYMBOL TABLE
921 2563 2110 ISZ PASS
922 2564 4774 JMS I INITAI
923 2565 4642 JMS I A1P
924 2566 7402 HLT
925 2567 5773 JMP I REE
926 2570 4455 ENDEND, JMS I WLNK /LIST THE "END" STATMT
927 2571 7602 HLT CLA
928 2572 5564 JMP I K200 /RESTART AT 200
929 2573 2205 REE, RSTRT
930 2574 4327 INITAI, INITA
931 2575 4714 LEAD, LEADER
932
933

```

```

934          *620
935          /READ INPUT ITEM
936          /      IGNORES SPACES & TABS TO 1ST CHAR OF ITEM
937          /ASSUMES AC=0
938          /CALLING SEQ: JMS I GETSYM
939          /      NULL RETURN (IF NO ITEM FOUND BEFORE CR / *
940          /      SYMBOL RET, (WITH SYM PACKED IN S1=S3
941          /      AND S0=SYMBOL LENGTH)
942          /      CONST, RET, (WITH VALUE IN S0)
943          /      LITERAL RET, (WITH VALUE IN S0)
944          /SYNTAX: LITERALS: (000      NUMERIC LIT.
945          /      (=000      NEG.
946          /      (K000      OCTAL
947          /      (D000      DECIMAL
948          /      ("A      ASCII LIT.
949          /      (= "A      NEGATIVE ASCII
950          /      CONSTANTS: 000,-000,"A,OR ="A
951          /      NOTE: AFTER A VALID QUOTE ANY ASCII CHAR MAY APPEAR
952          /      AND WILL BE STORED AS THE CONST OR LIT VALUE.
953          /      THIS INCLUDES CR / * SO THESE DO NOT
954          /      TERMINATE A LINE AFTER A QUOTE.
955          /ALL EXITS LEAVE AC=0
956          /NOTE: TO PROVIDE A CHECK OF THE PUNCTUATION
957          /CHAR, FOLLOWING PREVIOUSLY READ SYMBOL, GTSYM
958          /DECREMENTS THE LINE PTR BEFORE STARTING
959          /THE READ, IF THIS IS NOT WANTED
960          /CALL TO GTSYM MUST BE PRECEDED
961          /BY "ISZ X0"
962
963          0600 0000 GTSYM, 0
964          0621 7040          CMA          /DECREMENT CHARACTER PTR
965          0602 1010          TAD X0
966          0623 3010          DCA X0
967          0604 1105          TAD DSW          /SAVE NUMERIC MODE
968          0605 3126          DCA TEM4
969          0606 7040          CMA          /SIGN=1 FOR POSITIVE
970          0627 3355          ITM4, DCA SIGN          /SIGN=0 TO FORCE NEGATION
971          0610 4754          ITM2, JMS I, RC          /READ 1ST CHAR
972          0611 5264          JMP ITM5          /DIGIT: GET NUMERIC CONST
973          0612 5217          JMP ITM3          /ALPHA: GET SYMBOL
974          0613 4453          JMS I TEST          /SORT LEADING PUNCT.
975          0614 0751          SL2=1
976          0615 0011          BL2=SL2
977          0616 5576          CERROR          /ILLEGAL CHAR
978
979          /READ IN A SYMBOL
980          /ASSUMES 1ST CHAR ALREADY READ IN & SAVED IN CHR
981          /LEAVES SYMBOL PACKED IN 68BIT CHAR PAIRS IN S1=S3
982          /      S0=NUMBER OF CHAR PAIRS ACTUALLY USED
983
984          0617 2355          ITM3, ISZ SIGN          /CK FOR -SYMBOL
985          0620 5576          CERROR          /YES
986          0621 3150          DCA S0          /CLR FOR SYM LENGTH COUNT
987          0622 3123          DCA TEM1          /CLR FOR CHAR COUNT
988          0623 3125          DCA TEM3          /SET PTR FOR LEFT BYTE

```

```

/SABR.815      PAL10  V135  23-SEP-69  11128  PAGE 20-1
 989  0624  1005      TAD AS2      /AUTO-INDEX STORAGE IN S1-S3
 990  0625  3012      DCA X2
 991  0626  2123      RSM2,  ISZ TEM1      /COUNT CHAR
 992  0627  1123      TAD TEM1      /ARE MORE THAN 6 CHARS IN?
 993  0630  1004      TAD M7
 994  0631  7700      SMA CLA
 995  0632  5251      JMP RSM1      /YES, IGNORE
 996  0633  1061      TAD CHR      /NO, GET ASCII
 997  0634  0161      AND K77      /MASK TO 6BIT
 998  0635  2125      ISZ TEM3      /WHICH BYTE?
 999  0636  5242      JMP RSM3      /LEFT
1000  0637  1124      TAD TEM2      /ADD ON LEFT HALF
1001  0640  3412      DCA I X2      /STORE CHAR PAIR IN S1-S3
1002  0641  5251      JMP RSM1
1003  0642  7106      RSM3,  RTL CLL      /MOVE 6BIT TO LEFT BYTE
1004  0643  7006      RTL
1005  0644  7006      RTL
1006  0645  3124      DCA TEM2      /SAVE WHILE WAITING ON RT BYTE
1007  0646  7040      CMA          /SET PTR FOR RT BYTE
1008  0647  3125      DCA TEM3
1009  0650  2150      ISZ S0       /COUNT 1 SYMBOL WORD
1010  0651  4754      RSM1,  JMS I RC      /READ NEXT CHAR
1011  0652  5226      JMP RSM2      /DIGIT
1012  0653  5226      JMP RSM2      /ALPHA
1013  0654  2125      ISZ TEM3      /PUNCT=END OF SYM1 CHECK BYTE PTR
1014  0655  5260      JMP ,+3      /NOTHING IN TEM2
1015  0656  1124      TAD TEM2      /SAVE THE ODD CHAR
1016  0657  3412      DCA I X2
1017  0660  1756      TAD I IFCTP   /SKIP SYM TAB IF IF=COUNT NOT UP
1018  0661  7700      SMA CLA
1019  0662  4753      JMS I SRS     /LOOK IT UP IN SYM TAB, & ENTER IF NEC.
1020  0663  5350      JMP ITM14     /EXIT
1021
1022      /READ DIGIT STRING
1023      /ASSUMES 1ST DIGIT ALREADY READ AND ASCII SAVED IN CHR
1024      /
1025      /   SGN=#1 IF NUM, IS TO BE NEGATED
1026      /   DSW=#0 FOR OCTAL CONVERSION, 1 FOR DECIMAL
1027      /LEAVES AC=OCTAL VALUE OF DIGIT STRING (NEG IF SGN=#1)
1028      /   CHR=ASCII FOR TERMINAL PUNCTUATION
1029  0664  3123      ITM5,  DCA TEM1      /CLEAR FOR ACCUMULATION
1030  0665  1061      RDS1,  TAD CHR      /REDUCE CHR TO OCTAL VALUE
1031  0666  1352      TAD M200A
1032  0667  3124      DCA TEM2
1033  0670  1105      TAD DSW      /OCTAL OR DECIMAL CONVERSION?
1034  0671  7640      SZA CLA      /OCTAL, CK FOR 8 OR 9
1035  0672  5302      JMP MUL1     /DECIMAL, 8 OR 9 IS OK
1036  0673  1124      TAD TEM2      /VALUE = 8 OR 9?
1037  0674  1004      TAD M7
1038  0675  7740      SMA SZA CLA  /NO, GO ON
1039  0676  5976      CERROR      /YES
1040      /MULT, PREV. VAL. BY CONV, FACTOR
1041  0677  1123      TAD TEM1
1042  0700  7106      CLL RTL      /ARG *4
1043  0701  5305      JMP MUL1+3

```

/SABR, 815

PAL10

V135

23=SEP=69

11128 PAGE 24=2

1044	0702	1123	MUL1,	TAD TEM1	
1045	0703	7126		CLL RTL	/ARG * 4
1046	0704	1123		TAD TEM1	/PLUS ARG=ARG*5
1047	0705	7024		RAL	/*2
1048	0706	1124		TAD TEM2	/ADD NEW DIGIT
1049	0707	3123		DCA TEM1	/SAVE ACCUMULATED VALUE
1050	0710	4754		JMS I RC	/READ NEXT CHAR,
1051	0711	5265		JMP RDS1	/DIGIT
1052	0712	5576		CERROR	/ALPHA
1053	0713	1123		TAD TEM1	/PUNCT, GET TOTAL
1054	0714	2355	ITM6,	ISZ SIGN	/IS NEGATE SW, SET?
1055	0715	7041		CIA	/YES
1056	0716	3150		DCA S0	/STORE CONST VALUE
1057	0717	1126		TAD TEM4	/RESTORE NUMERIC MODE
1058	0720	3105		DCA DSW	
1059	0721	5347		JMP ITM13	/EXIT
1060	0722	4426	ITM7,	JMS I GETCHR	/READ ALPHA CONST,
1061	0723	7450		SNA	
1062	0724	5575		IERROR	/NOTHING THERE
1063	0725	3123		DCA TEM1	
1064	0726	4426		JMS I GETCHR	/READ NEXT CHAR FOR BENEFIT OF SKIPL
1065	0727	7200		CLA	
1066	0730	1123		TAD TEM1	
1067	0731	5314		JMP ITM6	
1068	0732	4757	ITM8,	JMS I CKIFP	/MOVE PTR TO LITERAL EXIT
1069	0733	4754	ITM9,	JMS I RC	/READ 1ST CHAR OF LIT,
1070	0734	5264		JMP ITM5	/DIGIT NUMERIC LIT,
1071	0735	7000		NOP	/ALPHA MUST BE K OR D
1072	0736	4453		JMS I TEST	/LOOK FOR K, D, ", =
1073	0737	0737		SL3=1	
1074	0740	0033		BL3=SL3	
1075	0741	5576		CERROR	/ILLEGAL CHAR
1076	0742	3355	ITM10,	DCA SIGN	/SET FLAG FOR NEG. LIT,
1077	0743	5333		JMP ITM9	
1078	0744	7001	ITM11,	IAC	/FORCE DECIMAL LIT,
1079	0745	3105	ITM12,	DCA DSW	/FORCE OCTAL LIT,
1080	0746	5333		JMP ITM9	
1081	0747	4757	ITM13,	JMS I CKIFP	/CONST, EXIT
1082	0750	4757	ITM14,	JMS I CKIFP	/SYMBOL EXIT
1083	0751	5600	ITM15,	JMP I GTSYM	/NULL EXIT
1084	0752	7520	M260A,	=260	
1085	0753	4400	SRS,	SRSYM	
1086	0754	4546	RC,	RCH	
1087	0755	0000	SIGN,	0	
1088	0756	6371	IFCTP,	IFCTR	
1089	0757	6746	CKIFP,	CKIF	

1090		5760	AERROR=JMP I ,	
1091	0760	2712	ERRA	
1092	0761	1563	CALLSP; CALLSW	
1093			/	
1094			/CHECK FOR TOO FEW ARGS	
1095			/AERROR IF CALLSW MINUS	
1096			/	
1097	0762	0000	CKCSW, 0	
1098	0763	1761	TAD I CALLSP	/CK
1099	0764	7700	SMA CLA	
1100	0765	5762	JMP I CKCSW	/OK
1101	0766	2761	ISZ I CALLSP	/COUNT MISSING ARG
1102	0767	7000	NOP	
1103	0770	5760	AERROR	/FLAG
1104			/	
1105			/CHECK FOR TOO MANY ARGS	
1106			/AERROR IF CALLSW POSITIVE	
1107			/	
1108	0771	0000	CKCLS, 0	
1109	0772	1761	TAD I CALLSP	/DO WE WANT THIS ARG?
1110	0773	7700	SMA CLA	
1111	0774	5760	AERROR	/NO, ARG COUNT OVERFLOW
1112	0775	2761	ISZ I CALLSP	/YES, COUNT THIS ARG
1113	0776	7000	NOP	
1114	0777	5771	JMP I CKCLS	

1115 1000 \*1000  
 1116 /  
 1117 /  
 1118 /  
 1119 1000 0000  
 1120 1001 7201  
 1121 1002 3122  
 1122 1003 3116  
 1123 1004 3071  
 1124 1005 3070  
 1125 1006 3075  
 1126 1007 3113  
 1127 1010 4241  
 1128 1011 7240  
 1129 1012 3636  
 1130 1013 7201  
 1131 1014 3637  
 1132 1015 1101  
 1133 1016 7440  
 1134 1017 5222  
 1135 1020 3064  
 1136 1021 3063  
 1137 1022 3517  
 1138 1023 3521  
 1139 1024 6211  
 1140 1025 3520  
 1141 1026 6201  
 1142 1027 1027  
 1143 1030 3066  
 1144 1031 3074  
 1145 1032 1235  
 1146 1033 3033  
 1147 1034 5600  
 1148  
 1149 1035 2001  
 1150 1036 5774  
 1151 1037 5775  
 1152 1040 7567  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160 1041 0000  
 1161 1042 1174  
 1162 1043 3117  
 1163 1044 1174  
 1164 1045 7001  
 1165 1046 3121  
 1166 1047 1164  
 1167 1050 3120  
 1168 1051 7240  
 1169 1052 3073

ROUTINE TO INITIALIZE POINTERS FOR THE COLLECTION OF A PAGE

INILPT, 0

CLA IAC  
 DCA PTSZE /SET PAGE TABLE SIZE  
 DCA PSTSZE /ZERO PAGE SYMBOL TABLE SIZE  
 DCA LTSZE /ZERO LITERAL TABLE SIZE (COLL. PHASE)  
 DCA LITSZE /8 ASMBLY PHASE LIT TABL  
 DCA OPSCTR /ZERO OFF PAGE SYMBOL COUNTER  
 DCA PHASE /SET PHASE SWITCH TO COLLECTION  
 JMS INISUB  
 CLA CMA  
 DCA I BNKSV  
 CLA IAC  
 DCA I LSTSKK  
 TAD EQVBIT /ANY EQUIV. LEFT FROM LAST PAGE?  
 SZA  
 JMP EQSAV /YES, SKIP TABLE REINIT & SAVE BIT  
 DCA EQVIPR /NO. RE-INIT EQ. TAB. PTRS  
 DCA EQVOPR  
 EQSAV, DCA I PTCPR /INITIALIZE PAGE TABLE CODE WORD  
 DCA I PTSPR /INITIALIZE PAGE TABLE SYMBOL WORD  
 CDF 10  
 DCA I PTOPR /INITIALIZE PT OP CODE WORD  
 CDF 00  
 TAD LFSBSE /INITIALIZE LFS TABLE POINTER  
 DCA LFSPTR  
 DCA OBACTR /ZERO OFF BANK ADDITION COUNTER  
 TAD RDL1X /RESTORE IN CASE OF REORG OR PAGE PSUEDO  
 DCA DCIL1  
 JMP I INILPT /RETURN

RDL1X, RDL1  
 BNKSV, BNKSAV  
 LSTSKK, SKPSAV  
 M211, =211

/GENERAL PAGE TABLE INITIALIZATION  
 /DOES PARTS OF INITIALIZ. COMMON TO SEVERAL  
 /ROUTINES  
 /

INISUB, 0

TAD PTBSE /INITIALIZE PAGE TABLE CODE POINTER  
 DCA PTCPR  
 TAD PTBSE /INITIALIZE PAGE TABLE SYMBOL POINTER  
 IAC  
 DCA PTSPR  
 TAD PTOPTB /INITIALIZE PT OP CODE POINTER  
 DCA PTOPR  
 CLA CMA /SET LAST BANK UNKNOWN  
 DCA LSTBNK

/SABR.015

PALIV

V135

23-SEP-69

11128 PAGE 22-1

1170 1053 7241  
1171 1054 3072  
1172 1055 7240  
1173 1056 3147  
1174 1057 5641

CLA IAC  
DCA LSTSKP  
CLA CMA  
DCA BANK  
JMP I INISUB

/SET LAST INSTRUCTION SKIP INDICATOR ON

/SET CURRENT BANK UNKNOWN

1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187  
 1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229

1060 2000  
 1061 1130  
 1062 7650  
 1063 5353  
 1064 1133  
 1065 7640  
 1066 5272  
 1067 4423  
 1070 4423  
 1071 5274  
 1072 1135  
 1073 4435  
 1074 1131  
 1075 4423  
 1076 1132  
 1077 7640  
 1100 5304  
 1101 4423  
 1102 4423  
 1103 5306  
 1104 1136  
 1105 4435  
 1106 1165  
 1107 4454  
 1110 1134  
 1111 4423  
 1112 6211  
 1113 1764  
 1114 6201  
 1115 7650  
 1116 5353  
 1117 4423  
 1120 1165  
 1121 4454  
 1122 1170  
 1123 3011  
 1124 3127  
 1125 6211  
 1126 1411  
 1127 6201

```

/
/SUBR, TO WRITE A LINE
/MAY BE USED ONLY DURING PASS 2 (LISTING)
/FUNCTION TYPES (OR PUNCHES) EACH LINE OF SOURCE
/ WITH PROPER ASSEMBLY ADDR, & CODES
/ AT BEGINNING OF LINE (OR SPACES IF
/ THESE ARE OMITTED).
/LINE FORMAT:
/ADDR VALU RC CONTENTS OF LINE BUFFER
/ERROR FLAGS TYPED BETWEEN ADDR & VALU
/COLUMNS, RC=RELOCATION CODE, THE LINE
/BUFFER IS IN FIELD 1 AT "LINBUF,"
/
WLN, 0
TAD LFLG /NULL LINE?
SNA CLA
JMP WLN3 /YES
TAD AFLG
SZA CLA
JMP ,+4
JMS I CTYPE /IF AFLG=0 TYPE 4 SPACES
JMS I CTYPE
JMP ,+3
TAD ADDRESS /OTHERWISE TYPE 4 DIGITS
JMS I OTYPE
TAD EFLG /TYPE ERR, FLAG & SPACE
JMS I CTYPE
TAD VFLG /SAME TREATMENT FOR VALUE
SZA CLA /AS FOR ADDRESS
JMP ,+4
JMS I CTYPE
JMS I CTYPE
JMP ,+3
TAD VALUE
JMS I OTYPE
TAD K240 /SPACE
JMS I TYPE
TAD CODE /2 DIGITS OR 2 SPACES
JMS I CTYPE
CDF 10
TAD I LINEB /IS THERE ANY LINE TO TYPE?
SNA CLA
JMP WLN3 /NO, EXIT
JMS I CTYPE /2 SPACES
TAD K240 /3RD SPACE
JMS I TYPE
TAD LINAX /INDEX LINE BUFFER
DCA X1
DCA CHARCT /CLR COUNTER
WLN1, CDF 10
TAD I X1 /GET CHAR
CDF 00

```



```

1232 1130 7450 SNA
1231 1131 5353 JMP WLN3 /END OF LINE
1232 1132 3061 DCA CHR
1233 1133 1061 TAD CHR /CK FOR TAB
1234 1134 1240 TAD M211
1235 1135 7650 SNA CLA
1236 1136 5343 JMP WLN2 /YES
1237 1137 2127 ISZ CHARCT /COUNT 1 CHAR
1238 1140 1061 TAD CHR /OUTPUT IT
1239 1141 4454 JMS I TYPE
1240 1142 5325 JMP WLN1
1241 1143 1165 WLN2, TAD K240 /SIMULATE TAB
1242 1144 2127 ISZ CHARCT
1243 1145 4454 JMS I TYPE
1244 1146 1127 TAD CHARCT
1245 1147 0155 AND K7
1246 1150 7640 SZA CLA
1247 1151 5343 JMP WLN2 /CONTINUE TAB
1248 1152 5325 JMP WLN1
1249 1153 4424 WLN3, JMS I CRLF
1250 1154 3132 DCA VFLG
1251 1155 3131 DCA EFLG
1252 1156 3133 DCA AFLG
1253 1157 3134 DCA CODE
1254 1160 6211 CDF 10
1255 1161 3764 DCA I LINER
1256 1162 6201 CDF 00
1257 1163 5660 JMP I WLN
1258
1259 1164 1646 LINEB, LINBUF
1260 0127 CHARCT=TEM5
1261 /
1262 /PATCH FOR SETCT
1263 /NOT USED BY ANY OTHER PART OF PROGRAM
1264 /(ADDED AT V15)
1265 /
1266 /FUNCTION: SET BANK=1 AFTER A "CALL"
1267 /(MUST BE DONE FOR BENEFIT OF RECT ROUTINE)
1268 /
1269 1165 0000 SETCAL, 0
1270 1166 1517 TAD I PTCPR /CK FOR CALL CONST.
1271 1167 0162 AND K100
1272 1170 7650 SNA CLA
1273 1171 5374 JMP ,+3 /NO
1274 1172 7001 IAC /YES, BANK TO CURRENT
1275 1173 3147 DCA BANK
1276 1174 1145 TAD EXP /DO 2 INSTRUCTIONS THAT
1277 1175 1143 TAD UMIC /WERE KNOCKED OUT OF SETCT
1278 1176 5765 JMP I SETCAL
1279
1280

```

1281 1200 \*1200  
 1282 /  
 1283 /  
 1284 /  
 1285 /  
 1286 /  
 1287 /  
 1288 /  
 1289 /  
 1290 /  
 1291 /  
 1292 /  
 1293 /  
 1294 /  
 1295 /  
 1296 /  
 1297 /  
 1298 /  
 1299 /  
 1300 /

ASSEMBLY PHASE PAR  
 /  
 /PPAR1 IS ACTUALLY A PART OF THE BASIC ASSEMBLY  
 /ROUTINE ASM02,  
 /IT ASSEMBLES ALL PARAMETERS  
 /TYPES ARE: RC=00 ABSOLUTE CONSTANT  
 RC=01 RELOCATABLE ADDRESS  
 RC=05 CDF TO CURRENT FIELD  
 RC=06 CALL CONSTANT (#ARGS\*EXT,SYM,#)  
 /  
 /ALSO LITERALS USED IN ARG STATEMENTS  
 /SUCH LITS. ARE PUT IN LIT. POOL  
 /AND RC=01 ADDRESS OF LIT. PUT WHERE  
 /THE ARG STATEMENT OCCURS.  
 /ADDRESS PARAMETERS ARE ACTUALLY TAKEN  
 /CAR OF BY SUBR, PPAR3S,  
 /

1301	1200	3234	PPAR1, DCA	PPARY	/CLR OUTPUT CODE
1302	1201	1517	TAD I	PTCPR	/CK FOR LITERAL ARG OR PARAM.
1303	1222	0014	AND	K2	
1304	1203	7640	SZA CLA		
1305	1204	5241	JMP	PARLIT	/YES
1306	1205	1517	TAD I	PTCPR	/PT CODE WORD
1307	1206	0157	AND	K20	/IS IT PAR CONSTANT
1308	1207	7650	SNA CLA		
1309	1210	5236	JMP	PPAR3	/NO
1310	1211	1521	TAD I	PTSPPR	/YES ... ACTUAL CONSTANT
1311	1212	3123	DCA	TEM1	/TO DIRECTLY ADDRESSABLE LOC
1312	1213	1517	TAD I	PTCPR	/IS THIS A SPECIAL CONSTANT USED BY CALL
1313	1214	0162	AND	K100	
1314	1215	7650	SNA CLA		
1315	1216	5223	JMP	,+5	
1316	1217	7001	IAC		/YES, SET BANK TO CURRENT (NEW IN V15)
1317	1220	3147	DCA	BANK	
1318	1221	7001	IAC		/8 FORCE CODE=06
1319	1222	5227	JMP	,+5	
1320	1223	1517	TAD I	PTCPR	/IS THIS A CDF INSTRUCTION TO THIS BANK
1321	1224	0030	AND	K1000	
1322	1225	7650	SNA CLA		
1323	1226	5231	JMP	,+3	
1324	1227	1154	TAD	K5	
1325	1230	3234	DCA	PPARY	
1326	1231	4456	JMS I	WRITEP	
1327	1232	4436	JMS I	OUTBIN	/OUTPUT IT
1328	1233	0123	TEM1		/NO RELOCATION
1329	1234	0000	PPARY, 0		
1330	1235	7410	SKP		
1331	1236	4253	PPAR3, JMS	PPAR3S	/DO ALL WORK
1332	1237	5715	JMP I	PPAR5=1	
1333	1240	3443	SERALI, SRALT		
1334					
1335	1241	1521	PARLIT, TAD I	PTSPPR	/PUT LIT ON TAB.

```

/SABP, 815      PAL10  V135  23-SEP-69      11128  PAGE 24-1

1336  1242  3151      DCA    S1
1337  1243  7001      IAC
1338  1244  3150      DCA    S0
1339  1245  4640      JMS I  SERALI
1340  1246  0163      AND    K177      /GET PAGE ADDRESS
1341  1247  1077      TAD    PAG      /* PAGE BITS
1342  1250  3123      DCA    TEM1
1343  1251  2234      ISZ    PPARY      /CODE FOR RELOCATABLE ADDR
1344  1252  5231      JMP    PPARY*3
1345
1346  /
1347  /      SUBROUTINE TO ASSEMBLE PAR SYMBOL
1348  /
1349  /ASSEMBLE ADDRESS PARAMETER
1350  /SYMBOL MAY BE ABSOLUTE OR RELOCATABLE
1351  /NORMAL OR # REF.
1352  /IF SYMBOL IS YET UNDEFINED, AN ENTRY IS
1353  /MADE FOR IT & THE CURRENT ADDRESS IN THE
1354  /OCCURANCE TABLE.
1355  /
1356  /
1357  1253  0000      PPAR3S, 0
1358  1254  3302      DCA    PPARX      /CLR OUTPUT CODE
1359  1255  1057      TAD    ACTR      /WHCH TIME ARE WE ASSEMBLING THIS PAGE
1360  /NOTE! ACTR REMAINS 1 DURING PASS 2
1361  1256  7650      SNA CLA
1362  1257  5653      JMP I  PPAR3S      /FIRST TIME JUST RETURN
1363  1260  1521      TAD I  PTSPR      /SYMBOL
1364  1261  3142      DCA    AFS      /TO DIRRECTLY ADDRESSABLE LOCATION
1365  1262  4432      JMS I  OBSYM      /GET IT FROM MST
1366  1263  0142      AFS
1367  1264  1001      TAD    USE      /MST USE WORD
1368  1265  0045      AND    K400      /IS IT DEFINED YET
1369  1266  7650      SNA CLA
1370  1267  5384      JMP    PPAR4      /NO ... OCCURANCE
1371  1270  1001      TAD    USE      /MST USE WORD
1372  1271  0031      AND    K3000      /IS SYMBOL ABSOLUTE
1373  1272  7640      SZA CLA
1374  1273  2302      ISZ    PPARX      /OUTPUT RELOCATABLE
1375  1274  4722      JMS I  NSCHKI
1376  1275  1002      TAD    VAL      /INCREMENT IF # REF.
1377  1276  3123      PPAR6, DCA    TEM1
1378  1277  4456      JMS I  WRITEP
1379  1300  4436      JMS I  OUTBIN
1380  1301  0123      TEM1
1381  1302  0000      PPARX, 0
1382  1303  5653      JMP I  PPAR3S      /RETURN
1383  1304  1142      PPAR4, TAD    AFS      /SYMBOL
1384  1305  3716      DCA I  PPAR5      /TO SUBROUTINE LOCATION
1385  1306  1065      TAD    ILC      /CUR LOC
1386  1307  3717      DCA I  PPAR5+1    /TO SUBROUTINE LOC
1387  1310  4722      JMS I  NSCHKI
1388  1311  7106      CLL RTL
1389  1312  3721      DCA I  PPAR5+3
1390  1313  4720      JMS I  PPAR5+2      /SET ATEM2 FOR NORMAL OR # REFERENCE
                        /CREATE AN OCCURANCE

```

/SABR,815

PAL10

V135

23-SEP-69

11128 PAGE 24-2

1391	1314	5276	JMP	PPAR6	/OUTPUT ZERO WORD FOR LOADER
1392	1315	5746	ASM01		
1393	1316	3357	PPAR5,	ATEM3	
1394	1317	3360		ATEM4	
1395	1320	3321		L538	
1396	1321	3356		ATEM2	
1397	1322	6333	NSCHKI,	NSCHK	

```

1398
1399
1400
1401
1402
1403
1404
1405      1323  0000
1406      1324  3123
1407      1325  1123
1408      1326  7012
1409      1327  7012
1410      1330  7012
1411      1331  4335
1412      1332  1123
1413      1333  4335
1414      1334  5723
1415
1416      1335  0000
1417      1336  0161
1418      1337  7450
1419      1340  5344
1420      1341  5746
1421      1342  4454
1422      1343  5735
1423      1344  1165
1424      1345  5342
1425      1346  3367
1426

/
/TWO CHARACTER TYPEOUT
/FROM PACKED ASCII PAIR
/CALL WITH 6-BIT PAIR IN AC
/L61A ACTS AS SUBR FOR L61
/
L61,      0
          OCA      TEM1      /SAVE CHARACTERS
          TAD      TEM1
          RTR
          RTR      /SHIFT HIGH 6 BITS TO LOW
          RTR
          JMS      L61A      /MASK AND TYPE FIRST CHARACTER
          TAD      TEM1
          JMS      L61A      /MASK AND TYPE SECOND CHARACTER
          JMP I    L61      /RETURN

L61A,     0
          AND      K77      /MASK CHAR TO 6 BITS
          SNA
          JMP      L61B      /ZERO MEANS SPACE
          JMP I    L61CP
          JMS I    L61D      /HAVE DO SOME OF THIS WORK ON ANOTHER PAGE
          TAD      L61A      /TYPE CHAR
          JMP I    L61A      /RETURN
          TAD      K240      /SPACE
          JMP      L61D
L61CP,   L61C

```

```

1427 /
1428 /
1429 /
1430 /
1431 /
1432 /
1433 /
1434 /
1435 /
1436 /
1437 /
1438 /
1439 /
1440 /
1441 /
1442 /
1443 /
1444 /
1445 /
1446 /
1447 /
1448 /
1449 /
1450 /
1451 /
1452 /
1453 /
1454 /
1455 /
1456 /
1457 /
1458 /
1459 /
1460 /
1461 /
1462 /
1463 /
1464 /
1465 /
1466 /
1467 /
1468 /
1469 /

```

```

/
/
ROUTINE TO TEST CHARACTERS AND TAKE SELECTIVE EXITS
/
CALL IS
/
JMS I TEST
/
SORT LIST ADDR =1
/
BRANCH LIST ADDR = SORT LIST ADDR
/
RETURN IF ALL TESTS UNSUCCESSFUL
/
ASSUMES AC=0 & CHAR TO LOOK FOR IS IN CHR
/
/SORT ENDS UNSUCCESSFULLY AT
/NEGATIVE NUMBER FOLLOWING SORT LIST
/IF SORT IS SUCCESSFUL, A BRANCH IS
/TAKEN VIA BR, LIST ITEM CORRESPONDING
/TO MATCHING SORT LIST ITEM.
/
T5CHR, 0
CLA
TAD I T5CHR /GET SORT LIST ADDR =1
DCA X1 /AUTO=INDEX SORT LIST
ISZ T5CHR /MOVE ARG PTR
T5CHR2, CDF 10
TAD I X1 /GET SORT LIST ITEM
CDF 00
SPA
JMP T5CHR3 /NEG = END OF SORT LIST
CIA /COMPARE ITEM WITH CHR
TAD CHR
SZA CLA /0 = MATCH FOUND
JMP T5CHR2 /NO MATCH, TRY NEXT ITEM
TAD X1 /GET ADDR, OF MATCH
TAD I T5CHR /*BR, LIST ADDR = SORT LIST ADDR
DCA T5CHR /* PTR TO BR, LIST ITEM
CDF 10
TAD I T5CHR /GET BR, LIST ITEM
CDF 00
DCA T5CHR /* BRANCH PTR FOR THE MATCH
JMP I T5CHR /TAKE SUCCESSFUL EXIT
T5CHR3, ISZ T5CHR /NO MATCH ON LIST
CLA
JMP I T5CHR / RETURN UNSUCCESSFUL

```

```

1470      1400      *1400
1471      /
1472      /
1473      /
1474      1400  4427  /CALL,  JMS I  GETSYM  /GET NEXT INPUT ITEM
1475      1401  7000  NOP      /NOTHING THERE
1476      1402  7410  SKP      /SYMBOL
1477      1403  7610  SKP CLA  /CONSTANT
1478      1404  5220  JMP      CALERR /LITERAL
1479      1405  1061  TAD      CHR   /BREAK CHARACTER
1480      1406  1167  TAD      M254
1481      1407  7640  SZA CLA  /IS BREAK CHARACTER A COMMA
1482      1410  5220  JMP      CALERR /NO ... ERROR
1483      1411  1150  TAD      S0    /SAVE ARG COUNT
1484      1412  3362  DCA      ARGCT
1485      1413  2010  ISZ     X0    /PROHIBIT FLAGGING THE COMMA
1486      1414  4427  JMS I  GETSYM  /GET SUBROUTINE NAME
1487      1415  7410  SKP      /NONE THERE
1488      1416  5221  JMP      ,+3  /SYMBOL
1489      1417  7000  NOP      /CONSTANT
1490      1420  5575  CALERR, IERROR /LITERAL
1491      1421  4446  JMS I  SKIPL
1492      1422  4451  JMS I  SREST  /SEARCH EXTERNAL SYMBOL TABLE AND OUTPUT TV DEF
1493      1423  3370  DCA      PCALL1  /SAVE EXTERNAL SYMBOL NUMBER
1494      1424  1137  TAD      LFS
1495      1425  3441  DCA I  CALLFS
1496      1426  1362  TAD      ARGCT  /SET ARG COUNT IN DYNAMIC LOCATION
1497      1427  7041  CIA
1498      1430  3363  DCA      CALLSW  /SET CALL = ARG IN PROCESS SWITCH & COUNTER
1499      1431  1362  TAD      ARGCT  /COUNT OF ARGS
1500      1432  7104  RAL     CLL    /*2
1501      1433  1362  TAD      ARGCT  /*3 IN CASE USING LITERAL ARGS
1502      1434  1014  TAD      K2     /*2
1503      1435  4764  JMS I  PARG2  /CAN THE CURRENT PAGE HOLD IT
1504      1436  7410  SKP      /YES
1505      1437  4430  JMS I  INI    /NO ... INITIALIZE PT PTRS ... HAD TO ASSEMBLE PAG
1506      1440  1441  TAD I  CALLFS
1507      1441  3137  DCA      LFS
1508      1442  4421  JMS I  ICPLFS  /PROCESS COLLECTION LFS
1509      1443  1517  TAD I  PTCPR  /PT CODE WORD
1510      1444  1020  TAD      K30    /ADD CONSTANT BIT & PAR BIT
1511      1445  3517  DCA I  PTCPR  /TO PT CODE WORD
1512      1446  1366  TAD      PARG6  /PLACE JMS LINK INSTRUCTION
1513      1447  3521  DCA I  PTSPR  /AS CONSTANT
1514      1450  4352  JMS     PARG5  /INC PT PTRS & ASSMBL IF PASS 2
1515      1451  1017  TAD      K130   /CORRECT BIT PATTERN FOR CALL
1516      1452  3517  DCA I  PTCPR  /TO PT CODE WORD
1517      1453  7001  IAC      /A CALL FORCES BANK TO CURRENT
1518      1454  3073  DCA      LSTBNK / (NEW IN V15)
1519      1455  7001  IAC
1520      1456  3147  DCA      BANK
1521      1457  1362  TAD      ARGCT  /COUNT OF ARGS
1522      1460  7106  CLL     RTL    /TO HIGH ORDER AC
1523      1461  7006  RTL
1524      1462  7006  RTL

```

/SABR.R15

PAL14

V135

23-SEP-69

11128

PAGE 27-1

1525 1463 1370  
1526 1464 3521  
1527 1465 5327

TAD PCALL1  
DCA I PTSPR  
JMP ARGPP0

/OR IN EXTERNAL SYMBOL NUMBER  
/PLACE IN PT SYMBOL WORD  
/COMMON EXIT



```

1528 /
1529 /
1530 / ARG PSEUDO OPERATION
1531 PARG, JMS I GETSYM /GET NEXT INPUT ITEM
1532 IERROR /NOTHING THERE
1533 JMP ,+6 /SYMBOL
1534 JMP ,+3 /CONSTANT CODE IS 2
1535 JMS I SLITAB /PUT LIT ON TABLE
1536 CMA /LIT CODE IS 1
1537 TAD K2
1538 SKP
1539 TAD SYMBOL /PAR ADDRESS
1540 DCA AFS
1541 JMS I SKIPL
1542 JMS I CKCLSP /CK FOR TOO MANY ARGS
1543 /
1544 / ROUTINE TO PUT A CDF IN THE PAGE TABLE
1545 /
1546 TAD K30 /PT CODE WORD
1547 DCA I PTCPR /TO PT
1548 JMS I ICPLFS /PROCESS ANY LFS
1549 TAD K6201 /CDF
1550 DCA I PTSPR /TO PT SYMBOL WORD
1551 TAD M2
1552 TAD AFS /IS AFS A CONSTANT
1553 SNA /YES
1554 JMP ARGPP4 /IS AFS A LITERAL
1555 IAC
1556 SNA CLA /YES
1557 JMP ARGPP5 /NO ... SYMBOL ... GET ITS POINTERS TO MST
1558 JMS I OBSYM /AFS MST USE WORD
1559 AFS /IS IT A COMMON SYMBOL
1560 TAD USE
1561 AND K40
1562 SNA CLA /NO
1563 JMS CDFCHG /INCREMENT PT PTRS AND PUT OUT A PAR
1564 JMS ARGPP2
1565 TAD AFS
1566 DCA I PTSPR /PLACE SYMBOL IN PT SYMBOL WORD
1567 ARGPP0, JMS PARG5 /INC PT PTRS & ASSMBL IF PASS 2
1568 JMP I POPEXP /EXIT TO GET NEXT LINE
1569 /
1570 /
1571 ARGPP5, JMS CDFCHG
1572 JMS ARGPP2 /INCREMENT PTRS AND PUT OUT A PAR
1573 TAD K2 /SET LITERAL BIT
1574 JMP ,+3 /SAVE AS CONSTANT FROM HERE
1575 /
1576 ARGPP4, JMS ARGPP2 /INCREMENT PTRS AND PUT OUT A PAR
1577 TAD K20 /SET CONSTANT BIT
1578 TAD I PTCPR /PT CODE WORD
1579 DCA I PTCPR /FOR PROPER WORD
1580 TAD S0 /PLACE CONSTANT IN PROPER LOCATION
1581 DCA I PTSPR
1582 JMP ARGPP0

```

```

1583
1584
1585
1586      1544      0000
1587      1545      4352
1588      1546      1156
1589      1547      3517
1590      1550      5744
1591      1551      6211
1592
1593      1552      2000
1594      1553      1110
1595      1554      7640
1596      1555      4760
1597      1556      4761
1598      1557      5752
1599
1600      1560      5651
1601      1561      3551
1602
1603      1562      0000
1604      1564      0241
1605      1563      7000
1606      1564      2646
1607      1565      0771
1608      1562      1552
1609      1566      4033
1610      1567      7770
1611
1612
1613      1570      0000
1614      1571      1517
1615      1572      1030
1616      1573      3517
1617      1574      1521
1618      1575      1367
1619      1576      3521
1620      1577      5770
1621      1570      1570

/
/
/
ROUTINE TO INCREMENT POINTERS AND SET UP FOR A PAR IN THE PAGE TABLE
/
ARGPP2, 0
JMS      PARG5      /INC PT PTRS & ASSMBL IF PASS 2
TAD      K10
DCA I    PTCPR
JMP I    ARGPP2      /RETURN
K6201, CDF 10

ASMIF1, 2
TAD      PASS
SZA CLA
JMS I    ASM02S      /ASSMBL NOW IF LISTING PASS
JMS I    INC
JMP I    ASMIF1

ASM02S, ASM02
INC,    INCPT

ARGCT, 0
CALLFS=PRSYMP      /TEMP
CALLSW, 0
PARG2, IFFSUB
CKCLSP, CKCLS
PARG5=ASMIF1
PARG6, JMS      LINK
M10,    =10

/ROUTINE TO CHANGE CDF 00 TO CDF *
CDFCHG, 0
TAD I    PTCPR
TAD      K1000      /SET CDF * BIT IN P.T.
DCA I    PTCPR
TAD I    PTSPR      /CHANGE 6211
TAD      M10      /TO 6201
DCA I    PTSPR
JMP I    CDFCHG
PCALL1=CDFCHG      /TEMP

```

```

1622
1623
1624      1640      *1624
1625      /
1626      /      COMMN PSEUDO OPERATION
1627      /
1628      1640      4427      PCOMM1, JMS I   GETSYM      /GET ADDRESS FIELD SYMBOL
1629      1641      7000      NOP              /NOTHING THERE
1630      1642      7410      SKP              /SYMBOL THERE
1631      1643      7610      SKP CLA         /CONSTANT
1632      1644      5575      IERROR          /LITERAL
1633      1645      4446      JMS I   SKIPL
1634      1646      1137      TAD      LFS
1635      1647      7650      SNA CLA         /IS THERE AN LFS
1636      1610      5245      JMP      COMMN2  /NO ... JUST INCREMENT COUNTERS
1637      1611      4432      JMS I   OBSYM   /GET POINTERS TO LFS
1638      1612      0137      LFS
1639      1613      1001      TAD      USE     /MST USE WORD
1640      1614      0016      AND      K3      /SAVE SYMBOL LENGTH
1641      1615      1256      TAD      K440    /ADD CORRECT BITS
1642      1616      3001      DCA      USE     /FOR NEW MST USE WORD
1643      1617      1150      TAD      S0      /NO OF COMMON LOCATIONS
1644      1620      7650      SNA CLA         /ARE THERE ZERO
1645      1621      5241      JMP      COMMN1  /YES ... EQUIVALENCE OUTPUT
1646      1622      1076      TAD      HICOM   /NO ... HIGHEST COMMON LOCATION USED
1647      1623      1150      TAD      S0      /*SIZE OF THIS BLOCK
1648      1624      3123      DCA      TEM1    /FOR TENTATIVE NEW HIGHEST
1649      1625      1123      TAD      TEM1    /ACTUAL ADDRESS
1650      1626      0166      AND      K7600   /ARE WE OVERFLOWING ONTO THE LAST PAGE
1651      1627      1164      TAD      M7600
1652      1630      7650      SNA CLA
1653      1631      5577      SERROR          /YES ... ERROR
1654      1632      1076      TAD      HICOM   /LAST COMMON ASSIGNMENT
1655      1633      7001      IAC              /*1
1656      1634      3002      DCA      VAL     /GIVES NEW ADDRESS
1657      1635      1123      TAD      TEM1    /NEW HIGHEST COMMON LOCATION
1658      1636      3076      DCA      HICOM   /TO PROPER LOC
1659      1637      1002      COMMN0, TAD     VAL
1660      1640      5434      JMP I   NULLP   /GO GET NEXT LINE
1661      /
1662      /      EQUIVALENCE GENERATED COMMON OUTPUT
1663      /
1664      1641      1076      COMMN1, TAD     HICOM   /PLACE LAST COMMON ASSIGNMENT
1665      1642      7001      IAC              /*1
1666      1643      3002      DCA      VAL     /IN MST AS ADDRESS
1667      1644      5237      JMP      COMMN0  /EXIT
1668      /
1669      /      NON LOCATION FIELD SYMBOL COMMON ASSIGNMENT
1670      /
1671      1645      1076      COMMN2, TAD     HICOM   /LAST HIGHEST
1672      1646      1150      TAD      S0      /*CUR ASSIGNMENT
1673      1647      3076      DCA      HICOM   /FOR NEW HIGHEST
1674      1650      1076      TAD      HICOM   /NEW HIGHEST
1675      1651      0166      AND      K7600   /ARE WE OVERFLOWING ONTO THE LAST PAGE
1676      1652      1164      TAD      M7600

```

/SABR.615 PAL12 V135 23-SEP-69 11128 PAGE 29-1

1677	1653	7650		SNA CLA	
1678	1654	5577		SERROR	/YES ... ERROR
1679	1655	5237		JMP	/NO ... EXIT
1680	1656	0440	K442,	COMM0	0440

```

1681
1682
1683
1684      1657  1106
1685      1660  7740
1686      1661  5433
1687      1662  4426
1688      1663  4453
1689      1664  0756
1690      1665  1006
1691      1666  1061
1692      1667  7041
1693      1670  3371
1694      1671  3151
1695      1672  1010
1696      1673  3152
1697      1674  4426
1698      1675  7450
1699      1676  5575
1700      1677  1371
1701      1700  7650
1702      1701  5304
1703      1702  2151
1704      1703  5274
1705      1704  4426
1706      1705  7200
1707      1706  4446
1708      1707  4765
1709      1710  1151
1710      1711  7001
1711      1712  7110
1712      1713  4770
1713      1714  7610
1714      1715  4430
1715      1716  4766
1716      1717  4421
1717      1720  1151
1718      1721  7041
1719      1722  3151
1720      1723  1152
1721      1724  3010
1722      1725  3371
1723      1726  4426
1724      1727  0161
1725      1730  2371
1726      1731  7410
1727      1732  5342
1728      1733  7106
1729      1734  7006
1730      1735  7006
1731      1736  3153
1732      1737  7040
1733      1740  3371
1734      1741  7410
1735      1742  4351

/TEXT PSUEDO=OP
PTEXT,  TAD      FORFLG
        SMA SZA  CLA
        JMP I    DCIL1
        JMS I    GETCHR
        JMS I    TEST
                SL1=1
                BL1=SL1
        TAD      CHR
        CIA
        DCA      DELIN
        DCA      TEXCTR
        TAD      X0
        DCA      TEXPTR
TEX1,   JMS I    GETCHR
        SNA
TEXERR, IERROR
        TAD      DELIN
        SNA CLA
        JMP      TEX2
        ISZ     TEXCTR
        JMP     TEX1
TEX2,   JMS I    GETCHR
        CLA
        JMS I    SKIPL
        JMS I    PUSH
        TAD     TEXCTR
        IAC
        CLL RAR
        JMS I    IFFS
        SKP CLA
        JMS I    INI
        JMS I    POP
        JMS I    ICPLFS
        TAD     TEXCTR
        CIA
        DCA     TEXCTR
        TAD     TEXPTR
        DCA     X0
        DCA     BYTE
TEX5,   JMS I    GETCHR
        AND     K77
        ISZ     BYTE
        SKP
        JMP     TEX4
        CLL RTL
        RTL
        RTL
        DCA     TXSV
        CMA
        DCA     BYTE
        SKP
TEX4,   JMS     TEXSUB

/LOOK FOR STRING START
/SAVE OPENING DELINEATOR
/CLR CHAR CTR
/SAVE AUTO=INDEX TO START OF STR
/LOOK FOR END OF STRING
/TOO SOON END OF LINE
/THE END OF THE LINE
/KEEP STRING TALLY
/MOVE LINE PTR TO CHAR, AFTER DELINEATOR
/SAVE INFO FOR A MINUTE
/DIV BY 2
/SEE IF STR WILL FIT ON PAGE
/HAD TO ASSMBLI RE=INIT PT
/POP LINE INFO
/PROCESS LFS
/RE=INIT STRING INDEX
/SET FOR LEFT BYTE
/EXTRACT 6 BIT
/RIGHT BYTE
/MOVE LEFT
/SET PTR TO RT BYTE

```

```

/SABR,R15      PAL12      V135      23=SEP=69      11128      PAGE 30=1
1736      1743      2151      ISZ      TEXCTR
1737      1744      5326      JMP      TEX5      /NOT DONE
1738      1745      2371      ISZ      BYTE      /CK FOR ODD CHAR LEFT OVER
1739      1746      7410      SKP
1740      1747      4351      JMS      TEXSUB    /NO
1741      1750      5440      JMP I    POEXP     /YES
1742
1743      1751      0000      TEXSUB, 0
1744      1752      1153      TAD      TXSV      /COMBINE LEFT & RT BYTES
1745      1753      3521      DCA I    PTSPR
1746      1754      1020      TAD      K30      /PAR CONST BITS
1747      1755      1517      TAD I    PTCPR
1748      1756      3517      DCA I    PTCPR
1749      1757      1010      TAD      X0      /SAVE INDEX
1750      1760      3153      DCA      TXSV
1751      1761      4767      JMS I    ASIF      /INC PTRS & ASSMBL IF PASS 2
1752      1762      1153      TAD      TXSV     /RESTOR INDEX
1753      1763      3010      DCA      X0
1754      1764      5751      JMP I    TEXSUB
1755
1756      1765      4200      PUSH,    PUSHIN
1757      1766      4217      POP,     POPIN
1758      1767      1552      ASIF,    ASMIF1
1759      1770      2646      IFFS,    IFFSUB
1760      1771      0000      DELIN, 0
1761      2151      TEXCTR=S1
1762      2152      TEXPTR=S2
1763      2153      TXSV=S3
1764      1771      BYTE=DELIN
1765
1766
1767
1768
1769
1770
1771
1772      1772      0000      /
1773      1773      1110      /WRITE LINE IF IN PASS 2
1774      1774      7640      /
1775      1775      4455      WLNIF1, 0
1776      1776      5772      TAD      PASS      /WHICH PASS?
1777      1777      7640      SZA CLA
1778      1778      4455      JMS I    WLNPF     /LISTING
1779      1779      5772      JMP I    WLNIF1

```

```

1777 /
1778 2000 *2000
1779
1780 /READ & DECODE 1 LINE
1781 /IGNORES NULL LINES & COMMENT LINES
1782 / EXP=NON=0 IF NO OPERATION ON LINE (CONST, LIT,
1783 / OR ADDRESS ONLY)
1784 / SK=NON=0 IF SKIP INSTR.
1785 / UMIC=NON=0 IF OP CODE IS 6 OR 7
1786 / IB=NON=0 IF INSTR IS INDIRECT
1787 / NSGN=NON=0 IF AFS IS # SYMBOL
1788 / OP=OP CODE
1789 / LFS=PTR TO LFS IN SYM, TAB., IF ANY
1790 /*** AFS=2 IF CONSTANT PARAMETER OR CONST. AFS***
1791 /*** AFS=1 IF LITERAL PARAMETER OR LIT. AFS***
1792 / AFS=SYM, TAB. PTR. TO ADDRESS PARAMETER OR AFS
1793 /
1794 2000 0000 DCIL. 0
1795 2001 4764 RDL1. JMS I RLNP /READ IN A LINE
1796 2002 3137 DCA LFS /CLR STORAGE FOR LINE INFO
1797 2003 3145 DCA EXP
1798 2004 3140 DCA OP
1799 2005 3146 DCA SK
1800 2006 3141 DCA IB
1801 2007 3144 DCA NSGN
1802 2010 3143 DCA UMIC
1803 2011 3444 DCA I RECTI /CLR RECOUNT FLAG FOR CPLFS
1804 2012 2067 ISZ LINE /INC LINE COUNT
1805 2013 2010 ISZ X0 /DO NOT BACK UP X0
1806 2014 4427 JMS I GETSYM /READ 1ST ITEM
1807 2015 5372 JMP RDL11 /NULL LINE OR COMMENT
1808 2016 5227 JMP RDL7 /SYMBOL = POSSIBLE LFS
1809 2017 5222 JMP ,+3 /SET AFS=2 FOR CONSTANT
1810 2020 4447 RDL3. JMS I SLITAB /PUT LIT ON TAB
1811 2021 7040 CMA /AFS=1 FOR LITERAL
1812 2022 1014 RDL2. TAD K2
1813 2023 2145 ISZ EXP /SET PARAMETER EXPRESSION FLAG
1814 2024 3142 RDL5. DCA AFS
1815 2025 4446 JMS I SKIPL /SKIP TO END OF LINE
1816 2026 5600 JMP I DCIL /RETURN
1817 2027 1061 RDL7. TAD CHR /CK FOR COMMA
1818 2030 1167 TAD M254
1819 2031 7640 SZA CLA
1820 2032 5246 JMP RDL9 /NO, SHOULD BE SPACE,TAB,CR,OR ;
1821 2033 4770 JMS I WHATPP
1822 2034 7410 SKP
1823 2035 5575 IERROR /OP SYMBOL AS TAG
1824 2036 1003 TAD SYMBOL /NO, ENTER PTR TO LFS
1825 2037 3137 DCA LFS
1826 2040 2010 ISZ X0 /PROHIBIT FLAGGING COMMA
1827 2041 4427 JMS I GETSYM /GET ITEM AFTER LFS
1828 2042 5765 JMP I P00 /NULL AFTER LFS IS BSS0
1829 2043 5246 JMP RDL9 /SYMBOL=OP OR PARAMETER
1830 2044 5222 JMP RDL2 /CONSTANT
1831 2045 5220 JMP RDL3 /LITERAL

```

1832	2246	4770	RDL9,	JMS I	WHATPP	
1833	2247	5325		JMP RDL4		/NO-MUST BE ADDRESS PARAMETER
1834	2250	1201		TAD USE		/IS SYMBOLE A PSUEDO-OP
1835	2251	2160		AND	K40	
1836	2252	7640		SZA CLA		/NO
1837	2253	5374		JMP RDL18		/YES
1838	2254	1201		TAD USE		/IS SYMBOL AN MRI?
1839	2255	2045		AND	K400	
1840	2256	7650		SNA CLA		
1841	2257	5325		JMP RDL14		/NO=OPR OR I/O INSTR.
1842	2260	1201		TAD USE		/MRI=PUT OP SKIP BIT
1843	2261	2157		AND	K20	/INTO SKIP FLAG
1844	2262	3146		DCA SK		
1845	2263	1202		TAD	VAL	
1846	2264	3140		DCA	OP	
1847	2265	7410		SKP		
1848	2266	2141	RDL10,	ISZ IB		/ SET INDIRECT FLAG
1849	2267	4427		JMS I	GETSYSM	/READ SYMBOL AFTER MRI
1850	2270	5575		IERROR		/NOTHING THERE
1851	2271	5275		JMP RDL12		/SYMBOL
1852	2272	7201		IAC		/AFS=2 FOR CONST, AFS
1853	2273	7201		IAC		/AFS=1 FOR LIT,AFS
1854	2274	5224		JMP RDL5		/SKIP TO END OF LINE
1855						
1856	2275	1203	RDL12,	TAD SYMBOL		/CK FOR I
1857	2276	7241		CIA		
1858	2277	1363		TAD IBTI		/SYM. ADDR=I ADDR
1859	2100	7650		SNA CLA		/NOT I
1860	2121	5266		JMP RDL10		/IT IS I
1861	2122	4770		JMS I	WHATPP	
1862	2123	5306		JMP	,*3	
1863	2124	5575		IERROR		/AFS NOT USER SYMBOL
1864	2125	2145	RDL4,	ISZ	EXP	/ENTER HERE ON ADDRESS PAR,
1865	2126	1261		TAD	CHR	/CK FOR #
1866	2127	1371		TAD	M243	
1867	2110	7640		SZA CLA		
1868	2111	5315		JMP	,*4	
1869	2112	2144		ISZ	NSGN	/YES
1870	2113	4426		JMS I	GETCHR	/PREVENT FLAGGING #
1871	2114	7200		CLA		
1872	2115	1203		TAD SYMBOL		/SET PTR TO AFS
1873	2116	5224		JMP RDL5		
1874	2117	4770	RDL13,	JMS I	WHATPP	
1875	2120	5575		IERROR		/ELIM USER SYM
1876	2121	1201		TAD USE		/CK FOR OPR OR I/O INST.
1877	2122	2366		AND	K4440	/ELIM, MRI, PSUEDO
1878	2123	7640		SZA CLA		/OK
1879	2124	5575		IERROR		/ILLEGAL SYMBOL
1880	2125	1201	RDL14,	TAD USE		/COMPARE NEW MICRO-GRP
1881	2126	2367		AND	K300	/WITH OLD, IF ANY
1882	2127	7450		SNA		
1883	2130	5343		JMP RDL16		/GRP OK WITH ANYTHING
1884	2131	3123		DCA	TEM1	/NEW IS NOT
1885	2132	1143		TAD	MGRP	/CK OLD MGP, IF ANY
1886	2133	7450		SNA		/THERE IS ONE



/SAB# 815	PAL12	V135	23-SEP-69	11128	PAGE 31-2
1887	2134	5341	JMP RDL15		/0 OK WITH ANY NEW
1888	2135	7041	CIA	/COMPARE OLD	
1889	2136	1123	TAD TEM1	/WITH NEW	
1890	2137	7640	SEA CLA	/SAME=OK	
1891	2140	5575	IERROR	/ILLEGAL COMBINATION	
1892	2141	1123	RDL15, TAD TEM1	/MICRO=GRP=NEW	
1893	2142	3143	DCA MGRP		
1894	2143	1002	RDL16, TAD VAL	/OR NEW VALUE INTO OLD OP	
1895	2144	7040	CMA	/NOT A	
1896	2145	0140	AND OP	/AND B	
1897	2146	1002	TAD VAL	/*A	
1898	2147	3140	DCA OP	/*A OR B	
1899	2150	1001	TAD USE	/GET NEW SKIP BIT	
1900	2151	0157	AND K20		
1901	2152	7640	SEA CLA	/NON=SKIP	
1902	2153	2146	ISZ SK	/SET SKIP FLAG	
1903	2154	4427	JMS I GETSYH	/GET NEXT INSTR OF STRING	
1904	2155	5361	JMP RDL17	/NONE THERE = END OF STRING	
1905	2156	5317	JMP RDL13	/SYMBOL (AS EXPECTED)	
1906	2157	7000	NOP	/CONST, ILLEGAL	
1907	2160	5575	IERROR	/LIT ILLEGAL	
1908	2161	2143	RDL17, ISZ UMIC	/SET MICRO INST FLAG	
1909	2162	5224	JMP RDL5	/SKIP TO END OF LINE	
1910	2163	2425	IBTI, II		
1911		0143	MGRP=UMIC		
1912	2164	4600	RLNP, RLN		
1913	2165	2435	PB0, PBSS2		
1914	2166	0440	K4440, 440		
1915	2167	0300	K300, 300		
1916	2170	6737	WHATPP, WHATYP		
1917	2171	7535	M243, =243		
1918					
1919			/NULL LINE OR COMMENT		
1920					
1921	2172	4446	RDL11, JMS I SKIPL		
1922	2173	5434	JMP I NULLP		
1923					
1924			/PSUEDO=OP		
1925					
1926	2174	1002	RDL18, TAD VAL	/GET PSUEDO=OP ADDRESS	
1927	2175	3123	DCA TEM1	/STORE PTR	
1928	2176	5523	JMP I TEM1	/TO PROPER PSUEDO=OP HANDLER	
1929					

```

1930      2200      *2220
1931      /
1932      /END OF LINE PROCESSOR FOR COLLECTION PHASE
1933      /LOOKS FOR SEMI-COLON BEFORE A SLASH
1934      /STAR OR SLASH OR END (CR) MEANS NORMAL
1935      /END OF LINE. SEMI-COLON MEANS WE MUST
1936      /SAVE CURRENT ADDRESS IN LINE BUFFER FOR
1937      /START OF "NEXT" LINE.
1938      /THIS ROUTINE ALSO HAS THE IMPORTANT
1939      /FUNCTION OF WATCHING THE FORTR PSUEDO=OP
1940      /FLAG. IF FLAG IS ON L72 CAUSES LINE
1941      /TO BE TREATED AS NON-EXISTENT. L72 MUST
1942      /BE CALLED FOR EVER INSTR. LINE OR PSUEDO=OP
1943      /LINE (EXCEPT END, PAUSE, FORTR) BEFORE
1944      /ACTUAL PROCESSING OF THAT LINE BEGINS.
1945      /
1946      2200      0000      L72,      0
1947      2201      7410      SKP
1948      2202      4426      JMS I   GETCHR
1949      2203      4453      JMS I   TEST
1950      2204      0754      SL6-1
1951      2205      1013      BL7-SL6
1952      2206      5575      IERROR
1953      2207      1010      L72S,   TAD    X0
1954      2210      3107      DCA    SCOLON
1955      2211      1106      L72X,   TAD    FORFLG      /IF FLG=1 WE ARE SKIPPING
1956      2212      7740      SMA SZA CLA      /1ST HALF OF FORTRAN OUTPUT
1957      2213      5433      JMP I   OCIL1
1958      2214      5600      JMP I   L72
1959
1960      /
1961      /COMPUTE CURRENT PAGE SIZE
1962      /ENTER WITH AC=0
1963      /EXIT WITH PAGE SIZE IN AC
1964      /
1965      2215      0000      CPGES,  0
1966      2216      1102      TAD    APMSW      /OMIT PGEESC IF NON-AUTO PAGING
1967      2217      7650      SNA CLA
1968      2220      1111      TAD    PGEESC    /*SIZE OF ESCAPE REQUIRED
1969      2221      1122      TAD    PTSZE     /*SIZE OF PAGE TABLE
1970      2222      1071      TAD    LTSZE     /*SIZE OF LITERAL TABLE
1971      2223      1075      TAD    OPSCTR    /*OFF PAGE SYMBOL COUNTER
1972      2224      1074      TAD    OBACKR    /*OFF BANK ADDITION COUNTER
1973      2225      5615      JMP I   CPGES    /*IS DESIRED RESULT
1974
1975

```

```

1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986 2226 0000
1987 2227 1116
1988 2230 7440
1989 2231 5267
1990 2232 1116
1991 2233 1314
1992 2234 7700
1993 2235 5577
1994 2236 1116
1995 2237 7104
1996 2240 1173
1997 2241 3115
1998 2242 2116
1999 2243 1003
2000 2244 3515
2001 2245 1115
2002 2246 7001
2003 2247 3114
2004 2250 1110
2005 2251 7650
2006 2252 5265
2007 2253 4432
2008 2254 0003
2009 2255 1002
2010 2256 0166
2011 2257 7041
2012 2260 1077
2013 2261 7640
2014 2262 5265
2015 2263 1041
2016 2264 2226
2017 2265 3514
2018 2266 5626
2019
2020 2267 7041
2021 2270 3123
2022 2271 1173
2023 2272 3124
2024 2273 1524
2025 2274 7041
2026 2275 1003
2027 2276 7650
2028 2277 5305
2029 2300 2124
2030 2301 2124

/
/COLLECTION PHASE ROUTINE
/SEARCH PAGE SYMBOL TABLE FOR SYMBOL
/CALLING SEQUENCE: (ASSUMES SYM.ID.IS IN "SYMBOL")
/ JMS SPSTB
/ RETURN IF NOT FOUND (HAD TO ENTER IT)
/ RETURN IF FOUND
/ THE SEARCH IS AT L31; ENTERING DONE BY L32.
/
SPSTB, 0
TAD PSTSZE /SIZE OF PST
SZA /IS IT EMPTY
JMP L31 /NO
L32, TAD PSTSZE /IS PST FULL?
TAD PSTMAX
SMA CLA
SERROR /YES
TAD PSTSZE /SIZE OF PST*2
RAL CLL
TAD PSTBSE /*BASE
DCA PSTSPR /GIVES POINTER TO SYMBOL
ISZ PSTSZE / INCREMENT COUNTER
TAD SYMBOL /PHYSICALLY MOVE SYMBOL
DCA I PSTSPR
TAD PSTSPR
IAC /ADD 1
DCA PSTCPR /FOR CODE WORD POINTER
TAD PASS
SNA CLA
JMP L32A /ASSEMBLY: JUST ZERO CODE WORD
JMS I OBSYM /LISTING
SYMBOL
TAD VAL /CK IF SYM IS ON PAGE FORWARD REF.
AND K7600 /EXTRACT PAGE BITS
CIA
TAD PAG
SZA CLA
JMP L32A /NOT ON PAGE
TAD K4000 /ON PAGE: SET DEFINED BIT
ISZ SPSTB /& SET FOR "FOUND" RETURN
L32A, DCA I PSTCPR
JMP I SPSTB /NOT FOUND
/
L31, CIA /PLACE * COUNT OF TABLE
DCA TEM1 /IN INDEX LOC
TAD PSTBSE /PLACE TABLE BASE
DCA TEM2 /IN ADDRESS LOC
L31B, TAD I TEM2 /*SYMBOL
CIA
TAD SYMBOL /* REQUESTED SYMBOL
SNA CLA
JMP L31A /FOUND
ISZ TEM2 /NOT FOUND ... INCREMENT ADDRESS
ISZ TEM2

```

/SARR,815

FAL10

V135

23-SEP-69

11128 PAGE 33-1

2031 2302 2123  
 2032 2323 5273  
 2033 2324 5232  
 2034 2325 2226  
 2035 2326 1124  
 2036 2327 3115  
 2037 2310 1115  
 2038 2311 7001  
 2039 2312 3114  
 2040 2313 5626  
 2041  
 2042 2314 7700  
 2043

L31A,

ISZ TEM1  
 JMP L31B  
 JMP L32  
 ISZ SPSTB  
 TAD TEM2  
 DCA PSTSPR  
 TAD PSTSPR  
 IAC  
 DCA PSTCPR  
 JMP I SPSTB

/OVER  
 /NO ... TRY AGAIN  
 /YES ... PLACE ON TABLE  
 /FOUND ... INDEX FOR EXIT  
 /POINTER TO SYMBOL  
 /TO PROPER LOC  
 /SYMBOL POINTER  
 /+1  
 /GIVES CODE POINTER  
 /EXIT

/  
PSTMAX, =100

/MUST BE (PSTB=PTB)/2

2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098

2315 0000  
2316 1110  
2317 7640  
2320 5715  
2321 1367  
2322 3054  
2323 3151  
2324 3152  
2325 3153  
2326 1005  
2327 3011  
2330 1003  
2331 3012  
2332 1001  
2333 0016  
2334 7041  
2335 3126  
2336 6211  
2337 1412  
2340 6201  
2341 3411  
2342 2126  
2343 5336  
2344 1005  
2345 3012  
2346 1172  
2347 3126  
2350 1412  
2351 4423  
2352 2126  
2353 5350  
2354 1370  
2355 3054  
2356 5715  
  
2226  
2357 0000  
2360 3226  
2361 1226  
2362 1062

```

/
/OUTPUT 6 CHARACTER ASCII NAME
/TO BINARY TAPE
/FOR EXTERNAL SYMBOL DEFN.
/USED BY LFSCK (FOR RC=03) & 666 (FOR RC=17)
/OUTPUT GOES VIA TYPE PTR, BUT PTR IS
/CHANGED TO L66E SO CHAR CAN BE PUNCHED
/8 ADDED TO CK,SUM INSTEAD OF TYPED.
/668 IS USED ONLY IN PAS1=
/ASSEMBLY PHASE1
/
L68, 0
TAD PASS
SZA CLA
JMP I L68 /EXIT IF LISTING
TAD L668 /FOOL OUTPUT ROUTINE
DCA TYPE /SO IT THINKS PUNCH IS TTY
DCA S1
DCA S2
DCA S3
TAD AS0
DCA X1
TAD SYMROL /MST SYMBOL ADDRESS = 1
DCA X2 /TO AUTO X2
TAD USE /MST USE WORD
AND K3 /SYMBOL LENGTH
DCA TEM4 /=WORDS TO LOC
CDF 10
TAD I X2 /OBTAIN SYMBOL
CDF 00
DCA I X1
ISZ TEM4
JMP ,=5
TAD AS0
DCA X2
TAD M3
DCA TEM4
TAD I X2
JMS I CTYPE /PUNCH IT EXPANDED
ISZ TEM4 /MORE
JMP ,=3 /YES
TAD L660 /RESTORE TYPE ROUTINE
DCA TYPE
JMP I L68
/
/ DUMMY TYPE ROUTINE FOR EST TV DEFINITION
/
T8=SPSTB /SCRATCH LOC
L66E, 0
DCA T8 /SAVE CHAR
TAD T8
TAD CSUM /ADD CHAR TO BINARY CHECK SUM

```

```

/SABR,815      PAL1#  V135  23-SEP-69      1112R  PAGE 34-1
2099      2363  3062      DCA      CSUM
2100      2364  1226      TAD      T8
2101      2365  4442      JMS I    PUNCH      /OUTPUT CHAR ON BINARY TAPE
2102      2366  5757      JMP I    L66E      /RETURN
2103      2367  2357      L66R,   L66E
2104      2370  4772      L66D,   L64
2105
2106
2107
2108      /
2109      /INITIALIZATION THAT WONT FIT IN "INITA"
2110      /
2110      2371  0000      INITMR, 0
2111      2372  3776      DCA I    VALPTP
2112      2373  3777      DCA I    LLFSP
2113      2374  3067      DCA      LINE
2114      2375  5771      JMP I    INITMR
2115      2376  4545      VALPTP, VALPTR
2116      2377  5366      LLFSP,  LLFS

```

2117  
 2118  
 2119  
 2120  
 2121  
 2122  
 2123  
 2124  
 2125  
 2126  
 2127  
 2128  
 2129  
 2130  
 2131  
 2132  
 2133  
 2134  
 2135  
 2136  
 2137  
 2138  
 2139  
 2140  
 2141  
 2142  
 2143  
 2144  
 2145  
 2146  
 2147  
 2148  
 2149  
 2150  
 2151  
 2152  
 2153  
 2154  
 2155

2400 \*2400

```

/
/COLLECTION PHASE ROUTINE.
/SEARCH LITERAL TABLE FOR VALUE IN S0.
/PLACES LITERAL ON TABLE IS NOT THERE.
/OTHERWISE DOES NOTHING.
/
SLTAB, 0
    CLA
    TAD      LTSZE      /SIZE OF TABLE
    SZA      /IS TABLE EMPTY
    JMP      SLITB1    /NO ... SEARCH IT
    TAD      LTBSE      /BASE COLL. PHASE LIT. TABLE)
    TAD      LTSZE      /*DISPLACEMENT
    DCA      TEM1      /GIVES ADDRESS POINTER
    TAD      S0        /PHYSICALLY MOVE LITERAL
    CDF 10
    DCA I    TEM1
    ISZ     LTSZE      /INCREMENT COUNT
    CDF 00
    JMP I   SLTAB      /RETURN
SLITB1, CIA      /PLACE = COUNT
    DCA     TEM1
    CMA
    TAD     LTBSE      /LTBSE=1
    DCA     X1        /TO AUTO X1
SLITB2, CDF 10
    TAD I   X1        /=TABLE
    CDF 00
    CIA
    TAD     S0        /*REQUESTED LITERAL
    SNA CLA      /SAME
    JMP I   SLTAB      /YES, RETURN
    ISZ     TEM1      /MORE SYMBOLS TO TEST
    JMP     SLITB2    /YES
    JMP     SLTAB+5   /NO
    
```

2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195

2435 4446  
2436 1137  
2437 7650  
2440 5434  
2441 6211  
2442 1060  
2443 7640  
2444 5251  
2445 1064  
2446 3273  
2447 3673  
2450 2064  
2451 1137  
2452 3464  
2453 2673  
2454 6201  
2455 4421  
2456 7240  
2457 1066  
2460 3066  
2461 2060  
2462 7040  
2463 1517  
2464 0275  
2465 1164  
2466 3517  
2467 1114  
2470 3101  
2471 2064  
2472 5434  
2473 2000  
2474 3017  
2475 7577

```

/
/COLLECTION PHASE EQUIVALENCE PROCESSOR
/(FORMERLY CALLED BSS0 PROCESSOR)
/ENTERS SYMBOL ID, IN EQ. TAB
/
PBSS2, JMS I   SKIPL
        TAD   LFS           /LOCATION FIELD SYMBOL
        SNA CLA           /IS THERE ANY
        JMP I   NULLP      /NO
        CDF 10           /ARE WE PROCESSING A BSS 0 SEQUENCE
        TAD   BSSSW
        SZA CLA           /YES ... SKIP INITIALIZING
        JMP   ,+5         /NO ... INITIALIZE
        TAD   EQVIPR      /SET INPUT POINTER TO COUNT LOCATION
        DCA  CTPTR       /ZERO COUNT
        ISZ  EQVIPR      /INCREMENT INPUT POINTER
        TAD   LFS           /LOCATION FIELD SYMBOL
        DCA  EQVIPR      /PLACE LFS ON EQUIVALENCE TABLE
        ISZ  CTPTR       /INCREMENT COUNT
        CDF 00
        JMS I   ICPLFS    /PROCESS IT FOR COLLECTION
        CLA CMA           /REMOVE LFS FROM LFS TABLE
        TAD   LFSPTR
        DCA  LFSPTR
        ISZ  BSSSW       /SET BSS 0 IN PROGRSS SWITCH
                           /REMOVE EXTRA LFS BIT
        CMA
        TAD I   PTCPR
        AND   K7577X     /REMOVE EXTRA BSS0 BIT
        TAD   K200       /PLACE BSS0 BIT ON PT
        DCA  PTCPR
        TAD   PSTCPR     /SAVE PST ADDRESS IN CASE NEXT LINE OVERFLOWS
        DCA  EQVBIT
        ISZ  EQVIPR      /INCREMENT POINTER
        JMP I   NULLP    /EXIT FOR NEXT LINE
CTPTR,  EQUTB
LFSBSI, LFSBSS
K7577X, 7577

```



```

2196
2197
2198
2199
2200
2201
2202
2203      2476 0000
2204      2477 1517
2205      2500 0164
2206      2501 7650
2207      2502 5676
2208      2503 7040
2209      2504 3147
2210      2505 1323
2211      2506 3431
2212      2507 4324
2213      2510 7041
2214      2511 3332
2215      2512 4324
2216      2513 5674
2217      2514 4450
2218      2515 7000
2219      2516 4722
2220      2517 2332
2221      2520 5312
2222      2521 5676
2223      2522 5570
2224      2523 2514
2225
2226
2227
2228      2524 0000
2229      2525 6211
2230      2526 1463
2231      2527 2063
2232      2530 6201
2233      2531 5724

/
/ASSEMBLY PHASE EQUIVALENCE PROCESSOR
/EXTRACTS ENTIRE GROUP OF TAGS EQUIVALENCED
/TO SAME ADDRESS FROM TABLE & DEFINES
/THEM BY USING LFSCK FROM LFSBSS ON.
/
ANUMCK, 0
      TAD I   PTCPR      /PT CODE WORD
      AND    K200      /MASK OUT BSS 0 BIT
      SNA CLA      /IS IT A BSS 0 SYMBOL
      JMP I   ANUMCK   /NO ... EXIT
      CMA
      DCA     BANK      /BANK UNKNOWN
      TAD    AANUM7    /CHEAT RETURN ADDRESS
      DCA I   LFSCHK   /SO IT LOOKS LIKE A JMS FROM SOMEWHERE ELSE
      JMS    GNEQ      /GET COUNT
      CIA
      DCA    OPICTR    /NEGATE
      JMS    GNEQ      /SAVE IN INDEX LOC
      JMP I   LFSBSI   /GET SYMBOL
      ANUM7, JMS I   SPSTAB /PROCESS SYMBOL
      NOP                      /SET DEFINED BIT ON PST IN CASE
      JMS I   PSTD     /THIS WAS CARRIED OVER
      ISZ    OPICTR    /THE LAST PAGE
      JMP I   ANUM7=2   /ANY MORE ?
      JMP I   ANUMCK   /YES
      PSTD,  PSTDEF   /EXIT
      AANUM7, ANUM7

/
/      ROUTINE TO GET NEXT ITEM OFF EQUIVALENCE TABLE
/
GNEQ, 0
      CDF 10
      TAD I   EQVOPR
      ISZ    EQVOPR
      CDF 00
      JMP I   GNEQ

```

```

2234 /
2235 /SUBR, TO LIST A LINE IF IN PASS 2
2236 /
2237 /CALLING SEQUENCE: JMS I WRITEP
2238 / JMS I OUTBIN
2239 / LOCATION OF WORD TO OUTPUT
2240 / CONSTANT=RELOC. CODE
2241 / RETURN
2242 / (CALL TO OUTBIN MUST ALWAYS FOLLOW CALL
2243 / TO WRITE.)
2244 / ASSUMES CURRENT PC IS IN "ILC"
2245 / SETS FLAGS FOR PROPER LISTING
2246 / & CALLS WLN TO DO THE DRUDGE WORK.
2247 /
2248 2532 0000 WRITE, 0
2249 2533 1110 TAD PASS
2250 2534 7650 SNA CLA
2251 2535 5732 JMP I WRITE /PASS 1
2252 2536 2332 ISZ WRITE
2253 2537 1732 TAD I WRITE /ADDRESS OF VALUE
2254 2540 3136 DCA VALUE
2255 2541 1536 TAD I VALUE /GET VALUE
2256 2542 3136 DCA VALUE
2257 2543 2132 ISZ VFLG
2258 2544 2332 ISZ WRITE
2259 2545 1732 TAD I WRITE /GET RELOC. CODE
2260 2546 3134 DCA CODE
2261 2547 1134 TAD CODE
2262 2550 7450 SNA
2263 2551 5367 JMP WRITE2
2264 2552 7012 RTR /CONVERT TO 6BIT
2265 2553 7010 RAR
2266 2554 0155 AND K7
2267 2555 1376 TAD K60
2268 2556 7106 CLL RTL
2269 2557 7006 RTL
2270 2560 7006 RTL
2271 2561 3135 DCA ADDRES /TEM SAVE
2272 2562 1134 TAD CODE
2273 2563 0155 AND K7
2274 2564 1376 TAD K60
2275 2565 1135 TAD ADDRES
2276 2566 3134 DCA CODE
2277 2567 1065 WRITE2, TAD ILC /CURRENT ADDRESS
2278 2570 3135 DCA ADDRES
2279 2571 2133 ISZ AFLG
2280 2572 2130 ISZ LFLG
2281 2573 4455 JMS I WLN /LIST
2282 2574 2332 ISZ WRITE
2283 2575 5732 JMP I WRITE
2284 2576 0060 K60, 60
2285 2532 OPICTR=WRITE
2286 2577 4051 CDZSKP, JMS CDZSK

```

```

2287      2600      *2600
2288
2289      /
2290      /
2291      2600 4427      PBSS,   JMS I   GETSYM   /GET NEXT INPUT ITEM
2292      2601 5630      JMS I   PBSS2I  /NOTHING THERE (BSS 0)
2293      2602 7410      SKP     /SYMBOL
2294      2603 7610      SKP CLA  /CONSTANT
2295      2604 5575      IERROR  /LITERAL
2296      2605 4446      JMS I   SKIPL
2297      2606 4667      JMS I   IPSHIN  /SAVE ALL CURRENT INFO
2298      2607 5632      JMS I   PBSS4I  /CHECK BLOCK SIZE
2299      2610 4246      PBSS5,  JMS     IFFSUB  /CAN THIS FIT IN CORE
2300      2611 7610      SKP CLA  /YES
2301      2612 4430      JMS I   INI     /NO ... INITIALIZE PT POINTERS
2302      2613 4670      JMS I   IPOPIN  /POP CURRENT INFORMATION
2303      2614 3060      DCA     BSSSW   /CLEAR BSS0 SWITCH
2304      2615 4421      JMS I   ICPLFS  /PROCESS CURRENT LFS
2305      2616 1150      TAD     S0      /=BLOCK CONSTANT
2306      2617 7041      CIA
2307      2620 3231      DCA     TEM12   /TO INDEX LOCATION
2308      2621 1020      PBSS1,  TAD     KJ0    /PAR CONSTANT PT BIT STRUCTURE
2309      2622 1517      TAD I   PTCPR  /DONT LOSE LFS AND BSS 0 INFORMATION
2310      2623 3517      DCA I   PTCPR
2311      2624 4672      JMS I   ASMIF
2312      2625 2231      ISZ     TEM12  /DO THEM INDIVIDUALLY IF PASS 2
2313      2626 5221      JMP     PBSS1  /MORE
2314      2627 5440      JMP I   POPEXP /YES
2315      2630 2435      PBSS2I, PBSS2  /EXIT TO GET NEXT LINE
2316      2631 0000      TEM12, 0
2317      2632 6710      PBSS4I, PBSS4  /RESRV STORAGE CTR
2318

```

```

2319 /
2320 /
2321 /
2322 2633 4427 PIFF, JMS I GETSYM /GET NEXT INPUT ITEM
2323 2634 7000 NOP /NONE THERE
2324 2635 7410 SKP /SYMBOL
2325 2636 7610 SKP CLA /CONSTANT
2326 2637 5575 IERROR /LITERAL
2327 2640 4446 JMS I SKIPL
2328 2641 4671 JMS I WLNIF /LIST IF PASS 2
2329 2642 1150 TAD S0 /BINARY CONSTANT
2330 2643 4246 JMS IFFSUB /USE GLOBAL IFF SUBROUTINE
2331 2644 5433 JMP I DCIL1 /DIDNT HAVE TO ASSEMBLE PAGE
2332 2645 5673 JMP I RSTR1 /GO INITIALIZE
2333 /
2334 / IFF SUBROUTINE
2335 / CALL IS TAD PAGE INCREMENT
2336 / JMS IFFSUB
2337 / OK RETURN
2338 / HAD TO ASSEMBLE PAGE RETURN
2339 /
2340 /FUNCTION: TO SEE IF GIVEN NO. OF WORDS
2341 /WILL FIT ON CUR. PAGE: IF SO, RETURN
2342 /AT OK RET. OTHERWISE ASSEMBLE PAGE WE
2343 /HAVE NOW & INIT A NEW PAGE & RET. AT
2344 /SECOND RET. LOC.
2345 /IFFSUB IS USED BY CPAGE,BLOCK &
2346 /SEVERAL OTHER P=OPS
2347 /
2348 /
2349 2646 0000 IFFSUB, 0
2350 2647 3123 DCA TEM1 /SAVE INCREMENT
2351 2650 4422 JMS I ICPGES /COMPUTE PAGE SIZE
2352 2651 1123 TAD TEM1 /ADD INCREMENT
2353 2652 1275 TAD *201 /IS TOTAL .GT. PAGE SIZE (1 EXTRA BECAUSE
2354 2653 7750 SPA SNA CLA /PTSZE INCREMENTED BEFORE PSUEDO-OP
2355 2654 5646 JMP I IFFSUB /NO ... RETURN
2356 2655 7240 CLA CMA /YES ... DECREMENT PAGE TABLE SIZE
2357 2656 1122 TAD PTSIZ
2358 2657 7450 SNA /WATCH FOR AN EMPTY PAGE
2359 2660 5264 JMP ,+4 /LEAVE THINGS ALONE IF PAGE EMPTY
2360 2661 3122 DCA PTSIZ
2361 2662 4445 JMS I L551 /ASSEMBLE THE PAGE
2362 2663 4674 JMS I UPDATE
2363 2664 4676 JMS I FIXIL
2364 2665 2246 ISZ IFFSUB /INCREMENT FOR EXIT
2365 2666 5646 JMP I IFFSUB /RETURN
2366 2667 4200 IPSHIN, PUSHIN
2367 2670 4217 IPOPIN, POPIN
2368 2671 1772 WLNIF, WLNIF1
2369 2672 1552 ASMIF, ASMIF1
2370 2673 0205 RSTR1, RSTR1
2371 2674 3172 UPDATE, UDPAGE
2372 2675 0022 ICPGES=CPGESI
2373 2675 7577 *201, -201

```

/SABR,815 PAL10 V135 23=SEP=69 11128 PAGE 40=1  
2374 2676 5561 FIXIL, FIXILC

```

2375
2376
2377
2378
2379 2677 6200
2380 2700 0200
2381
2382 2721 1277
2383 2722 2320
2384 2723 1027
2385 2704 5307
2386 2705 1137
2387 2706 3770
2388 2707 1045
2389 2710 1027
2390 2711 1164
2391 2712 1162
2392 2713 3131
2393 2714 1110
2394 2715 7640
2395 2716 5356
2396 2717 4424
2397 2720 1131
2398 2721 4423
2399 2722 1371
2400 2723 4423
2401 2724 4423
2402 2725 1770
2403 2726 3151
2404 2727 6211
2405 2730 1551
2406 2731 2151
2407 2732 0016
2408
2409 2733 7040
2410 2734 3152
2411 2735 1172
2412 2736 3153
2413 2737 2152
2414 2740 7410
2415 2741 5373
2416 2742 6211
2417 2743 1551
2418 2744 6201
2419 2745 2151
2420 2746 4423
2421 2747 2153
2422 2750 5337
2423 2751 1372
2424 2752 4454
2425 2753 1067
2426 2754 4435
2427 2755 4424
2428 2756 1300
2429 2757 7650

/
/
K6200, 6200
FATAL, 0

ERRE, TAD K6200 /0500
ERRS, ISZ FATAL /SET FATAL ERROR SWITCH
TAD K600 /2300
JMP ,+3
ERRM, TAD LFS
DCA I LLFSI
TAD K400 /1500
ERRI, TAD K600 /1100
ERRC, TAD K200 /0300
ERRA, TAD K100 /0100
DCA EFLG
TAD PASS
SZA CLA
JMP ERREX /LISTING PASS
JMS I CRLF /TYPE CRLF
TAD EFLG /TYPE E#
JMS I CTYPE
TAD AT
JMS I CTYPE
JMS I CTYPE /TYPE 2 SPACES
TAD I LLFSI
DCA INDEX
CDF 10
TAD I INDEX
ISZ INDEX
AND K3
CMA
DCA COUNT
TAD M3 /SET 6 CHAR PRINT CTR
DCA MSCTR
ISZ COUNT
SKP
JMP ERR1 /NOT DONE YET WITH SYMBOL
CDF 10 /DONE I SEE IF SPACES NEEDED
TAD I INDEX
CDF 00
ISZ INDEX
JMS I CTYPE /TYPE THE LETTERS OR SPACES
ISZ MSCTR
JMP ,=11
ERR11, TAD SPPLUS /TYPE SPACE +
JMS I TYPE
TAD LINE /TYPE LINS FROM LAST LFS
JMS I CTYPE
JMS I CRLF
ERREX, TAD FATAL /FATAL ERROR?
SNA CLA

```

/SABR,P15

PAL10

V135

23-SEP-69

11128 PAGE 41-1

```
2430 2760 5363      JMP      ,+3
2431 2761 7402      HLT
2432 2762 5964      JMP I   K200
2433 2763 1113      TAD     PHASE
2434 2764 7640      SZA    CLA
2435 2765 5767      JMP I   ERR2
2436 2766 5434      JMP I   NULLP
2437 2767 5650      ERR2,  ASM02=1
2438
2439 2770 5366      LLFSI,  LLFS
2440          0151      INDEX=S1
2441          0152      COUNT=S2
2442          0153      MSCTR=S3
2443 2771 0124      AT,     0124
2444 2772 0253      SPPLUS, 253
2445
2446 2773 4423      ERR1,  JMS I   CTYPE
2447 2774 2153      ISZ   MSCTR
2448 2775 5373      JMP   ,=2
2449 2776 5351      JMP   ERR11
2450
```

/NO

/IF YES GO TO START AFTER WAIT  
/WHAT PHASE ARE WE IN

/ASSEMBLY  
/COLLECTION

/FILL OUT THE REST WITH SPACES

2451 3000 \*3000

2452  
 2453 /  
 2454 /ASSEMBLY PHASE ROUTINE TO CHECK FOR A  
 2455 /LOC, TAG (LFS) & PROCESS IF FOUND.  
 2456 /FUNCTION: (ASMBLY PHASE 1 - ACTR#0)  
 2457 /  
 2458 / (1) DEFINE TAG  
 2459 / (2) OUTPUT VALUE AT PAST OCCURANCES OF  
 2460 / FORWARD REF, TO THIS TAG  
 2461 / (3) CONDENSE OCC. TAB IF POSSIBLE.  
 2462 /  
 2463 / (ASMBLY PHASE 2 - ACTR#1  
 2464 / (THIS INCLUDES ALL OF PASS 2 AS  
 2465 / ACTR STAYS#1 IN PASS 2)  
 2466 / (1) OUTPUT EXT. SYM. DEFN. ON REL=TAPE  
 2467 /

2468 3000 0000 LFSCK, 0  
 2469 3001 1517 TAD I PTCPR /PT CODE WORD  
 2470 3002 7010 RAR  
 2471 3003 7620 SNL CLA /IS THERE A LFS  
 2472 3004 5600 JMP I LFSCK /NO ... RETURN  
 2473 3005 7040 CMA  
 2474 3006 3147 DCA BANK /BANK UNKNOWN  
 2475 3007 1110 TAD PASS /MOVE BACK PTR IF IN LISTING PASS  
 2476 3010 7041 CIA  
 2477 3011 1066 TAD LFSPTR  
 2478 3012 3066 DCA LFSPTR  
 2479 3013 6211 CDF 10  
 2480 3014 1466 TAD I LFSPTR /ACTUAL LFS  
 2481 3015 6201 CDF 00  
 2482 3016 2066 ISZ LFSPTR  
 2483 3017 3137 LFSBSS, DCA LFS  
 2484 3020 4432 JMS I OBSYM /OBTAIN LFS FROM MST  
 2485 3021 0137 LFS  
 2486 3022 1057 TAD ACTR /WHICH TIME ARE WE ASSEMBLING THIS PAGE  
 2487 /ACTR REMAINS 1 DURING PASS2  
 2488 3023 7640 SZA CLA  
 2489 3024 5232 JMP L67 /SECOND TIME: NO TEST  
 2490 3025 1001 TAD USE /CHK FOR MULTI DEF.  
 2491 3026 0045 AND K400  
 2492 3027 7640 SZA CLA  
 2493 3030 5763 MERROR /YES  
 2494 3031 5242 JMP LFSCK1  
 2495 3032 1001 L67, TAD USE /MST USE WORD  
 2496 3033 0164 AND K200 /(L67 HAS NO EFFECT IN PASS 2)  
 2497 3034 7650 SNA CLA /IS IT AN ENTRY  
 2498 3035 5242 JMP LFSCK1 /NO  
 2499 /  
 2500 /  
 2501 / EXTERNAL SYMBOL DEFINITION  
 2502 3036 4436 JMS I OUTBIN /OUTPUT BINARY DEFINITION  
 2503 3037 0065 ILC  
 2504 3040 0003 3  
 2505 3041 4760 JMS I L60I /PUNCH SYMBOL ON TAPE



2586	3042	1065	LFSCK1,	TAD	ILC	/CUR ILC
2587	3043	3002		DCA	VAL	/PLACE ON MST AS DEFINITION
2588	3044	1001		TAD	USE	/SYMBOL TABLE USE WORD
2589	3045	3361		AND	K7377	/MASK OUT DEFINED BIT
2510	3046	1045		TAD	K400	/ADD IN DEFINED BIT
2511	3047	3001		DCA	USE	/SYMBOL IS NOW DEFINED IN MST
2512			/			
2513			/			
2514			/			
2515			/			
2516	3050	1006		TAD	OTP	/SIZE OF OCCURANCE TABLE
2517	3051	7040		CMA		
2518	3052	1362		TAD	TOPCOR	
2519	3053	7450		SNA		
2520	3054	5600		JMP I	LFSCK	/RETURN IF EMPTY
2521	3055	7041		CIA		
2522	3056	3123		DCA	TEM1	/PLACE = SIZE IN INDEX LOC
2523	3057	1006		TAD	OTP	/PLACE TABLE BASE IN TEM2
2524	3060	3124		DCA	TEM2	/TEM2=PTR TO SYMBOL
2525	3061	6211		CDF	10	
2526	3062	3372	L51,	DCA	L51FLG	/CLR # SWITCH
2527	3063	2124		ISZ	TEM2	
2528	3064	7040		CMA		/CK 1ST WORD FOR # FLAG
2529	3065	1524		TAD I	TEM2	
2530	3066	7640		SEA	CLA	
2531	3067	5273		JMP	,+4	/NO
2532	3070	2372		ISZ	L51FLG	/YES, SET SWITCH
2533	3071	2124		ISZ	TEM2	/MOVE PTR & CTR
2534	3072	2123		ISZ	TEM1	/PAST EXTRA WORD
2535	3073	1524		TAD I	TEM2	/= OCCURRING SYMBOL
2536	3074	7041		CIA		
2537	3075	1003		TAD	SYMBOL	/*SYMBOL JUST DEFINED
2538	3076	7650		SNA	CLA	/ARE THEY EQUAL
2539	3077	5306		JMP	,+7	
2540	3100	2124		ISZ	TEM2	
2541	3101	2123	L51E,	ISZ	TEM1	/NO ... ARE THERE MORE
2542	3102	2123		ISZ	TEM1	/(2 WORDS PER OCCURANCE)
2543	3103	5262		JMP	L51	/YES
2544	3104	6201		CDF	00	
2545	3105	5600		JMP I	LFSCK	/NO ... RETURN
2546			/			
2547			/			
2548			/			
2549	3106	2124		ISZ	TEM2	
2550	3107	1524		TAD I	TEM2	/ACTUAL ADDRESS
2551	3110	6201		CDF	00	
2552	3111	3126		DCA	TEM4	
2553	3112	4425		JMS I	DUMMY	/OUTPUT ADDRESS AS ORIGIN
2554	3113	0126		TEM4		
2555	3114	0004		4		
2556	3115	1002		TAD	VAL	
2557	3116	1372		TAD	L51FLG	/ADD 1 IF # REF
2558	3117	3126		DCA	TEM4	
2559	3120	4425		JMS I	DUMMY	/OUTPUT SYMBOL VALUE AS RELOCATABLE DEF
2560	3121	0126		TEM4		

2561	3122	0001		1		
2562	3123	6211		ODF 10		
2563			/			
2564			/	NOW MOVE OCCURANCE TABLE UP 2		
2565			/			
2566	3124	1006	L51G,	TAD	OTP	
2567	3125	3126		DCA	TEM4	/SAVE
2568	3126	1124		TAD	TEM2	
2569	3127	3006		DCA	OTP	/RESET
2570	3130	1372		TAD	L51FLG	
2571	3131	1014		TAD	K2	
2572	3132	7041		CIA		
2573	3133	1124		TAD	TEM2	
2574	3134	7041		CIA		
2575	3135	1126		TAD	TEM4	
2576	3136	7450		SNA		
2577	3137	5301		JMP	L51E	/NOTHING TO MOVE
2578	3140	3125		DCA	TEM3	/CTR FOR MOVE UP
2579	3141	1125		TAD	TEM3	
2580	3142	7041		CIA		
2581	3143	1126		TAD	TEM4	
2582	3144	3126		DCA	TEM4	/TO PTR
2583	3145	1526	L51J,	TAD I	TEM4	
2584	3146	3406		DCA I	OTP	
2585	3147	7040		CMA		
2586	3150	1126		TAD	TEM4	
2587	3151	3126		DCA	TEM4	
2588	3152	7040		CMA		
2589	3153	1006		TAD	OTP	
2590	3154	3006		DCA	OTP	
2591	3155	2125		ISZ	TEM3	
2592	3156	5345		JMP	L51J	
2593	3157	5301		JMP	L51E	
2594						
2595						
2596	3160	2315	L68I, L68			
2597	3161	7377	K7377, 7377			
2598	3162	7600	TOPCOR, CORE1			
2599		5763	MERROR#JMP I ,			
2600	3163	2705				ERRM

```
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626

3164 7000
3165 6026
3166 6021
3167 5366
3170 7200
3171 5764

3172 7000
3173 7200
3174 1077
3175 1164
3176 3077
3177 5772
3172

/
/
/
L63, 0
PLS
PSF
JMP ,=1
CLA
JMP I L63

/SELECT IT
/WAIT FOR PUNCH
/EXIT WITH CLEAR AC

/
/UPDATE "PAGE" TO NEXT CORE PAGE
/I.E., PAGE #PAGE+200
/
UDPAGE, 0
CLA
TAD PAG
TAD K200
DCA PAG
JMP I UDPAGE

/OLD PAGE SETTING
/*SIZE OF ONE PAGE
/FOR NEW PAGE SETTING
/EXIT

L51FLG=UDPAGE
```

```

2627      3200      *3200
2628      /
2629      /
2630      /      SUBROUTINE TO OUTPUT ASSEMBLY PHASE LITERAL
2631      /      TABLE AND REMEMBER OCCURANCES
2632      3200      0000      OAPLT, 0
2633      3201      1057      TAD      ACTR      /SKIP IT THE 1ST TIME
2634      3202      7650      SNA CLA
2635      3203      5600      JMP I      OAPLT
2636      3204      4761      JMS I      SAVLNI      /PREVENT ANY LINE TYPEOUT
2637      3205      1070      TAD      LITSIZ      /SIZE OF TABLE
2638      3206      7450      SNA
2639      3207      5600      JMP I      OAPLT      /RETURN IF NONE
2640      3210      7041      CIA
2641      3211      3355      DCA      ATEM1      /PLACE * SIZE IN LOC
2642      3212      7240      CLA CHA
2643      3213      1045      TAD      LITBSE      /BASE = 1
2644      3214      3012      DCA      X2      /TO AUTO 12
2645      3215      1102      TAD      APMSW      /ARE WE IN AUTO PAGING MODE
2646      3216      7640      SZA CLA
2647      3217      5222      JMP      ,*3      /NO ... OK
2648      3220      1111      TAD      PGEESC      /YES ... SUBTRACT SIZE OF PAGE ESCAPE
2649      3221      7110      RAR CLL      /DIVIDED BY 2
2650      3222      1070      TAD      LITSIZ      /& SUBTR. LITSIZ
2651      3223      7041      CIA
2652      3224      3362      DCA      LITPTR      /TEM SAVE
2653      3225      1362      TAD      LITPTR
2654      3226      1077      TAD      PAG      /INITIALIZE PAGE ADDRESS
2655      3227      1164      TAD      K200
2656      3230      3065      DCA      ILC
2657      3231      1070      TAD      LITSIZ      /INIT LIT TBL PTR
2658      3232      7104      CLL RAL      /MULT BY 2
2659      3233      1045      TAD      LITBSE
2660      3234      3362      DCA      LITPTR
2661      3235      7041      IAC
2662      3236      7410      SKP
2663      /
2664      3237      2065      L52,  ISZ      ILC      /INC PAGE LOC
2665      3240      1172      TAD      M3      /DECREMENT LIT TBL PTR
2666      3241      1362      TAD      LITPTR
2667      3242      3362      DCA      LITPTR
2668      3243      6211      CDF 10
2669      3244      1762      TAD I      LITPTR      /CODE
2670      3245      3356      DCA      ATEM2
2671      3246      2362      ISZ      LITPTR
2672      3247      1762      TAD I      LITPTR      /SYMBOL OR LITERAL
2673      3250      3357      DCA      ATEM3
2674      3251      6201      CDF 00
2675      3252      7240      CLA CHA
2676      3253      1356      TAD      ATEM2      /IS CODE 1 ... LITERAL
2677      3254      7640      SZA CLA
2678      3255      5271      JMP      L53
2679      3256      4436      JMS I      OUTBIN      /NO ... SYMBOL
2680      3257      7065      ILC
2681      3260      7024      4

```

2682	3261	4456	JMS I	WRITEP	
2683	3262	4436	JMS I	OUTBIN	/OUTPUT LITERAL WITH NO RELOCATION
2684	3263	3357	ATEM3		
2685	3264	0000			
2686	3265	2355	L52A,	ISZ	ATEM1
2687	3266	5237	JMP	L52	/MORE
2688	3267	2065	ISZ	ILC	/YES
2689	3270	5600	JMP I	DAPLT	/SET FOR ESCAPE
2690	3271	4432	L53,	JMS I	DBSYN
2691	3272	3357	ATEM3		/NO ... RETURN
2692	3273	1001	TAD	USE	/OBTAIN SYMBOL FROM MST
2693	3274	0045	AND	K400	/MST USE WORD
2694	3275	7650	SNA CLA		/IS SYMBOL DEFINED
2695	3276	5315	JMP	L53A	/NO ... OCCURANCE
2696	3277	4436	JMS I	OUTBIN	/YES ... OUTPUT ORIGIN
2697	3300	0065	ILC		
2698	3301	0004	4		
2699	3322	1356	TAD	ATEM2	
2700	3303	0015	AND	K4	
2701	3324	7640	SZA CLA		
2702	3305	7001	IAC		/ITS A #
2703	3306	1002	TAD	VAL	/ACTUAL VALUE
2704	3307	3357	DCA	ATEM3	/TO DIRECTLY ADDRESSABLE LOC
2705	3310	4456	JMS I	WRITEP	
2706	3311	4436	JMS I	OUTBIN	/OUTPUT VALUE
2707	3312	3357	ATEM3		
2708	3313	0001	1		/RELOCATABLE
2709	3314	5265	JMP	L52A	/TRY MORE
2710	3315	1065	L53A,	TAD	ILC
2711	3316	3360	DCA	ATEM4	
2712	3317	4321	JMS	L53B	/PLACE ON OCCURANCE TABLE
2713	3320	5265	JMP	L52A	/TRY MORE
2714			/		
2715			/		
2716			/		
2717	3321	0000	L53B,	0	
2718	3322	1006	TAD	OTP	
2719	3323	7141	CIA CLL		
2720	3324	7001	IAC		/ALLOW FOR # FLAG
2721	3325	1007	TAD	STT	/+TOP OF MST
2722	3326	7630	SZL CLA		/OVERFLOW?
2723	3327	5577	SERROR		/YES ... OUT OF CORE
2724	3330	1171	TAD	M2	
2725	3331	1006	TAD	OTP	/OT SIZE = 2
2726	3332	3006	DCA	OTP	/GIVES ADDRESS ON OCCUR TABLE
2727	3333	1006	TAD	OTP	
2728	3334	3011	DCA	X1	
2729	3335	6211	ODF 10		
2730	3336	1357	TAD	ATEM3	/SYMBOL
2731	3337	3411	DCA I	X1	/TO OCCUR TABLE
2732	3340	1360	TAD	ATEM4	/PAGE ADDRESS
2733	3341	3411	DCA I	X1	/TO OCCUR TABLE
2734	3342	1356	TAD	ATEM2	/CK FOR #
2735	3343	0015	AND	K4	
2736	3344	7650	SNA CLA		

/SABP,815	PAL10	V135	23=SEP=69	11128	PAGE 44=2
2737	3345	5353	JMP	,+6	/NO
2738	3346	7001	IAC		/SET FLAG WORD
2739	3347	3406	DCA I	OTP	
2740	3350	7040	CMA		/MOVE DOWN PTR
2741	3351	1206	TAD	OTP	/PAST EXTRA WORD
2742	3352	3006	DCA	OTP	
2743	3353	6201	ODF 00		
2744	3354	5721	JMP I	L538	/TRY MORE
2745			/		
2746	3355	0000	ATEM1,	0	
2747	3356	0000	ATEM2,	0	
2748	3357	0000	ATEM3,	0	
2749	3360	0000	ATEM4,	0	
2750	3361	4743	SAVLNI,	SAVLIN	
2751	3362	0000	LITPTR,	0	
2752			/		
2753	3363	2106	PFORT,	ISZ FORFLG	/SET TO 1 FOR 1ST PASS THRU FORTRAN CODE
2754	3364	7000	NOP		/END PSUEDO SETS IT TO -1 TO NULLIFY
2755	3365	4446	JMS I	SKIPL	/SO BACK TO 0 FOR 2ND PASS
2756	3366	5434	JMP I	NULLP	
2757			/		
2758			/DO SOME WORK FOR L61A		
2759			/		
2760			/		
2761	3367	3124	L61C,	DCA TEM2	/SAVE 6-BIT CODE
2762	3370	1124	TAD	TEM2	
2763	3371	0160	AND	K40	
2764	3372	7650	SNA	CLA	
2765	3373	1162	TAD	K100	/ADD CORRECT LEADING BITS
2766	3374	1164	TAD	K200	
2767	3375	1124	TAD	TEM2	/ADD CHAR BITS
2768	3376	5777	JMP I	L610P	
2769	3377	1342	L61DP,	L61D	
2770					

2771	3400	*3400		
2772		/		
2773		/	ROUTINE TO PUNCH WORD AND RELOCATION BITS ON TAPE	
2774		/	CALL IS	
2775		/	JMS OUTBN	
2776		/	ADDRESS OF WORD	
2777		/	BITS	
2778		/		
2779	3400	0000	OUTBN,	0
2780	3401	7500	CLA CLL	
2781	3402	1600	TAD I	OUTBN /ADDRESS OF WORD
2782	3403	3141	DCA	OUT1
2783	3404	2200	ISZ	OUTBN
2784	3405	1600	TAD I	OUTBN /RELOCATION BITS
2785	3406	7006	RTL	/SHIFT LEFT 4
2786	3407	7006	RTL	
2787	3410	3127	DCA	OUT2 /SAVE
2788	3411	1110	TAD	PASS
2789	3412	7640	SZA CLA	
2790	3413	5230	JMP	OUTEX
2791	3414	1541	TAD I	OUT1 /ACTUAL WORD
2792	3415	7006	RTL	/ROTATE HIGH 4 BITS TO LOW
2793	3416	7006	RTL	
2794	3417	7004	RAL	
2795	3420	0242	AND	K17 /MASK
2796	3421	1127	TAD	OUT2 /ADD REL BITS
2797	3422	4232	JMS	SUM /ADD TO CHECK SUM
2798	3423	4442	JMS I	PUNCH /PUNCH IT
2799	3424	1541	TAD I	OUT1 /REMAINDER OF WORD
2800	3425	0241	AND	K377 /MASK TO 8 BITS
2801	3426	4232	JMS	SUM /ADD TO CHECK SUM
2802	3427	4442	JMS I	PUNCH /PUNCH IT
2803	3430	2200	OUTEX, ISZ	OUTBN /INDEX FOR EXIT
2804	3431	5600	JMP I	OUTBN /RETURN
2805	3432	0000	SUM,	0
2806	3433	3144	DCA	TSUM
2807	3434	1062	TAD	CSUM
2808	3435	1144	TAD	TSUM
2809	3436	3062	DCA	CSUM
2810	3437	1144	TAD	TSUM
2811	3440	5632	JMP I	SUM
2812		0144	TSUM=NSGN	
2813		0141	OUT1=IB	
2814		0127	OUT2=TEM5	
2815	3441	0377	K377,	377
2816	3442	0017	K17,	17

```

2817 /
2818 / ROUTINE TO SEARCH ASSEMBLY PHASE LITERAL TABLE
2819 / FOR 2 WORD ENTRY IN S0=S1
2820 / PLACES ON TABLE IF NOT THERE
2821 / RETURNS PAGE ADDRESS IN AC
2822 /
2823 3443 0000 SRALT, 0
2824 3444 7200 CLA
2825 3445 3124 DCA TEM2 /ZERO SEARCH COUNTER
2826 3446 6211 CDF 10
2827 3447 1070 TAD LITSIZ /NO OF ENTRYS
2828 3450 7440 SZA
2829 3451 5304 JMP L40 /NON ZERO ... SEARCH
2830 3452 1070 L39, TAD LITSIZ /NO OF ENTRYS = 1
2831 3453 7104 RAL CLL /MULTIPLY BY 2
2832 3454 1045 TAD LITRSE /ADD BASE OF TABLE
2833 3455 3123 DCA TEM1 /GIVES ADDRESS OF NEW ENTRY
2834 3456 2070 ISZ LITSIZ /INCREMENT COUNT
2835 3457 1150 TAD S0 /FIRST WORD
2836 3460 3523 DCA I TEM1 /TO TABLE
2837 3461 2123 ISZ TEM1 /INCREMENT ADDRESS
2838 3462 1151 TAD S1 /SECOND WORD
2839 3463 3523 DCA I TEM1 /TO TABLE
2840 3464 1070 TAD LITSIZ /ENTRY NO
2841 /
2842 / COMPUTE PAGE ADDRESS FROM DISPLACEMENT IN TABLE
2843 / AND STATUS OF AUTOMATIC PAGING MODE SWITCH AND SIZE OF PAGE
2844 / ESCAPE REQUIRED
2845 /
2846 3465 3123 L40A, DCA TEM1 /SAVE LOCATION IN TABLE
2847 3466 1110 TAD PASS
2848 3467 7640 SZA CLA
2849 3470 5332 JMP L40C /LISTING
2850 3471 1111 TAD PGEESC /TAKE PAGE ESCAPE SIZE
2851 3472 7110 RAR CLL /DIVIDE BY 2
2852 3473 3124 DCA TEM2
2853 3474 1102 TAD APMSW /ARE WE IN AUTOMATIC PAGING MODE
2854 3475 7650 SNA CLA
2855 3476 1124 TAD TEM2 /YES COUNT ESC, WORD
2856 3477 1123 L40D, TAD TEM1 /NO ... COMPUTE PAGE ADDRESS
2857 3500 7041 CIA /BY STRAIGHT COMPLEMENTATION METHOD
2858 3521 0241 AND K377 /MASK
2859 3522 6201 CDF 00
2860 3523 5643 JMP I SRALT /EXIT
2861 /FOR AUTO PAGING MODE
2862 /
2863 3524 7041 L40, CIA
2864 3525 3123 DCA TEM1 /* NO OF ENTRYS TO LOC
2865 3526 7240 CLA CMA
2866 3527 1045 TAD LITRSE /BASE OF TABLE = 1
2867 3510 3011 DCA X1 /TO AUTO 10
2868 3511 2124 L41, ISZ TEM2 /INCREMENT SEARCH COUNTER
2869 3512 1411 TAD I X1 /* FIRST WORD FROM TABLE
2870 3513 7041 CIA
2871 3514 1150 TAD S0 /*FIRST COMP WORD

```



/SABR,815

FAL1.

0135

23-SEP-69

11128 PAGE 46-1

2872	3515	7542		SZA CLA		
2873	3516	5326		JMP	L400	/NO MATCH
2874	3517	1411		TAD I	X1	/-SECOND TABLE WORD
2875	3520	7241		CIA		
2876	3521	1151		TAD	S1	
2877	3522	7640		SZA CLA		
2878	3523	5327		JMP	,+4	/NO MATCH
2879	3524	1124		TAD	TEM2	/MATCH ... CTR TO AC
2880	3525	5265		JMP	L400	/RETURN
2881	3526	2011	L400,	ISZ	X1	/INCREMENT FOR NO SECOND COMPARISON
2882	3527	2123		ISZ	TEM1	/OVER
2883	3530	5311		JMP	L41	/NO ... TRY MORE
2884	3531	5252		JMP	L39	/YES ... PLACE ON TABLE
2885	3532	6201	L400,	CPF 00		
2886	3533	1736		TAD I	REDUCP	/GET PAGE ESC COMPUTED BY A1
2887	3534	7110		CLL RAR		
2888	3535	5277		JMP	L400	
2889	3536	5524	REDUCP,	REDUCE		
2890						

```

2891 /
2892 / HAS COMMON BEEN PUNCHED YET SUBROUTINE
2893
2894 /IF IT HAS ALREADY BEEN PUNCHED, EXIT
2895 /IF NOT, PUNCH IT & SET FLAG
2896 /THIS ROUTINE IS CALLED ONLY ONCE PER PROGRAM
2897 /BUT IT COULD BE CALLED FROM ANY OF SEVERAL PLACES
2898 /
2899 HCBPS, 7
2900 3537 0200 TAD CPSW /COMMON PUNCHED SWITCH
2901 3540 1104 SNA CLA /HAS IT BEEN PUNCHED
2902 3541 7650 JMP I HCBPS /YES ... RETURN
2903 3542 5737 DCA CPSW /NO ... CLEAR SWITCH
2904 3543 3104 JMS I OUTBIN /AND PUNCH HIGHEST COMMON ASSIGNED
2905 3544 4436 HICOM
2906 3545 0076 12
2907 3546 0012
2908 3547 5737 JMP I HCBPS /EXIT
2909
2910 DUMSUB, JMS DUMS
2911
2912 /
2913 /INCREMENT PAGE TABLE PTRS
2914 /TO PREPARE FOR NEXT INSTRUCTION (OR PARAMETER)
2915 /
2916 INCPT, 0
2917 3551 0200 ISZ PTSZE /INCREMENT PAGE TABLE SIZE
2918 3552 2122 JMS I ISZPT2
2919 3553 4767 DCA EQVBIT /CLR
2920 3554 3101 DCA I PTCPR /INITIALIZE PAGE TABLE CODE WORD
2921 3555 3517 DCA I PTPSR /INITIALIZE PAGE TABLE SYMBOL WORD
2922 3556 3521 CDF 10
2923 3557 6211 DCA I PTOPR /INITIALIZE PT OP CODE POINTER
2924 3560 3520 CDF 00
2925 3561 6201 TAD CURSKP /MOVE CURRENT SKIP INSTRUCTION INDICATOR
2926 3562 1146 DCA LSTSKP /TO LAST INSTRUCTION SKIP INDICATOR
2927 3563 3072 TAD BANK /MOVE CURRENT BANK
2928 3564 1147 DCA LSTBNK /TO LAST BANK
2929 3565 3073 JMP I INCPY /RETURN
2930 3566 5751
2931 3567 6701 ISZPT2, ISZPT
2932 //
2933 //FOLLOWING CODE MOVED HERE TO MAKE ROOM FOR V03 IN ASME3
2934 ASMEXT, JMS I OUTSKP /YES, OUTPUT SKP
2935 3570 4437 TAD ILC /GET PG.LOC.PTR.
2936 3571 1065 AND K177
2937 3572 0163 TAD K5204 /OUTPUT JMP ,+4
2938 3573 1377 DCA TEM1
2939 3574 3123 JMP I ,+1
2940 3575 5776 REEASH
2941 3576 6016
2942 3577 5204 K5204, 5204

```

```

2942
2943      3600      *3600
2944
2945      /          ABSYM PSEUDO OPERATOR
2946      /
2947      3600      1045      PABSYM, TAD      K400
2948      3671      4242      JMS      DEFSUB
2949      3672      7200      CLA
2950      3603      5226      JMP      DEF1
2951
2952      /SKPDF & OPDEF PSUEDO-OPS
2953
2954
2955      3624      1157      SKPDEX, TAD      K20      /PUT IN SKIP BIT
2956      3605      1241      OPDEX,  TAD      K3010   /STANDARD OP BITS
2957      3606      4242      JMS      DEFSUB
2958      3627      7100      CLL
2959      3610      0247      AND      K7000   /CK TYPE OF INST
2960      3611      1043      TAD      K2000
2961      3612      7650      SNA CLA
2962      3613      5226      JMP      DEF1      /IOT
2963      3614      7420      SNL
2964      3615      5236      JMP      DEF3      /MRI
2965      3616      1150      TAD      S0       /OPR, BUT WHICH GRP?
2966      3617      0240      AND      K401
2967      3620      7110      CLL RAR
2968      3621      7650      SNA CLA
2969      3622      5234      JMP      DEF2      /GRP1
2970      3623      7430      SEL
2971      3624      1162      TAD      K100     /GRP3
2972      3625      1164      TAD      K200     /GRP2
2973      3626      1127      DEF1,  TAD      TEM5
2974      3627      3001      DCA      USE
2975      3630      1150      TAD      S0
2976      3631      3002      DCA      VAL
2977      3632      1002      TAD      VAL
2978      3633      5434      JMP I    NULLP
2979      3634      1162      DEF2,  TAD      K100
2980      3635      5226      JMP      DEF1
2981      3636      1045      DEF3,  TAD      K400
2982      3637      5226      JMP      DEF1
2983      3640      0401      K401,  401
2984      3641      3010      K3010, 3010
2985
2986
2987
2988      /
2989      /UTILITY FOR PABSYM & OPDEX
2990      /CALL WITH MST CODE WORD EXCEPT BITS 10-11
2991      /IN AC, EXITS WITH SYMBOL VALUE
2992      /AS DEF, BY SOURCE TAPE IN S0 & IN AC.
2993
2994      3642      0000      /
2995      3643      3127      DEFSUB, 0
2996      3644      4427      DCA      TEM5
2997      3645      5260      JMS I    GETSYM /GET THE SYMBOL NAME
2998      JMP      DEFERR /NULL

```

/SABR,815	PAL10	V135	23-SEP-69	11128	PAGE 48-1
2997	3646	5251	JMP	,+3	/SYMBOL
2998	3647	7000	K7000,	NOP	
2999	3650	5260	JMP	DEFERR	/CONST. OR LIT.
3000	3651	1150	TAD	S0	/ADD IN SYM LENGTH
3001	3652	1127	TAD	TEMS	
3002	3653	3127	DCA	TEMS	
3003	3654	4427	JMS I	GETSYM	/GET VALUE
3004	3655	7000	NOP		/NULL
3005	3656	7410	SKP		/SYMBOL
3006	3657	7410	K7410,	SKP	/CONST
3007	3660	5575	DEFERR,	IERROR	/LIT.
3008	3661	4446	JMS I	SKIPL	
3009	3662	1150	TAD	S0	/VALUE
3010	3663	5642	JMP I	DEFSUB	

```

3011
3012
3013
3014
3015
3016
3017
3018
3019 3664 2000 L62, 2
3020 3665 7104 CLL RAL /PUSH THRU LINK
3021 3666 3123 DCA TEM1
3022 3667 1304 TAD M4 /SET CTR
3023 3670 3124 DCA TEM2
3024 3671 1123 L62A, TAD TEM1
3025 3672 7006 RTL
3026 3673 7004 RAL
3027 3674 3123 DCA TEM1
3028 3675 1123 TAD TEM1
3029 3676 0155 AND K7
3030 3677 1305 TAD K260
3031 3700 4454 JMS I TYPE
3032 3701 2124 ISZ TEM2
3033 3702 5271 JMP L62A
3034 3703 5664 JMP I L62
3035 3704 7774 M4, -4
3036 3705 0260 K260, 0260
3037
3038
3039 /
3040 /DUMMY OUTPUT ROUTINE
3041 /REPLACES OUTBN DURING ASMBLY PHASE 1
3042 /CALLING SEQUENCE: JMS DUMMY
3043 / ADDR, OF ARG
3044 / RELOC, CONST.
3045 / RETURN
3046 /NOTE: SAME CALLING SEQ. AS OUTBN
3047 /
3048 3706 0000 DUM, 0
3049 3707 7300 CLA CLL
3050 3710 2306 ISZ DUM /INDEX FOR PROPER EXIT
3051 3711 2306 ISZ DUM /INDEX FOR PROPER EXIT
3051 3712 5706 JMP I DUM

```

```

3052 /
3053 /
3054 / ROUTINE TO SEARCH EXTERNAL SYMBOL TABLE
3055 / FOR CUR SYMBOL = RETURNS EXTERNAL SYMBOL
3056 / NUMBER IN AC = PLACES SYMBOL ON TABLE
3057 / AND OUTPUTS SIN CODE FOR TV IF NOT ON TABLE
3058 3713 0000 L66, 0
3059 3714 1100 TAD ESTSIZ /IS TABLE FULL?
3060 3715 1375 TAO M100
3061 3716 7700 SMA CLA
3062 3717 5577 SERROR /YES
3063 3720 1100 TAD ESTSIZ /SIZE OF EST
3064 3721 7440 SZA /IS TABLE EMPTY
3065 3722 5341 JMP L66A1 /NO ... SEARCH IT
3066 3723 7240 L66A3, CLA CMA /YES ... PLACE SYMBOL ON IT
3067 3724 2100 ISZ ESTSIZ /INCREMENT TABLE SIZE
3068 3725 1162 TAD BSEEST /BASE
3069 3726 1100 TAD ESTSIZ /*SIZE
3070 3727 3123 DCA TEM1 /GIVES ADDRESS OF NEW ENTRY
3071 3730 1003 TAD SYMBOL /PHYSICALLY PLACE ON TABLE
3072 3731 6211 CDF 10
3073 3732 3523 DCA I TEM1
3074 3733 6201 CDF 00
3075 3734 1001 TAD USE /MST CODE WORD
3076 3735 0016 AND K3 /SAVE LENGTH
3077 3736 1043 TAD K2000 /ADD EXTERNAL BITS
3078 3737 3001 DCA USE
3079 3740 5365 JMP L66A /GO TO PUNCH TV DEF
3080 3741 7041 L66A1, CIA
3081 3742 3124 DCA TEM2 /PLACE =SIZE IN INDEX LOC
3082 3743 3125 DCA TEM3 /ZERO COUNT
3083 3744 7240 CLA CMA
3084 3745 1162 TAD BSEEST /BASE OF EST = 1
3085 3746 3011 DCA X1 /TO AUTO X1
3086 3747 2125 L66A2, ISZ TEM3 /INCREMENT COUNT LOC
3087 3750 6211 CDF 10
3088 3751 1411 TAD I X1 /*TABLE SYMBOL
3089 3752 6201 CDF 00
3090 3753 7041 CIA
3091 3754 1003 TAD SYMBOL /* CUR SYMBOL
3092 3755 7650 SNA CLA /COMPARE
3093 3756 5362 JMP ,+4 /SAVE
3094 3757 2124 ISZ TEM2 /NOT SAME ... ANY MORE
3095 3760 5347 JMP L66A2 /YES ... KEEP TRYING
3096 3761 5323 JMP L66A3 /NO ... PLACE ON TABLE
3097 3762 1125 TAD TEM3 /PLACE COUNT IN AC
3098 3763 5713 JMP I L66 /RETURN
3099 /
3100 / OUTPUT BINARY EXTERNAL SYMBOL
3101 /
3102 3764 3537 HCBPS
3103 3765 4764 L66A, JMS I ,=1 /CHECK TO SEE IF COMMON HAS BEEN PUNCHED
3104 3766 4436 JMS I OUTBIN /TV DEF FOR 1 SYMBOL
3105 3767 3776 K1
3106 3770 0017 17

```

/SARD,815

FAL1e

V135

23-SEP-69

11128 PAGE 50-1

3107	3771	4774		JMS I	L62A1	/PUNCH ASCII CHARS
3108	3772	1100		TAD	ESTSIZ	/EST NO TO AC
3109	3773	5713		JMP I	L66	/RETURN
3110	3774	2315	L62A1,	L68		
3111	3775	7700	M100,	=100		
3112	3776	0001	K1,	1		
3113						

3114 4000 \*4000  
 3115  
 3116  
 3117  
 3118  
 3119  
 3120  
 3121  
 3122  
 3123  
 3124  
 3125  
 3126  
 3127

/  
 /SYMBOL TABLE LISTING ROUTINE  
 /TYPES: TABLE FROM "STTP" UP  
 /WITH NAME=VALUE=FLAG  
 /POSSIBLE FLAGS ARE: EXT, COM, UNDF, ABS, OP  
 /FLAGS TYPED BY "STFT"  
 /TABLE LISTED ALPHABETICALLY WITH NUMERIC  
 /CHARACTERS: ,GT, ALPHABETIC  
 /

3128	4000	0000	PRSYM,	0		
3129	4001	3125		DCA	PFLG	/CLR PRSYM=PASS FLAG
3130	4002	1777		TAD I	LSTDEP	
3131	4003	7650		SNA	CLA	
3132	4004	5207		JMP	,=3	
3133	4005	1042		TAD	PUNCH	/LIST ON H.S. PUNCH
3134	4006	3054		DCA	TYPE	
3135	4007	4424		JMS I	CRLF	
3136	4010	1375	PRS1,	TAD	PST	/INIT SPTR AT TOP OF PERM. S.T.
3137	4011	3127		DCA	SPTR	
3138	4012	1172		TAD	M3	/FILL S1,S2,S3 WITH 7777'S (MAX)
3139	4013	3150		DCA	ALEN	
3140	4014	1005		TAD	APTR	
3141	4015	3011		DCA	X1	
3142	4016	7040		CMA		
3143	4017	3411		DCA I	X1	
3144	4020	2150		ISZ	ALEN	
3145	4021	5216		JMP	,=3	
3146	4022	1016		TAD	K3	/AND LENGTH=3
3147	4023	3150		DCA	ALEN	
3148	4024	3126		DCA	FOUND	/CLR EXIT FLAG
3149	4025	1007	PRS2,	TAD	SYT	/HAS SEARCH HIT END OF TABLE?
3150	4026	7041		CIA		
3151	4027	1127		TAD	SPTR	
3152	4030	7650		SNA	CLA	
3153	4031	5337		JMP	PRS7	/YES, USE THE A=SYM WE HAVE
3154	4032	4432		JMS I	OBSYM	/NO, GET NEXT MST ENTRY
3155	4033	0127			SPTR	
3156	4034	1001		TAD	BCODE	/EXTRACT LENGTH
3157	4035	0016		AND	K3	
3158	4036	3137		DCA	BLEN	
3159	4037	1003		TAD	BPTR	/INDEX NEW ENTRY
3160	4040	3012		DCA	X2	
3161	4041	1137		TAD	BLEN	/SET ENTRY CTR
3162	4042	7041		CIA		
3163	4043	3140		DCA	BCTR	
3164	4044	1150		TAD	ALEN	/SET A=SYM CTR
3165	4045	7041		CIA		
3166	4046	3141		DCA	AACTR	
3167	4047	1005		TAD	APTR	/INDEX A=SYM
3168	4050	3011		DCA	X1	



3169 4251 1125  
 3170 4252 7640  
 3171 4253 5257  
 3172 4254 1221  
 3173 4255 7376  
 3174 4256 3421  
 3175 4257 1401  
 3176 4260 7710  
 3177 4261 5332  
 3178 4262 1411  
 3179 4263 7141  
 3180 4264 6211  
 3181 4265 1412  
 3182 4266 6221  
 3183 4267 7450  
 3184 4270 5274  
 3185 4271 7620  
 3186 4272 5301  
 3187 4273 5332  
 3188 4274 2141  
 3189 4275 7410  
 3190 4276 5332  
 3191 4277 2140  
 3192 4100 5262  
 3193 4101 2126  
 3194  
 3195 4102 1003  
 3196 4103 3012  
 3197 4104 1005  
 3198 4105 3011  
 3199 4106 3411  
 3200 4107 3411  
 3201 4110 3411  
 3202 4111 1005  
 3203 4112 3011  
 3204 4113 1137  
 3205 4114 7041  
 3206 4115 3150  
 3207 4116 6211  
 3208 4117 1412  
 3209 4120 6201  
 3210 4121 3411  
 3211 4122 2150  
 3212 4123 5316  
 3213 4124 1137  
 3214 4125 3150  
 3215 4126 1002  
 3216 4127 3142  
 3217 4130 1003  
 3218 4131 3143  
 3219 4132 1003  
 3220 4133 1137  
 3221 4134 1014  
 3222 4135 3127  
 3223 4136 5225

PRS3.

PRS4.

PRS5.

PRS6.

TAD PFLG  
 SZA CLA  
 JMP PRS3  
 TAD BCODE  
 AND K3777  
 DCA BCODE  
 TAD BCODE  
 SPA CLA  
 JMP PRS6  
 TAD I X1  
 CIA CLL  
 CDF 10  
 TAD I X2  
 CDF 00  
 SNA  
 JMP ,+4  
 SNL CLA  
 JMP PRS5  
 JMP PRS6  
 ISZ AACTR  
 SKP  
 JMP PRS6  
 ISZ BCTR  
 JMP PRS4  
 ISZ FOUND  
 TAD BPTR  
 DCA X2  
 TAD APTR  
 DCA X1  
 DCA I X1  
 DCA I X1  
 DCA I X1  
 TAD APTR  
 DCA X1  
 TAD BLEN  
 CIA  
 DCA ALEN  
 CDF 10  
 TAD I X2  
 CDF 00  
 DCA I X1  
 ISZ ALEN  
 JMP ,+5  
 TAD BLEN  
 DCA ALEN  
 TAD BVAL  
 DCA AVAL  
 TAD BPTR  
 DCA ASAV  
 TAD BPTR  
 TAD BLEN  
 TAD K2  
 DCA SPTR  
 JMP PRS2

/IS THIS THE FIRST TIME THRU THE TABLE?  
 /NO  
 /YES, CLR ENTRY BIT  
 / (THE HAS-BEEN-PRINTED FLAG)  
 /HAS THIS SYMBOL BEEN PRINTED ALREADY?  
 /YES, IGNORE IT  
 /NO, COMRARE A=SYM WORD  
 /WITH B=SYM WORD  
 /MATCH SO FAR  
 /A=SYM WORD IS BIGGER-- USE B=SYM  
 /VICE=VERSA  
 /IS A=SYM DONE?  
 /NO  
 /YES, STICK WITH A=SYM  
 /IS B=SYM DONE  
 /NO, TRY NEXT WORD  
 /YES, B=SYM IS NEW A=SYM  
 /SET CONTINUE FLAG  
 /INDEX B=SYM  
 /CLR STORAGE FOR NEW A=SYM  
 /RESET A=SYM INDEX  
 /CTR FOR TRANSFER  
 /MOVE B=SYM TO A=SYM  
 /NEW LENGTH  
 /NEW VALUE  
 /NEW PTR  
 /MOVE SPTR TO NEXT MST ENTRY  
 /CONTINUE SEARCH

/SABR,815	PAL10	V135	23-SEP-69	11128	PAGE 51-2
3224	4137	1126	PRS7, TAD	FOUND	/HAS ANOTHER SYMBOL BEEN FOUND?
3225	4142	7652	SNA CLA		
3226	4141	5371	JMP	PRS8	/NO, EXIT
3227	4142	4432	JMS I	OBSYM	/YES
3228	4143	2143	ASAV		
3229	4144	1221	TAD	USE	
3232	4145	1241	TAD	K4202	
3231	4146	3001	DCA	USE	/SET HAS=BEEN=PRINTED BIT
3232	4147	2125	ISZ	PFLG	/SET PASS FLAG
3233	4150	4424	JMS I	CRLF	/POSITION PRINT
3234	4151	1005	TAD	APTR	/INDEX SYMBOL
3235	4152	3011	DCA	X1	
3236	4153	1172	TAD	M3	/SET CTR
3237	4154	3150	DCA	ALEN	
3238	4155	1411	TAD I	X1	/PRINT SYMBOL
3239	4156	4423	JMS I	CTYPE	
3240	4157	2150	ISZ	ALEN	
3241	4160	5355	JMP	,=3	
3242	4161	4423	JMS I	CTYPE	/PRINT 2 SPACES
3243	4162	1142	TAD	AVAL	/PRINT VALUE
3244	4163	4435	JMS I	OTYPE	
3245	4164	1001	TAD	USE	/MOVE TYPE BITS TO LOW AC
3246	4165	7006	RTL		/8 DEF, BIT TO LINK
3247	4166	7006	RTL		
3248	4167	4774	JMS I	STFTI	/TYPE FLAGS IF ANY
3249	4170	5210	JMP	PRS1	/LOOK FOR ANOTHER SYMBOL TO PRINT
3250					
3251	4171	4424	PRS8, JMS I	CRLF	
3252	4172	4424	JMS I	CRLF	
3253	4173	5600	JMP I	PRSYM	
3254					
3255					
3256	4174	6121	STFTI, STFT		
3257		0143	ASAV=UMIC		
3258		0125	PFLG=TEM3		
3259	4175	2430	PST, STTP		/TOP OF PERMANENT SYMBOL TABLE
3260		0150	ALEN=S0		
3261		0005	APTR=AS0		
3262		0003	BPTR=SYMBOL		
3263		0002	BVAL=VAL		
3264		0001	BCODE=USE		
3265		0126	FOUND=TEM4		
3266		0127	SPTR=TEM5		
3267		0137	BLEN=LFS		
3268		0140	BCTR=OP		
3269		0141	AACTR=IB		
3270		0142	AVAL=AFS		
3271	4176	3777	K3777, 3777		
3272	4177	6531	LST0EP, LSTDEV		
3273					
3274					

```

3275          4200  *4200
3276
3277          /
3278          /ROUTINE TO PUSH DOWN CUR.LINE FOR NEXT PAGE.
3279          /SAVES ENTIRE LIST OF VITAL INFO
3280          /(LFS, OP, IB,....,BANK, S0) IN TEMP.LOCS
3281          /(TLFS, TOP, TIB,...., TS0)
3282          /BOTH LISTS MUST BE KEPT IN SPECIFIED
3283          /ORDER.
3284          /IF THERE IS AN LFS ON LINE MUST MARK IT
3285          /NO-LONGER-DEFINED-ON-PAGE IN PST.
3286          /
3287
3288          4200  0000  PUSHIN, 0
3289          4201  1137          TAD     LFS          /IS THERE AN LFS
3290          4202  7650          SNA  CLA
3291          4203  5213          JMP     PSHIN2        /NO
3292          4204  4432          JMS  I  OBSYM
3293          4205  1137          LFS
3294          4206  4450          JMS  I  SPSTAB        /GET ITS POINTERS TO THE PAGE SYMBOL TABLE
3295          4207  7000          NOP
3296          4210  1514          TAD  I  PSTCPR  /KILL THE DEFINED BIT
3297          4211  0256          AND   K3777A
3298          4212  3514          DCA  I  PSTCPR        /SET PAGE SYMBOL TABLE CODE WORD OFF PAGE
3299          4213  4237          PSHIN2, JMS  PUSHER
3300          4214  0136          LFS=1
3301          4215  4224          TLFS=1
3302          4216  5600          JMP  I  PUSHIN        /RETURN
3303          /
3304          /
3305          /
3306          4217  0000  POPIN, 0
3307          4220  7200          CLA
3308          4221  4237          JMS   PUSHER
3309          4222  4224          TLFS=1
3310          4223  0136          LFS=1
3311          4224  5617          JMP  I  POPIN        /RETURN
3312          4225  0000  TLFS, 0  /KEEP THIS LIST ORDERED AS GIVEN
3313          4226  0000  TOP, 0
3314          4227  0000  TIB, 0
3315          4230  0000  TAFS, 0
3316          4231  0000  TUMIC, 0
3317          4232  0000  TNSGN, 0
3318          4233  0000  TEXP, 0
3319          4234  0000  TSK, 0
3320          4235  0000  TBANK, 0
3321          4236  0000  TS0, 0
3322
3323          /
3324          /
3325          /TRANSFER ANY LIST OF 12 (12 OCTAL) ITEMS
3326          /FROM ONE LIST TO ANOTHER
3327          /CALL SEQ,1  JMS   PUSHER
3328          /          ADDR=1 OF FROM-LIST
3329          /          ADDR=1 OF TO-LIST

```

3330

/ RET.

3331

/

3332

3333

4237 2000

PUSHER, 0

3334

4240 1255

TAD M12A

3335

4241 3123

DCA TEM1

/CTR

3336

4242 1637

TAD I PUSHER

/INDEX FROM LIST

3337

4243 3011

DCA X1

3338

4244 2237

ISE PUSHER

3339

4245 1637

TAD I PUSHER

/INDEX TO LIST

3340

4246 3012

DCA X2

3341

4247 1411

TAD I X1

3342

4250 3412

DCA I X2

3343

4251 2123

ISE TEM1

3344

4252 5247

JMP ,=3

3345

4253 2237

ISE PUSHER

3346

4254 5637

JMP I PUSHER

3347

4255 7766

M12A, =12

3348

4256 3777

K3777A, 3777

```

3349 /RETRN PSEUDO-OP
3350 /
3351 4257 4427 PRT , JMS I GETSYM /GET NEXT INPUT ITEM
3352 4260 7410 SKP /NOTHING
3353 4261 5264 JMP ,+3 /SYMBOL
3354 4262 7200 NOP /CONSTANT
3355 4263 5575 IERROR /LITERAL
3356 4264 4446 JMS I SKIPL
3357 4265 4451 JMS I SREST /PLACE SYMBOL ON EXTERNAL SYMBOL TABLE
3358 4266 3313 DCA PRTN /SAVE SYMBOL ID
3359 4267 4200 JMS PUSHIN /PUSH LFS INFO IN CASE OF PAGE ASSEMBLY
3360 4270 1014 TAD K2 /SET AC TO 2
3361 4271 4714 JMS I PRTN1 /ARE THERE 2 LOCATIONS ON THIS PAGE
3362 4272 7610 SKP CLA /YES
3363 4273 4430 JMS I INI /NO ... HAD TO ASSEMBLE PAGE ... INITIALIZE PT
3364 4274 4217 JMS POPIN /POP LFS INFO FROM PAGE PUSH LIST
3365 4275 4421 JMS I ICPLFS /PROCESS ANY LFS
3366 4276 1517 TAD I PTCPR /PT CODE WORD WITH POSSIBLE LFS BIT
3367 4277 1020 TAD K30 /ADD SPECIAL RELOCATION BIT
3368 4300 3517 DCA I PTCPR /PLACE PROPER CODE WORD ON PT
3369 4311 1312 TAD DOTRTN
3370 4322 3521 DCA I PTSPR /PLACE JMS ,RTN INSTRUCTION IN PT SYMBOL WORD
3371 4323 4715 JMS I PRTN3 /INCREMENT PT POINTERS
3372 4324 1017 TAD K130 /PROPER BIT PATTERN
3373 4325 3517 DCA I PTCPR /TO PT CODE WORD
3374 4326 1313 TAD PRTN0 /PLACE EXTERNAL SYMBOL NUMBER ON PT
3375 4327 3521 DCA I PTSPR /AS SYMBOL WORD
3376 4328 4715 JMS I PRTN3 /INCREMENT PT POINTERS
3377 4311 5440 JMP I POPEXP /EXIT FOR NEXT LINE
3378 4312 4040 DOTRTN, JMS RTN
3379 4313 0000 PRTN0, 0
3380 4314 2646 PRTN1, IFFSUB
3381 4315 1552 PRTN3, ASMIF1
3382 /
3383 /
3384 *PAUSE* PSEUDO OPERATION
3385 /
3386 4316 4726 PPAUSE, JMS I WLNIFI /LIST IF PASS 2
3387 4317 7602 CLA HLT /WAIT FOR OPERATOR ACTION
3388 4320 6014 RFC /SELECT READER
3389 4321 4772 JMS I INITRP
3390 4322 4446 JMS I SKIPL
3391 4323 7240 CMA /WE REACH THIS ONLY IF FORFLG. LE. 0
3392 4324 3106 DCA FORFLG /SHUT OFF FORTR IN CASE GUY
3393 /HAS STARTED HIS TAPE IN
3394 /THE MIDDLE
3395 4325 5433 JMP I DCIL1 /RETURN FOR NEXT LINE
3396
3397 4326 1772 WLNIFI, WLNIF1
3398

```

```

3399
3400
3421
3422
3423      4327  0000      /
3424      4330  6221      /OVERAL ASSEMBLY INITIALIZATION
3425      4331  3773      /
3426      4332  4424      /
3427      4333  4424      /
3428      4334  1376      /
3429      4335  3774      /
3410      4336  4237      /
3411      4337  4357      /
3412      4340  0075      /
3413      4341  4772      /
3414      4342  1110      /
3415      4343  7640      /
3416      4344  5727      /
3417      4345  3003      /
3418      4346  4777      /
3419      4347  3062      /
3420      4350  1156      /
3421      4351  4237      /
3422      4352  4355      /
3423      4353  0005      /
3424      4354  4775      /
3425      4355  5727      /
3426      4356  7577      /
3427      4357  2430      /
3428
3429      4360  0177      /
3430      4361  0200      /
3431      4362  0000      /
3432      4363  0000      /
3433      4364  0000      /
3434      4365  0001      /
3435      4366  0001      /
3436      4367  0000      /
3437      4370  0000      /
3438      4371  0000      /
3439
3440      4372  4574      /
3441      4373  2700      /
3442      4374  5522      /
3443      4375  4714      /
3444      4376  0700      /
3445      4377  2371      /

INITA, 0
COF 00
DCA I  FATALP
JMS I  CRLF
JMS I  CRLF
TAD   PEB
DCA I  PEPTRP
JMS   PUSHER /INIT HICOM, PAGE, ESTSIZ, EQVBIT & APMSW ETC
      K777=1
      HICOM=1
JMS I  INITRP
TAD   PASS
SEA CLA
JMP I  INITA
DCA   SYMBOL      /PROTECT FROM RUSVL
JMS I  INITMP
DCA   CSUM
TAD   K10
JMS   PUSHER /SET PUSH CTR==2
      OTPR=1
      OTP=1
JMS I  LEADI
JMP I  INITA
OTPR,  CORE1=1      /KEEP STTR IMMEDIATELY AFTER OTPR
STTR,  STTP
/***** KEEP ITEMS SO INCLOSED IN GIVEN ORDER
K777,  177
      200
      0
      0
      1
      1
      0
      0
      0
/*****
INITRP, INITR
FATALP, FATAL
PEPTRP, PEPTR
LEADI,  LEADER
PEB,    PEBSE
INITMP, INITMR

```

```

3446                    4400      *4427
3447                    /
3448                    /
3449                    /      ROUTINE TO SEARCH SYMBOL TABLE FOR SYMBOL IN S0-S3
3450                    /      PLACES SYMBOL ON TABLE IF NOT THERE
3451                    /      CALL IS
3452                    /      JMS SRSYM
3453                    /      NOT FOUND EXIT
3454                    /      FOUND EXIT
3455                    /
3456                    /RETURNS WITH SYMBOL CODE BITS IN "USE"
3457                    /SYMBOL VALUE (0 IF NOT DEFINED)
3458                    /IN "VAL"
3459                    /& PTR TO SYM,TAB, ENTRY IN "SYMBOL"
3460                    /THE LATTER ADDRESS IS REFERRED TO HERE IN
3461                    /AS THE SYMBOL "ID"
3462                    /SRSYM CALL RUSVL TO STORE USE & VAL
3463                    /OF LAST REFERENCED SYMBOL IN MST
3464                    /IN CASE THEY HAVE BEEN CHANGED
3465                    /IN THE MEANWHILE,
3466                    /
3467                    4420      0220      SRSYM, 0
3468                    4421      4335                    JMS      RUSVL
3469                    4422      1043                    TAD      MST
3470                    4423      3003      SRS1,      DCA      SYMBOL      /START AT SYM, TAB, BASE
3471                    4424      1007                    TAD      STT      /SET PTR, TO NEXT ENTRY
3472                    4425      7041                    CIA                    /COMPARE PTR, WITH SYM, TAB, TOP
3473                    4426      1003                    TAD      SYMBOL
3474                    4427      7650                    SNA CLA                    /CONTINUE SEARCH
3475                    4410      5250                    JMP      SRS2      /NAME NOT IN TABLE ENTER IT
3476                    4411      6211                    CDF 10
3477                    4412      1403                    TAD I      SYMBOL      /GET ENTRY CODE WORD
3478                    4413      6201                    CDF 00
3479                    4414      0016                    AND      K3      /EXTRACT SYMBOL LENGTH
3480                    4415      3124                    DCA      TEM2
3481                    4416      1124                    TAD      TEM2
3482                    4417      7041                    CIA                    /NEGATE FOR COMPARE & CTR,
3483                    4420      3123                    DCA      TEM1
3484                    4421      1123                    TAD      TEM1      /COMPARE ENTRY & LOOK-UP SYMBOL LENGTHS
3485                    4422      1150                    TAD      S2
3486                    4423      7640                    SZA CLA                    /SAME LENGTH; COMPARE LETTERS
3487                    4424      5244                    JMP      SRS5      /NOT SAME; GO TO NEXT ENTRY
3488                    4425      1005                    TAD      AS0      /AUTO-INDEX LOOP-UP SYMBOL
3489                    4426      3011                    DCA      X1
3490                    4427      1003                    TAD      SYMBOL      /AUTO-INDEX TABLE ENTRY
3491                    4430      3012                    DCA      X2
3492                    4431      6211      SRS3,      CDF 10
3493                    4432      1412                    TAD I      X2      /GET TABLE ENTRY CHAR, PAIR
3494                    4433      7041                    CIA
3495                    4434      6221                    CDF 00
3496                    4435      1411                    TAD I      X1      /COMPARE LOOK-UP SYMBOL CHAR, PAIR
3497                    4436      7640                    SZA CLA      /SAME
3498                    4437      5244                    JMP      SRS5      /NO MATCH
3499                    4440      2123                    ISZ      TEM1      /CHK SYM, LEN, CTR,
3500                    4441      5231                    JMP      SRS3      /NOT DONE, TRY NEXT CHAR, PAIR

```

```

/SABR,815      PAL10  V135  23-SEP-69  11128  PAGE 55-1
3521  4442  4320  SRS4,  JMS  SUSVL  /GET USE & VAL WORDS
3522  4443  5600          JMP I  SRSYM
3523
3524  4444  1003  SRS5,  TAD  SYMBOL  /PTR TO LAST ENTRY
3525  4445  1014          TAD  K2      /*2 FOR USE & VAL WORDS
3526  4446  1124          TAD  TEM2     /*ENTRY SYMBOL LENGTH
3527  4447  5203          JMP  SRS1    /*PTR TO NEXT ENTRY
3528
3529          /CURRENT SYMBOL NOT ON TABLE ... PLACE IT THERE
3510
3511
3512  4450  1006  SRS2,  TAD  OTP      /WILL NEW ENTRY FIT BELOW
3513  4451  7141          CIA CLL  /OCCURANCE TABLE?
3514  4452  1003          TAD  SYMBOL
3515  4453  1150          TAD  S0
3516  4454  7630          S2L CLA  /0 LINK=YES
3517  4455  5577          SERROR  /NO, SYMBOL TABLE OVERFLOW
3518  4456  1150          TAD  S0      /ENTRY CODE WORD = SYM, LEN.
3519  4457  1030          TAD  K1000  /*REL BIT
3520  4460  6211          CDF 10
3521  4461  3407          DCA I  STT  /PUT CODE IN 1ST WORD OF NEW ENTRY
3522  4462  1007          TAD  STT  /AUTO=INDEX ENTRY
3523  4463  3012          DCA  X2
3524  4464  1005          TAD  AS0  /AUTO=INDEX SYMBOL TO BE STORED
3525  4465  3011          DCA  X1
3526  4466  1150          TAD  S0      /SET SYM, LEN, CTR.
3527  4467  7041          CIA
3528  4470  3123          DCA  TEM1
3529  4471  6201  ERS1,  CDF 00
3530  4472  1411          TAD I  X1  /MOVE SYMBOL CHAR, PAIR TO TABLE
3531  4473  6211          CDF 10
3532  4474  3412          DCA I  X2
3533  4475  2123          ISZ  TEM1  /CK, CTR.
3534  4476  5271          JMP  ERS1  /NOT DONE
3535  4477  3412          DCA I  X2  /CLR VALUE WORD
3536  4520  1007          TAD  STT  /SAVE PTR TO NEW ENTRY
3537  4521  3003          DCA  SYMBOL
3538  4522  1012          TAD  X2  /RESET PTR. TO SYM, TAB, TOP
3539  4503  7001          IAC
3540  4504  3007          DCA  STT
3541  4525  6201          CDF 00
3542  4526  5242          JMP  SRS4
3543

```



```

3544 /
3545 /OBTAIN GIVEN SYMBOL'S VITAL INFO FROM MST
3546 /CALL SECT JMS OBNSYM
3547 / ADDRESS OF SYMBOL ID
3548 / RET.
3549 /OBNSYM LEAVES SYMBOL ID IN "SYMBOL",
3550 / SYMBOL CODE WORD IN USE,
3551 / SYMBOL VALUE IN VAL.
3552 /OBNSYM CALLS RUSVL BEFORE ACTION
3553 /FOR SAME REASON AS SRSYM DOES.
3554 /
3555 OBNSYM, 0
3556 JMS RUSVL
3557 TAD I OBNSYM /ADDRESS OF SYMBOL
3558 DCA TEM1
3559 TAD I TEM1 /ACTUAL SYMBOL
3560 DCA SYMBOL
3561 ISZ OBNSYM /INDEX FOR EXIT
3562 JMS SUSVL /SET UP USE AND VALUE WORDS
3563 JMP I OBNSYM /RETURN WHEN FOUND
3564 /
3565 / ROUTINE TO SET UP USE AND VALUE WORDS
3566 /
3567 SUSVL, 0
3568 CDF 10 /OFF TO BANK 1
3569 TAD I SYMBOL /MST USE WORD FROM BANK 1
3570 DCA USE /TO BANK 0 USE LOCATION
3571 TAD USE
3572 AND K3
3573 IAC
3574 TAD SYMBOL
3575 DCA VALPTR
3576 TAD I VALPTR /MST VALUE WORD FROM BANK 1
3577 DCA VAL /TO BANK 0 VALUE LOCATION
3578 CDF 00 /RESTORE DATA FIELD
3579 JMP I SUSVL /RETURN
3580 /
3581 RUSVL, 0
3582 CDF 10
3583 TAD USE
3584 DCA I SYMBOL
3585 TAD VAL
3586 DCA I VALPTR
3587 CDF 00
3588 JMP I RUSVL
3589 /
3590 VALPTR, 0 /PTR TO CURRENT VAL WORD IN MST

```

```

3591 /
3592 /READ A CHARACTER
3593 / IGNORES LF, FF, RO, LEADER
3594 / ALSO CHECKS CHAR AS TO TYPE
3595 /CALLING SEQ: JMS RCH
3596 / RETURN IF CHAR IS A DIGIT
3597 / RETURN IF CHAR IS ALPHABETIC
3598 / RETURN FOR ALL OTHER (PUNCT,ETC)
3599 /LEAVES AC=0
3600 / CHR=ASCII VALUE OF INPUT CHARACTER
3601 /CALLS SRT
3602
3603 4546 0000 RCH, 0
3604 4547 4426 JMS I GETCHR /GET 1 CHAR
3605 4550 7450 SNA
3606 4551 5365 JMP RCH3 /0=END OF LINE
3607 4552 1371 TAD M260
3608 4553 7510 SPA
3609 4554 5365 JMP RCH3 /TAKE PUNCT,EXIT (200=257)
3610 4555 1372 TAD M12
3611 4556 7510 SPA
3612 4557 5367 JMP RCH4 /TAKE DIGIT EXIT (260=271)
3613 4560 1004 TAD M7
3614 4561 7510 SPA
3615 4562 5365 JMP RCH3 /TAKE PUNCT, EXIT (272=300)
3616 4563 1373 TAD M37A
3617
3618 4564 7500 SMA
3619
3620 4565 2346 RCH3, ISZ RCH /PUNCT, EXIT (337=376)
3621 4566 2346 ISZ RCH /ALPHA EXIT (301=336)
3622
3623 4567 7200 RCH4, CLA /DIGIT EXIT
3624 4570 5746 JMP I RCH
3625 4571 7520 M260, =260
3626 4572 7766 M12, =12
3627 4573 7742 M37A, =36
3628
3629
3630 /FORCE BUFFER FILL ON FIRST READ
3631
3632 4574 0000 INTR, 0
3633 4575 1170 TAD MBE
3634 4576 3013 DCA X3
3635 4577 5774 JMP I INTR
3636
3637 0170 MBE=LINAX /LAST WORD OF DATA BUFFER

```

3638      4600      \*4600

/SUBR TO READ 1 LINE INTO LINE BUFFER

```

3641
3642      4620      2200      RLN,      0
3643      4621      3130           DCA      LFLG      /CLR NON=NULL LINE FLAG
3644      4622      1107           TAD      SCOLON      /IF LAST LINE ENDED WITH ;
3645      4623      7440           SZA           /NO NEED TO READ ANOTHER
3646      4624      5234           JMP      RLN4
3647      4625      1170           TAD      LINAX      /INIT STORAGE AUTO=INDEX
3648      4626      3012           DCA      X2
3649      4627      4241      RLN2,      JMS      FETCH      /GET A CHARACTER
3650      4610      4453           JMS I      TEST      /IS IT A CR,TAB,SP,FF,LF?
3651      4611      1756                SL7=1      /IF SO GO TO RLN15,3,3,2,2
3652      4612      6766                BL6=SL7
3653      4613      2130           ISZ      LFLG      /OTHERWISE A NON=NULL LINE
3654      4614      4637      RLN3,      JMS I      STOREP      /OTHERWISE PUT IT IN THE BUFFER
3655      4615      1012           TAD      X2      /IS BUFFER FULL?
3656      4616      1240           TAD      LINEND
3657      4617      7640           SZA      CLA
3658      4620      5207           JMP      RLN2      /NO
3659      4621      7040           CMA
3660      4622      1012           TAD      X2
3661      4623      3012           DCA      X2      /IF SO MOVE BACK PTR
3662      4624      5207           JMP      RLN2
3663      4625      3061      RLN15,      DCA      CHR      /TERMINATE LINE WITH 0
3664      4626      4637           JMS I      STOREP
3665      4627      3133           DCA      AFLG
3666      4630      3131           DCA      EFLG
3667      4631      3132           DCA      VFLG
3668      4632      3134           DCA      CODE
3669      4633      1170           TAD      LINAX      /INIT LINE INDEX
3670      4634      3010      RLN4,      DCA      X0
3671      4635      3107           DCA      SCOLON      /CLR
3672      4636      5600           JMP I      RLN

```

3673  
3674  
3675      4637      5164      STOREP, STORE  
3676      4640      6023      LINEND, =LINBUF=107

/SUBROUTINE TO READ 1 CHARACTER VIA INPUT DEVICE  
/IGNORES 200'S & 377'S

```

3677
3678
3679
3680
3681      4641      2200      FETCH,      2
3682      4642      4256           JMS      R
3683      4643      1163           AND      K177
3684      4644      1164           TAD      K200      /FORCE FULL 8BIT ASCII
3685      4645      3061           DCA      CHR
3686      4646      1061           TAD      CHR
3687      4647      1166           TAD      4200
3688      4650      7440           SZA
3689      4651      1255           TAD      M177
3690      4652      7652           SNA      CLA
3691      4653      5242           JMP      FETCH+1
3692      4654      5641           JMP I      FETCH

```

```

3693      4655      7601      M177,      =177
3694
3695
3696
3697
3698      4656      2000      R,      0
3699      4657      6211           CDF 10
3700      4660      1013           TAD      X3
3701      4661      1304           TAD      BUFEND      /CK FOR END OF BUFFER
3702      4662      7650           SNA CLA
3703      4663      5267           JMP      RG      /REFILL
3704      4664      1413      R1,      TAD I      X3      /GET NEXT CHAR
3705      4665      6201           CDF 00
3706      4666      5656           JMP I      R
3707      4667      1303      RG,      TAD      BUF      /INDEX THE BUFFER
3708      4670      3013           DCA      X3
3709      4671      4702      RG1,      JMS I      INDEV
3710      4672      3413           DCA I      X3
3711      4673      1013           TAD      X3      /CK FOR FULL
3712      4674      1304           TAD      BUFEND
3713      4675      7640           SZA CLA
3714      4676      5271           JMP      RG1      /NOT FULL
3715      4677      1303      RG3,      TAD      BUF      /RESET PTR
3716      4700      3013           DCA      X3
3717      4701      5264           JMP      R1
3718
3719      4702      6610      INDEV,      HSR
3720      4703      1077      BUF,      DATA=1
3721
3722      4704      6133      BUFEND,      1=LINBUF
3723
3724      /GET 1 CHAR FROM LINE BUFFER
3725
3726      4705      2000      L65,      0
3727      4706      6211           CDF 10
3728      4707      1410           TAD I      X0
3729      4710      6201           CDF 00
3730      4711      3061           DCA      CHR
3731      4712      1061           TAD      CHR
3732      4713      5725           JMP I      L65
3733

```

3734  
3735  
3746  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750  
3751  
3752  
3753  
3754  
3755  
3756  
3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766  
3767  
3768  
3769

4714 7200  
4715 1166  
4716 3123  
4717 1164  
4722 4442  
4721 2123  
4722 5317  
4723 5714  
  
4724 2215  
4725 2212  
4726 0000  
4727 7200  
4730 1324  
4731 4454  
4732 1325  
4733 4454  
4734 5726  
  
4735 4446  
4736 7001  
4737 7410  
4740 4446  
4741 3105  
4742 5434

LEADER, 7  
TAD 47600  
DCA TEM1  
TAD 4220  
JMS I PUNCH  
ISR TEM1  
JMP .-3  
JMP I LEADER

ROUTINE TO TYPE RETURN=LINE FEED  
0215  
0212  
L73, 0  
CLA  
TAD L73-2  
JMS I TYPE  
TAD L73-1  
JMS I TYPE  
JMP I L73

/DECIM & OCTAL PSUEDO-OPS

PDEC, JMS I SKIPL  
IAC /SET ARITHMETIC CONVERSION TO DECIMAL  
SKP  
POCT, JMS I SKIPL  
DCA DSW /SET ARITHMETIC CONVERSION TO OCTAL  
JMP I NULLP /GO GET NEXT INPUT LINE

```

3770
3771
3772 /ROUTINE TO STOP NEXT LINE FROM BEING LISTED
3773 /THO IT IS ALREADY IN THE BUFR.
3774 /E.G., STOP LISTING OF PUSHED DOWN LINE
3775 /WHILE ASSEMBLING LIT. POOL
3776 /
3777 4743 0000 SAVLIN, 0
3778 4744 6211 CDF 10
3779 4745 1763 TAD I LINEB2 /SAVE 1ST CHAR OF LINE
3780 4746 7450 SNA /IF ANY
3781 4747 5352 JMP ,+3 /THERE ISNT ANY
3782
3783 4750 3362 DCA SAVEIT
3784 4751 3763 DCA I LINEB2 /CLR TO PREVENT TYPEOUT
3785 4752 6201 CDF 00
3786 4753 5743 JMP I SAVLIN
3787
3788
3789 /
3790 /REENABLE LISTING OF LINE WHICH SAVLIN
3791 /PREVENTED
3792 /
3793
3794 4754 0000 RELINE, 0
3795 4755 6211 CDF 10
3796 4756 1362 TAD SAVEIT /RESTORE 1ST CHAR OF LINE
3797 4757 3763 DCA I LINEB2
3798 4760 6201 CDF 00
3799 4761 5754 JMP I RELINE
3800
3801 4762 0000 SAVEIT, 0
3802 4763 1646 LINEB2, LINBUF
3803
3804
3805 /ROUTINE TO LIST NULL, COMMENT OR PSUEDO-OP LINE
3806
3807 4764 3136 NULL, DCA VALUE /IF ANY GIVEN
3808 4765 1136 TAD VALUE /SET TYPEOUT FLAG IF NON=0
3809 4766 3132 DCA VFLG
3810 4767 4771 JMS I WLIF /LIST IF PASS 2
3811 4770 5433 JMP I DCIL1 /GO BACK TO RDL1 FOR NEXT LINE
3812 4771 1772 WLIF, WLNIF1
3813
3814 /
3815 / TYPE ROUTINE
3816 /
3817 4772 0000 L64, 0
3818 4773 6046 TLS /SELECT IT
3819 4774 6041 TSF /WAIT FOR TTY
3820 4775 5374 JMP ,=1
3821 4776 7200 CLA /EXIT WITH CLEAR AC
3822 4777 5772 JMP I L64

```

```

3823      5000      *5000
3824      /
3825      /
3826      /          ROUTINE TO SET THE CORRECT COUNTERS FOR THE CURRENT
3827      /          OP CODE AND ADDRESS FIELD SYMBOL
3828      /
3829      /THIS IS A MAJOR ROUTINE. IT IS CALLED ONCE
3830      /FOR EVERY NORMAL (MRI,OPR,IOT) INSTR. COLLECTED.
3831      /IT IS ALSO CALLED DURING PAGE
3832      /RECOUNTING, ONCE FOR EVERY ITEM ON THE
3833      /PAGE TABLE.
3834      /CALLING SEQ: AC=0,JMS,RET WITH AC=0
3835      /FUNCTION: DETERMINE THE TYPE OF LINE BEING
3836      /READ AND SET THE VARIOUS PAGE COUNTERS
3837      /AND FLAGS ACCORDINGLY.
3838      /A FLOW CHART OF TYPES & FLAG SETTINGS IS GIVEN BELOW.
3839      /CONSIDERABLE OVERLAPPING IS USED TO ACHIEVE
3840      /THE MIN. CORE USAGE. THIS IS SOMETIMES AT THE
3841      /EXPENSE OF LOGICAL CLARITY.
3842      /ALL POSSIBLE CONDITIONS EXIT VIA SETC00
3843      /SETC00: (1) IF LAST INSTR. WAS A SKIP & LAST BANK
3844      /          IS NOT= CUR,BANK, BANK#=1.
3845      /          (2) IF CUR, INSTR. IS A SKIP, PGEESC=4
3846      /          OTHERWISE PGEESC=2.
3847      /          (3) LASTSKIP CONDITION= CUR, SKIP CONDITION
3848      /          (4) LAST BANK= CUR, BANK
3849      /
3850      /FLOW OF INSTR. TYPES
3851      /SETC1: IF (PARAMETER OR MICRO=INSTR.) SETC00
3852      /          IF (LITERAL AFS) SETC02
3853      /          IF (CONSTANT AFS) SETC01
3854      /          CALL OBNSYM(AFS)
3855      /          IF (INSTR. IS INDIRECT) SETC07
3856      /          IF (AFS IS IN COMMON) SETC04
3857      /          IF (ABSOLUTE AFS) SETC05
3858      /          CALL SPSTB (AFS)          /SEARCH PST FOR AFS
3859      /          CALL SETSUB
3860      /          IF (AFS NOT BEFORE ON PST) SETC06
3861      /          IF (AFS WAS IN PST BUT NOT DEF. ON PAGE) SETC12
3862      /          IF (OP CODE=JMS) BANK#1
3863      /          GO TO SETC02          /ON PAGE MR1
3864      /SETC01: IF (CONST,AFS ON PG.0) J2
3865      /          IF (INSTR. IS INDIRECT) ERROR
3866      /          CALL SLTAB(CONST, AFS)          /PUT CONST. IN LIT.TAB.
3867      /J1: IF (BANK NOT=1) SETC13
3868      /          GO TO SETC02
3869      /          J2= IF( INSTR,INDIR.) J1          /PG.0 INDIRECT
3870      /          GO TO SETC02          /PG.0 DIRECT
3871      /SETC04: IF (BANK NOT 2) CALL INCOBA          /INC OBACTR
3872      /          CALL NUMSGN
3873      /          S0=RESULT+COMMON ADDR.
3874      /SETC02: CALL SLTAB(S?)          /LIT.OR. COMMN. ADDR. TO LIT. TAB.
3875      /          GO TO SETC02
3876      /          IF (AFS NOT PREV. ON PST) SETC11
3877      /          IF (AFS WAS ON PST BUT NOT DEF. ON PAGE) SETC11
3878      /          GO TO J3

```

```

3878 /SETC10:CALL NUMSGN
3879 / IF (ABS,AFS ON PAGE 0) J3
3880 /SETC11:AC=1 /FORCE BANK#1
3881 /SETC09:AC=AC+1 /FORCE BANK#0
3882 /SETC08:AC=AC+2 /FORCE BANK#-1
3883 / CALL INCOBA /INC OBACTR
3884 / AC=BANK /((BANK OFFSET BY -1)
3885 / GO TO SETC13 /((TAKEN CARE OF AT SETC13)
3886 /SETC12:IF (NEW PST CODE BITS 10-11=OLD SAME (IN TEM 3)) J3
3887 /SETC06:INC OPSCTR /OFF PAGE SYMBOL
3888 /J3: IF (BANK#1) SETC08
3889 / INC AC
3890 / CALL INCOBA
3891 /J4: IF (THERE HAS NOT BEEN A PST SEARCH) SETC08
3892 / ADD CHANGE IN OBACTR (OBACTR=OLDOBA) TO PST CODE BITS 3-9
3893 / GO TO SETC08
3894 /
3895 /NOTE: CONDITION AT J4 IS TESTED BY SETSUB HEADER
3896 /WORD (OBFLG), THIS IS ALWAYS CLEARED
3897 /WHEN SETCT STARTS AND WILL NOT CHANGE
3898 /UNLESS THERE IS A CALL TO SPSTB BECAUSE
3899 /A CALL TO SETSUB ALWAYS FOLLOWS CALL TO
3900 /SPSTB IN SETCT.
3901
3902
3903 /
3904 5000 0000 SETCT, 0
3905 5001 4421 JMS I ICPLFS /CHECK FOR AND PROCESS ANY LFS
3906 5002 3334 DCA OBFLG /CLR
3907 /NEXT 2 LINES MOVED TO
3908 /SETCAL (AS OF V15) TO MAKE ROOM FOR FOLLOWING INSTR, & PTR
3909 / TAD EXP /IS IT PAR OR A MICRO INST?
3910 / TAD UMIC
3911 5003 4733 JMS I SETCAP
3912 5004 7640 SZA CLA
3913 5005 5715 JMP I SET001 /YES
3914 5006 7240 CLA CMA
3915 5007 1142 TAD AFS /IS AFS A LITERAL
3916 5010 7450 SNA
3917 5011 5716 JMP I SET021 /YES
3918 5012 1332 TAD MM1 /IS AFS A CONSTANT
3919 5013 7650 SNA CLA
3920 5014 5274 JMP SETC01 /YES
3921 5015 4432 JMS I OBSYM /NO ... GET POINTERS TO AFS
3922 5016 1142 AFS
3923 5017 1141 TAD IB /INDIRECT BIT
3924 5020 7640 SZA CLA /IS IT SET
3925 5021 5250 JMP SETC07 /YES
3926 5022 1001 TAD USE /NO ... AFS MST USE WORD
3927 5023 2160 AND K42
3928 5024 7640 SZA CLA /IS AFS OFF BANK (COMMON)
3929 5025 5717 JMP I SET041 /YES
3930 5026 1001 TAD USE /AFS MST USE WORD
3931 5027 3031 AND K3022
3932 5030 7650 SNA CLA /IS AFS ABSOLUTE

```



3933	5031	5720	JMP I	SET05I	/YES
3934	5032	4450	JMS I	SPSTAB	/IS AFS ON PST
3935	5033	7040	CMA		/NOT FOUND
3936	5034	4334	JMS	SETSUB	
3937	5035	5723	JMP I	SET06I	/NO ... MUST BE OFF PAGE
3938	5036	1514	TAD I	PSTCPR	/YES ... PST CODE WORD
3939	5037	7700	SMA CLA		/IS AFS ON PAGE
3940	5040	5730	JMP I	SET12I	/NO
3941	5041	1140	TAD	OP	/YES ... IS OP CODE A JMS
3942	5042	1041	TAD	K4000	
3943	5043	7640	SZA CLA		
3944	5044	5715	JMP I	SET00I	/NO ... JUST CONTINUE
3945	5045	7201	CLA IAC		/YES ... SET BANK TO CURRENT
3946	5046	3147	DCA	BANK	
3947	5047	5715	JMP I	SET00I	
3948			/		
3949			/		
3950				INDIRECT MEMORY REFERANCE INSTRUCTION	
3951	5050	1001	SETC07, TAD	USE	/AFS MST USE WORD
3952	5051	0157	AND	K20	/IS AFS DUMMY
3953	5052	7640	SZA CLA		
3954	5053	5724	JMP I	SET08I	/YES
3955	5054	1001	TAD	USE	/AFS MST USE WORD
3956	5055	0160	AND	K40	
3957	5056	7640	SZA CLA		/IS AFS OFF BANK (COMMON)
3958	5057	5725	JMP I	SET09I	/YES
3959	5060	1001	TAD	USE	/AFS MST USE WORD
3960	5061	0031	AND	K3000	
3961	5062	7650	SNA CLA		/IS AFS ABSOLUTE
3962	5063	5726	JMP I	SET10I	/YES
3963	5064	4450	JMS I	SPSTAB	/IS AFS ON PST
3964	5065	7040	CMA		/NOT FOUND
3965	5066	4334	JMS	SETSUB	
3966	5067	5727	JMP I	SET11I	/NO
3967	5070	1514	TAD I	PSTCPR	/YES ... PST CODE WORD
3968	5071	7700	SMA CLA		/IS AFS ON PAGE
3969	5072	5727	JMP I	SET11I	/NO
3970	5073	5731	JMP I	SET6P1	/YES
3971			/		
3972			/		
3973				CONSTANT FOR AN ADDRESS FIELD SYMBOL	
3974	5074	1150	SETC01, TAD	S0	/ACTUAL BINARY CONSTANT
3975	5075	0166	AND	K7600	/IS CONSTANT ON PAGE ZERO
3976	5076	7650	SNA CLA		
3977	5077	5311	JMP	SET01A	/YES
3978	5100	1141	TAD	IB	/NO ... IS IT INDIRECT
3979	5101	7640	SZA CLA		
3980	5102	5575	IERROR		/YES ... ERROR
3981	5103	4447	JMS I	SLITAB	/IS CONSTANT ON LITERAL TABLE
3982					/IF NOT SUBROUTINE PUTS IT THERE
3983	5104	7240	SET01B, CLA	CMA	
3984	5105	1147	TAD	BANK	/BANK SETTING
3985	5106	7650	SNA CLA		/IS IT SET TO THE CURRENT BANK
3986	5107	5715	JMP I	SET00I	/YES ... NO PROBLEMS
3987	5110	5721	JMP I	SET00J	/EXIT TO COMMON AREA

```

3988 /
3989 5111 1141 SET01A, TAD IB /IS INDIRECT BIT SET
3990 5112 7640 SZA CLA
3991 5113 5304 JMP SET01B /YES
3992 5114 5715 JMP I SET01I /NO
3993 5115 5251 SET001, SETC00
3994 5116 5306 SET002, SETC02
3995 5117 5300 SET004, SETC04
3996 5120 5200 SET005, SETC05
3997 5121 5236 SET00J, SETC13
3998 5122 5600 SET00B, JMP I SETCT /OFF PAGE RETURN
3999 5123 5231 SET061, SETC06
4000 5124 5217 SET081, SETC08
4001 5125 5221 SET091, SETC09
4002 5126 5212 SET101, SETC10
4003 5127 5220 SET111, SETC11
4004 5130 5224 SET121, SETC12
4005 5131 5232 SET6P1, SETC06+1
4006 5132 7777 MM1, =1
4007 5133 1165 SETCAP, SETCAL

```

```

4028 /
4029 /SETSUB IS A UTILITY USED BY SETCT ONLY,
4030 /USED ONLY IMMEDIATELY AFTER A PST SEARCH,
4031 /CALLING SEQUENCE: JMS SPSTB
4032 / CMA /SPSTB MAY SKIP
4033 / JMS SETSUB
4034 / RETURN IF SPSTB SKIPPED OVER CMA
4035 / RETURN IF SPSTB DID NOT SKIP
4036 /HAS SEVERAL FUNCTIONS:
4037 / (1) SAVE COPY OF OLD VALUE OF OBACKR BEFORE
4038 / CHANGING STARTS= SO IT MAY BE USED AT SETC13.
4039 / (2) IF INSTR IS INDIRECT, THAT'S IT= EXIT
4040 / (3) OTHERWISE SAVE OLD VALUE OF PST CODE BITS 10-11
4041 / FOR LATER USE AT SETC12. WARNING! THIS
4042 / IS SAVED IN TEM3, SO TEM3 IS NOT
4043 / TEMPORARY FOR A FEW MINUTES.
4044 / (4) SET PST CODE BIT 10 IF THIS IS A #REF,
4045 / OR BIT 11 IF IT IS A NORMAL REF.
4046 / ALGORITHM IS A.OR.B=(.NOT.A.AND.B)+A
4047 /
4048 SETSUB, 0
4049 DCA TEM1 /0=FOUND, =1=NOT
4050 TAD OBACKR /SAVE FOR SETC11,12,6,13
4051 DCA I OLDOBP
4052 TAD IB /OMIT CHANGING PST BITS IF INDIRECT
4053 SZA CLA
4054 JMP SETSX
4055 TAD I PSTCPR /SAVE OLD CODE
4056 AND K3
4057 DCA TEM3
4058 TAD NSGN
4059 SZA CLA
4060 IAC /#
4061 IAC
4062 DCA TEM2
4063 TAD TEM2 /OR INTO CODE
4064 CMA
4065 AND I PSTCPR
4066 TAD TEM2
4067 DCA I PSTCPR
4068 SETSX, ISZ TEM1 /FOUND?
4069 ISZ SETSUB /YES
4070 JMP I SETSUB
4071
4072 OLDOBP, OLDOBA
4073 OBFLG=SETSUB
4074
4075 /SUBR TO STORE CHARACTER IN LINE BUFFER
4076 /ASSUMES X1 SET
4077 /CHAR MAY BE IN AC OR IN CHR
4078
4079 STORE, 0

```

4063	5165	7450	SNA	
4064	5166	1061	TAD	CHR
4065	5167	6211	CDP	10
4066	5170	3412	DCA	I X2
4067	5171	6201	CDP	00
4068	5172	5764	JMP	I STORE

4069

4070

4071

4072

4073

4074

4075

4076

4077

4078

4079

4080

4081

4082

4083

4084

4085

4086

4087

5173 2000

5174 1144

5175 7640

5176 7001

5177 5773

/

/ROUTINE TO CHECK NSGN FOR SETCT

/USED ONLY BY SETCT ROUTINE.

/CALLING SEQUENCE: AC=0

/ JMS NUMSGN

/ RETURN WITH AC=0 IF

/ NSGN=0, AC=1 IF NSGN

/ NOT=0.

/NOTE: NSGN MAY BE NON=0 AND NOT=1. THIS

/IS THE REASON FOR NUMSGN.

/

NUMSGN, 0

TAD NSGN

SZA CLA

IAC

JMP I NUMSGN

```

/SABR,815      PAL10  V135  23-SEP-69  11128  PAGE 64
4088           5200  *5222
4089           /
4090           /      AFS ABSOLUTE
4091           /
4092           5200  4764  SETC05, JMS I  NUMSGP
4093           5221  1002  TAD      VAL      /ABSOLUTE SYMBOL VALUE
4094           5222  0166  AND      K7600  /MASK OUT PAGE BITS
4095           5223  7650  SNA CLA      /IS ABSOLUTE SYMBOL ON PAGE ZERO
4096           5224  5251  JMP      SETC00 /YES ... EXIT
4097           5225  4764  JMS I  NUMSGP
4098           5226  1002  TAD      VAL      /NO ... ABSOLUTE SYMBOL VALUE
4099           5227  3150  DCA      S0      /TO LITERAL TABLE SEARCH LOCATION
4100           5210  4447  JMS I  SLITAB  /SEARCH LITERAL TABLE FOR VALUE
4101           /
4102           5211  5232  JMP      SETC06+1 /IF NOT THERE ROUTINE PLACES IT THERE
4103           /
4104           /      INDIRECT ABSOLUTE
4105           /
4106           5212  4764  SETC10, JMS I  NUMSGP
4107           5213  1002  TAD      VAL      /ACTUAL AFS VALUE
4108           5214  0166  AND      K7600
4109           5215  7650  SVA CLA      /IS ADDRESS FIELD SYMBOL ON PAGE ZERO
4110           5216  5232  JMP      SETC26+1 /YES
4111           /
4112           /      INDIRECT DUMMY ADDRESS FIELD SYMBOL
4113           /
4114           5217  1171  SETC08, TAD      M2      /SET BANK UNKNOWN
4115           /
4116           /      OFF PAGE INDIRECT
4117           /
4118           5220  7001  SETC11, IAC      /SET BANK TO CURRENT
4119           /
4120           /      OFF BANK INDIRECT = SET BANK TO 0
4121           /
4122           SETC09,
4123           5221  4367  JMS      INCOBA  /SET BANK & INCR, OBACTR
4124           5222  1147  TAD      BANK
4125           5223  5237  JMP      SETC13+1 /EXIT TO COMMON AREA
4126           /
4127           /
4128           /
4129           /      ADDRESS FIELD SYMBOL NOT ON PAGE SYMBOL TABLE,
4130           /
4131           5224  1514  SETC12, TAD I  PSTCPR  /HAS NEW TYPE REF BEEN ADDED?
4132           5225  0016  AND      K3
4133           5226  7041  CIA
4134           5227  1125  TAD      TEM3
4135           5230  7640  SZA CLA      /YES
4136           5231  2075  SETC06, ISZ  OPSCTR  /INCREMENT OFF PAGE SYMBOL COUNTER
4137           5232  7240  CLA CMA
4138           5233  1147  TAD      BANK      /IS BANK SET TO CURRENT
4139           5234  7650  SNA CLA
4140           5235  5251  JMP      SETC00  /YES ... EXIT TO COMMON AREA
4141           5236  7001  SETC13, IAC
4142           5237  4367  JMS      INCOBA  /SET BANK TO CUR, & INC OBACTR

```

/SABR,815	PAL10	V135	23-SEP-69	1112R	PAGE 64-1
4143	5240	1677	TAD I	OBFLGP	/WAS THERE A PST SEARCH?
4144	5241	7650	SNA CLA		
4145	5242	5251	JMP	SETC00	/NO
4146	5243	1276	TAD	OLDOBA	/YES GET CHANGE IN OBACTR
4147	5244	7041	CIA		
4148	5245	1074	TAD	OBACTR	
4149	5246	7106	CLL RTL		/IN BITS 1-9
4150	5247	1514	TAD I	PSTCPR	/ADD TO PST CODE
4151	5250	3514	DCA I	PSTCPR	
4152			/		
4153			/	COMMON AREA	
4154			/		
4155	5251	1072	SETC00, TAD	LSTSKP	/IS LAST INSTRUCTION A SKIP INSTRUCTION
4156	5252	7650	SNA CLA		
4157	5253	5263	JMP	SET00A	/NO
4158	5254	1073	TAD	LSTBNK	/YES ... LAST BANK
4159	5255	7041	CIA		
4160	5256	1147	TAD	BANK	/CUR BANK
4161	5257	7650	SNA CLA		/ARE THEY THE SAME
4162	5260	5263	JMP	,+3	/YES
4163	5261	7240	CLA CMA		/NO ... SET BANK UNKNOWN
4164	5262	3147	DCA	BANK	
4165	5263	1146	SET00A, TAD	CURSKP	/IS CURRENT INSTRUCTION A SKIP INSTRUCTION
4166	5264	7640	SZA CLA		
4167	5265	1014	TAD	K2	/YES ... PAGE ESCAPE = 4
4168	5266	1014	TAD	K2	/NO ... PAGE ESCAPE = 2
4169	5267	3111	DCA	PGEESC	
4170	5270	1146	TAD	CURSKP	/RESET LAST SKIP INDICATOR
4171	5271	3072	DCA	LSTSKP	/FOR NEXT INPUT LINE
4172	5272	1147	TAD	BANK	/RESET LAST BANK INDICATOR
4173	5273	3073	DCA	LSTBNK	
4174	5274	5675	JMP I	,+1	/RETURN
4175	5275	5122	SET00B		
4176					
4177	5276	0000	OLDOBA, 0		
4178	5277	5134	OBFLGP, OBFLG		
4179			/		
4180			/	DIRECT OFF BANK REFERANCE	
4181			/		
4182	5300	1147	SETC04, TAD	BANK	/BANK INDICATOR
4183	5301	7640	SZA CLA		/IS BANK SET TO OFF
4184	5302	4367	JMS	INCOBA	/NO, SET BANK TO COMMN & INC OBACTR
4185	5303	4764	JMS I	NUMSGP	
4186	5304	1002	TAD	VAL	/YES ... ACTUAL BANK 0 ADDRESS
4187	5305	3150	DCA	S0	/TO CONSTANT = LITERAL LOCATION
4188			/		
4189			/	LITERAL FOR AN AFS	
4190			/		
4191	5306	4447	SETC02, JMS I	SLITAB	/PLACE LITERAL ON LITERAL TABLE
4192	5307	5251	JMP	SETC00	/EXIT TO COMMON AREA

```

4193 /
4194 /
4195 /      COLLECTION ROUTINE TO CHECK FOR AND PROCESS AN LFS
4196 /
4197 /CALLING SEQUENCE: AC#0
4198 /      JMS      CPLFS
4199 /      RETURN WITH AC#0
4200 /FUNCTION:      USED DURING COLLECTION PHASE
4201 /      EXCEPT WHEN RECOUNTING A PAGE,
4202 /      IF TAG OCCURS ON CURRENT LINE, CPLFS
4203 /      LOCATES (OR ENTERS) IT IN PAGE SYM.TAB.
4204 /      AND SETS THE DEFINED-ON-PAGE BIT IN
4205 /      THE PST CODE WORD,
4206 /      ALSO SETS BANK CONDITION TO UNKNOWN
4207 /      SINCE USER CODE CAN JUMP TO TAG
4208 /      FROM ANYWHERE, ALSO SAVE
4209 /      TAG IN LLFS & RESET LINE COUNT
4210 /      IN CASE WE GET A MULT.DEF. ERROR
4211 /      IN PASS1 WE MUST ALSO DO THE
4212 /      FOLLOWING IF THE SYMBOL IS ALREADY
4213 /      IN THE PST WHEN WE GO LOOK FOR
4214 /      IT: (1) REDUCE THE OFF-PAGE SYM,
4215 /      CTR, BECAUSE OFF-PAGE POINTER (FOR
4216 /      EITHER NORMAL OR # REFERENCES) ARE
4217 /      NO LONGER NEEDED, (2) REDUCE
4218 /      OBACTR BY THE NO. OF EXTRA WORDS
4219 /      OF CODE DUE TO THIS SYMBOL.
4220 /SUBRS: CALLED: OBNSYM(LFS),SPSTB,PSTDEF
4221 /
4222 /CPLFS, 0
4223 /      TAD I  RECTI      /ARE WE RECOUNTING?
4224 /      SZA CLA
4225 /      JMP I  CPLFS      /YES ... RETURN
4226 /      TAD      LFS
4227 /      SNA
4228 /      JMP I  CPLFS      /NONE THERE
4229 /      CDF 10
4230 /      DCA I  LFSPTR     /PLACE ON LFS TABLE
4231 /      ISZ   LFSPTR     /INCREMENT LFS TABLE POINTER
4232 /      CDF 00
4233 /      ISZ I  PTCPR     /SET LFS BIT ON PAGE TABLE
4234 /      JMS I  OBSYM
4235 /      LFS
4236 /      JMS I  SPSTAB     /IS IT ON THE PAGE SYMBOL TABLE
4237 /      JMP   CPLFS3     /NO ... SKIP DECREMENTING
4238 /      DCA   TEM1      /CLR
4239 /      TAD   PASS      /SKIP DECREMENTING IF PASS 2
4240 /      SNA CLA
4241 /      TAD I  PSTCPR    /CK USE
4242 /      AND   K3
4243 /      CLL RAR
4244 /      SZL
4245 /      ISZ   TEM1      /NORMAL
4246 /      SZA CLA
4247 /      ISZ   TEM1      /#
4248 /      TAD   TEM1      /SUBTRACT

```

4248	5343	7041	CIA		
4249	5344	1075	TAD	OPSCTR	
4250	5345	3075	DCA	OPSCTR	
4251	5346	1514	TAD I	PSTCPR	/EXTRACT SHARE OF OBACKR DUE
4252	5347	2363	AND	C3774	/TO THIS SYMBOL
4253	5350	7112	CLL RTR		/MOVE TO LOW ORDER
4254	5351	7041	CIA		/SUB. FROM OBACKR
4255	5352	1074	TAD	OBACKR	
4256	5353	3074	DCA	OBACKR	
4257	5354	4765	CPLFS3, JMS I	PSTDEP	
4258	5355	7340	CLA CMA	CLL	/SET BANK UNKNOWN (THE CLL IS USED ELSEWHERE)
4259	5356	3147	DCA	BANK	
4260	5357	1137	TAD	LFS	/SAVE IN CASE OF ERROR
4261	5360	3366	DCA	LLFS	
4262	5361	3067	DCA	LINE	/ZERO LINE COUNT FROM LAST LFS
4263	5362	5710	JMP I	CPLFS	
4264	5363	3774	C3774, 3774		
4265	5364	5173	NUMSGP, NUMSGN		
4266	5365	5570	PSTDEP, PSTDEF		
4267	5366	0000	LLFS, 0		
4268					
4269					
4270			/		
4271			/UTILITY FOR SETC04,SETC08,SETC13		
4272			/NOT USED ELSEWHERE		
4273			/CALLING SEQUENCE: DESIRED BANK SETTING IN AC		
4274			/ JMS INCOBA		
4275			/ RETURN WITH AC=0		
4276			/FUNCTION(1) SET BANK AS SPECIFIED		
4277			/ (2) INCREMENT OFF-BANK ADDITION CTR		
4278			/ BY 1 OR 2; 2 IF PREVIOUS INSTR.		
4279			/ WAS A SKIP-TYPE, 1 OTHERWISE.		
4280			/		
4281	5367	0000	INCOBA, 0		
4282	5370	3147	DCA	BANK	
4283	5371	1072	TAD	LSTSKP	/LAST INSTRUCTION SKIP INDICATOR
4284	5372	7640	SEA CLA		/WAS LAST INSTRUCTION A SKIP INSTRUCTION
4285	5373	2074	ISZ	OBACKR	/+ OLD VALUE OF OFF BANK ADDITION COUNTER
4286	5374	2074	ISZ	OBACKR	/FOR NEW VALUE OF OFF BANK ADDITION COUNTER
4287	5375	5767	JMP I	INCOBA	





```

4324
4325
4326 /ASSEMBLY ROUTINE TO FINISH OFF A PAGE
4327 /(1) PUTS OUT PAGE ESCAPE
4328 /(2) LITERAL POOL (BY CALLING OAPLT)
4329 /(3) GET READY FOR NEXT PAGE
4330 /
4331 5424 0000 A2, 0
4332 5425 4720 JMS I SAVLNP /STOP NEXT LINE LISTING
4333 5426 1102 TAD APMSW /ARE WE IN AUTOMATIC PAGING MODE
4334 5427 7640 SZA CLA
4335 5430 5311 JMP A2NONA /NO ... DONT SEND PAGE ESCAPE
4336 5431 1111 TAD PGEESC /SIZE OF PAGE ESCAPE REQUIRED
4337 5432 7012 RTR /2 BIT TO LINK
4338 5433 7630 SZA CLA
4339 5434 5200 JMP ASM01A /2 INSTRUCTION PAGE ESCAPE
4340 /4 INSTRUCTION PAGE ESCAPE
4341 5435 4456 JMS I WRITEP
4342 5436 4436 JMS I OUTBIN /JMP NEXT TO LAST LOC ON THIS PAGE
4343 5437 6311 K5376
4344 5440 0000 0
4345 5441 2005 ISZ ILC /FOR BENEFIT OF "WRITE"
4346 5442 4456 JMS I WRITEP
4347 5443 4436 JMS I OUTBIN /JMP LAST LOC ON THIS PAGE
4348 5444 6312 K5377
4349 5445 0000 0
4350 5446 4717 JMS I OUPALT /OUTPUT LITERAL TABLE
4351 5447 1323 TAD K176 /SET ILC IN CASE NO LITERALS
4352 5450 1077 TAD PAG
4353 5451 3065 DCA ILC
4354 5452 4436 JMS I OUTBIN /OUTPUT ORIGIN
4355 5453 0065 ILC
4356 5454 0004 4
4357 5455 4437 JMS I OUTSKP /OUTPUT 2 SKIP INSTRUCTIONS
4358 5456 4437 JMS I OUTSKP
4359 5457 5277 JMP ASM01B /RETURN
4360 /
4361 5460 4456 ASM01A, JMS I WRITEP
4362 5461 4436 JMS I OUTBIN /OUTPUT JMP TO LAST INSTRUCTION ON THIS PAGE
4363 5462 6312 K5377
4364 5463 0000 0
4365 5464 4717 JMS I OUPALT /LIT. TAB.
4366 5465 1163 TAD K177
4367 5466 1077 TAD PAG
4368 5467 3065 DCA ILC
4369 5470 4436 JMS I OUTBIN /OUTPUT IT AS AN ORIGIN
4370 5471 0065 ILC
4371 5472 0004 4
4372 5473 4456 JMS I WRITEP
4373 5474 4436 JMS I OUTBIN /PLACE A NOP IN THE LAST LOCATION
4374 5475 3647 K7000
4375 5476 0000 0
4376 5477 4721 ASM01B, JMS I RELNP /RESTORE NEXT LINE FOR LISTING
4377 5500 1057 TAD ACTR /REMAINS 1 DURING PASS 2
4378 5501 7640 SZA CLA

```

/SABR,815

FAL10

V135

23-SEP-69

11128 PAGE 67-1

```

4379 5502 5624      JMP I  A2      /EXIT IF LISTING OR 2ND ASSEMBLY
4380 5503 1112      TAD  PUPGE      /SAVE ESCAPE ON PUSH DOWN LIST
4381 5504 6211      CDF 10
4382 5505 3722      DCA I  PEPTR
4383 5506 6201      CDF 00
4384 5507 2322      ISZ  PEPTR      /MOVE STACK PTR
4385 5510 5624      JMP I  A2      /RETURN
4386
4387 5511 4717      A2NONA, JMS I  OUAPLT
4388 5512 3112      DCA  PUPGE      /CLR
4389 5513 5277      JMP  ASM01B
4390
4391 5514 5651      ASM02I, ASM02
4392 5515 0000      INDX1, 0
4393 5516 1041      INIS, INISUB
4394 5517 3200      OUAPLT, OAPLT
4395 5520 4743      SAVLNP, SAVLIN
4396 5521 4754      RELNP, RELINE
4397 5522 0700      PEPTR, PEBSE
4398 5523 0176      K176, 176
4399 5524 0000      REDUCE, 0
4400 5525 4045      CDFSKP, JMS  CDFSK
4401
4402
4403
4404
4405
4406 5526 0000      /
4407 5527 1077      /INITIALIZE A PAGE ASSEMBLY
4408 5530 3065      /
4409 5531 3744      A1, 0
4410 5532 3070      TAD  PAG      /MOVE PAGE TO ILC
4411 5533 1110      DCA  ILC
4412 5534 7650      DCA I  SME1X  /CLR
4413 5535 5726      DCA  LITSIZ  /ZERO LITERAL TABLE SIZE
4414 5536 6211      TAD  PASS
4415 5537 1722      SNA CLA
4416 5540 6201      JMP I  A1      /EXIT IF PASS 1
4417 5541 2322      CDF 10
4418 5542 3324      TAD I  PEPTR  /GET NEXT PAGE ESC FROM STACK
4419 5543 5726      CDF 00
4420 5544 5745      ISZ  PEPTR  /MOVE PTR
          DCA  REDUCE
          JMP I  A1
SME1X, ASME1X

```

```
4421 /
4422 /
4423 /
4424 5545 4427 PDUMMY; JMS I GETSYM /GET NEXT INPUT ITEM
4425 5546 7410 SKP /NOTHING THERE
4426 5547 5352 JMP ,+3 /SYMTOL
4427 5550 7000 NOP /CONSTANT
4428 5551 5575 IERROR /LITERAL
4429 5552 4446 JMS I SKIPL
4430 5553 1001 TAD USE /MST USE WORD
4431 5554 0360 AND K3403 /SAVE SYMBOL LENGTH, TYPE BITS, AND DEF BIT
4432 5555 1157 TAD K20 /ADD CORRECT MST BIT FOR DUMMY
4433 5556 3001 DCA USE /FOR CORRECT CODE WORD
4434 5557 5434 JMP I NULLP /EXIT TO GET NEXT LINE
4435 5560 3403 K3403, 3403
4436
4437
4438 /KLUDGE TO RESET ILC BECAUSE A1 COMES BEFORE UDPAGE IN PASS 2
4439
4440 5561 0000 FIXILC; 0
4441 5562 1110 TAD PASS
4442 5563 7650 SNA CLA
4443 5564 5761 JMP I FIXILC
4444 5565 1077 TAD PAC
4445 5566 3065 DCA ILC
4446 5567 5761 JMP I FIXILC
4447
4448
4449 /ROUTINE TO SET DEFINED BIT ON PST
4450 /USED BY CPLFS & ANUMCK
4451
4452 5570 0000 PSTDEF; 0
4453 5571 1514 TAD I PSTCPR /PROTECT CODES
4454 5572 0376 AND C3777
4455 5573 1041 TAD K4000 /SET DEFINED BIT ON PST
4456 5574 3514 DCA I PSTCPR
4457 5575 5770 JMP I PSTDEF
4458 5576 3777 C3777, 3777
```

4459	5620	*5620		
4460		/		
4461		/	CONSTANT FOR A' ADDRESS FIELD SYMBOL	
4462		/		
4463	5620	1521	ASM25, TAD I	PTSPR      /ACTUAL BINARY CONSTANT
4464	5621	1166	AND	<7622      /IS IT ON PAGE ZERO
4465	5622	7640	SZA	CLA
4466	5623	5615	JMP I	ASM5CI      /NO
4467	5624	1521	TAD I	PTSPR      /ADD IN PAGE ZERO ADDRESS
4468	5625	3145	DCA	TEMP6
4469	5626	1517	TAD I	PTCPR      /YES ... IS IT INDIRECT
4470	5627	2045	AND	<420
4471	5610	7650	SNA	CLA
4472	5611	5730	JMP I	ASM20I      /EXIT TO COMMON AREA
4473	5612	1145	TAD	TEMP6
4474	5613	5614	JMP I	ASM5A1
4475	5614	6300	ASM5A1, ASM12E	
4476	5615	6231	ASM5CI, ASM05C	
4477			/	
4478			/	OFF BANK DIRECT (COMMON DIRECT)
4479			/	
4480	5616	4647	ASM08, JMS I	NSCHKP
4481	5617	1002	TAD	VAL      /ACTUAL ADDRESS IN BANK ZERO
4482	5620	3151	DCA	S1      /TO 2 WORD LITERAL TABLE SEARCH LOCATION
4483	5621	7201	CLA	IAC
4484	5622	3150	DCA	S0      /ABSOLUTE SEARCH
4485	5623	4646	JMS I	SERALP      /GET A PAGE ADDRESS
4486	5624	1045	TAD	<400      /ADD INDIRECT BIT
4487	5625	3145	DCA	TEMP6
4488	5626	1147	TAD	BANK      /BANK INDICATOR
4489	5627	7650	SNA	CLA      /IS IT SET
4490	5630	5730	JMP I	ASM00I
4491	5631	3640	DCA I	ASM5I
4492	5632	1242	TAD	KCDF1A
4493	5633	3641	DCA I	ASM4I
4494	5634	1245	TAD	COZSKI
4495	5635	3643	DCA I	ASM6I
4496	5636	3103	DCA	TEM7
4497	5637	5644	JMP I	ASME7I
4498	5640	6071	ASM5I, ASMX5	
4499	5641	6070	ASM4I, ASMX4	
4500	5642	5660	KCDF1A, KCDF10	
4501	5643	6076	ASM6I, ASMX6	
4502	5644	6063	ASME7I, ASME7	
4503	5645	2577	COZSKI, COZSKP	
4504	5646	3443	SERALP, SRALT	
4505	5647	6333	NSCHKP, NSCHK	

4526  
4527  
4528  
4529  
4510  
4511  
4512  
4513  
4514  
4515  
4516  
4517  
4518  
4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560

5650 5651  
5651 0000  
5652 3137  
5653 1110  
5654 7640  
5655 4773  
5656 4431  
5657 4737  
5660 6211  
5661 1520  
5662 6201  
5663 3140  
5664 1517  
5665 0160  
5666 3146  
5667 1517  
5670 0156  
5671 7640  
5672 5732  
5673 1517  
5674 0015  
5675 7640  
5676 5733  
5677 1517  
5700 0157  
5701 7640  
5702 5200  
5703 1517  
5704 0014  
5705 7640  
5706 5734  
5707 1521  
5710 3142  
5711 4432  
5712 0142  
5713 1517  
5714 0045  
5715 7640  
5716 5731  
5717 1001  
5720 0160  
5721 7640  
5722 5216  
5723 1001  
5724 0031

```

/
/ASM02 IS THE HEART OF ASSEMBLY
/IT IS CALLED ONCE FOR EACH ITEM ON
/THE PAGE TABLE,
/IT CONSISTS OF MANY PARTS, ONE FOR
/EACH BASIC TYPE OF INSTR, TO BE
/ASSEMBLED PLUS VARIOUS COMMON EXITS
/
      JMP I    ,+1          /OFF PAGE RETURN
ASM02, 2
      OCA     LFS          /ZERO LFS INDICATOR
      TAD     PASS
      SZA    CLA
      JMS I   GETBAP      /RESTORE BANK & LSTSKP IF PASS2
      JMS I   LFSCHK     /PROCESS LFS IF ANY
      JMS I   ANCHK      /PROCESS BSS 0 IF ANY
KCDF10, CDF 10
      TAD I   PTPRR
KCDF00, CDF 00
      OCA     OP
      TAD I   PTCPR
      AND     K40         /IS IT A SKIP INSTRUCTION
                          /YES ... SET SKIP INDICATOR
      OCA     CURSKP
      TAD I   PTCPR
      AND     K10         /IS IT A PSEUDO OP (PAR)
      SZA    CLA
      JMP I   ASM03I      /YES ... EXIT
      TAD I   PTCPR
      AND     K4          /IS IT A MEMORY REFERANCE INSTRUCTION
      SZA    CLA
      JMP I   ASM04I      /NO
      TAD I   PTCPR      /PT CODE WORD
      AND     K20        /IS AFS A CONSTANT
      SZA    CLA
      JMP     ASM05       /YES
      TAD I   PTCPR      /PT CODE WORD
      AND     K2         /IS AFS A LITERAL
      SZA    CLA
      JMP I   ASM06I      /YES
      TAD I   PTSPR      /AFS ID WORD FOR SYMBOL TABLE
      OCA     AFS        /TO DIRECTLY ADDRESSABLE LOCATION
      JMS I   OBSYM      /GET ITS POINTERS TO MST
      AFS
      TAD I   PTCPR      /PT CODE WORD
      AND     K400       /IS OP INDIRECT
      SZA    CLA
      JMP I   ASM2AI      /YES
      TAD     USE        /AFS MST USE WORD
      AND     K40        /IS AFS OFF BANK (COMMON)
      SZA    CLA
      JMP     ASM08       /YES
      TAD     USE        /AFS MST USE WORD
      AND     K3000

```

/SARR, E15	PAL10	V135	23-SEP-69	11128	PAGE 70-1
4561	5725	7650	SNA CLA		/IS AFS ABSOLUTE
4562	5726	5736	JMP I	ASM29I	/YES
4563	5727	5735	JMP I	ASM07I	/NO
4564	5730	6114	ASM00I,	ASME8	
4565	5731	6036	ASM2AI,	ASM02A	
4566	5732	1200	ASM03I,	PPAR1	
4567	5733	5740	ASM04I,	ASME1	
4568	5734	6252	ASM06I,	ASM06	
4569	5735	6200	ASM07I,	ASM07	
4570	5736	6241	ASM09I,	ASM09	
4571	5737	2476	ANCHK,	ANUMCK	

```

4572 /
4573 / END OF LINE NECESSITIES
4574 /
4575 5740 1140 ASME1, TAD OP
4576 5741 3123 ASME2, DCA TEM1
4577 5742 4456 JMS I WRITEP
4578 5743 4436 JMS I OUTBIN
4579 5744 0123 TEM1
4580 5745 0000 ASME1X, 0
4581 5746 1072 ASM01, TAD LSTSKP /IS LAST INSTRUCTION A SKIP INSTRUCTION
4582 5747 7650 SNA CLA /NO
4583 5750 5360 JHP ,+10 /YES ... LAST BANK
4584 5751 1073 TAD LSTBNK
4585 5752 7041 CIA /+CURRENT BANK
4586 5753 1147 TAD BANK /ARE THEY THE SAME
4587 5754 7650 SNA CLA /YES
4588 5755 5360 JHP ,+3 /NO ... SET BANK UNKNOWN
4589 5756 7240 CLA CMA
4590 5757 3147 DCA BANK
4591 5760 1146 TAD CURSKP /PLACE CUR SKIP INDICATOR
4592 5761 3072 DCA LSTSKP /AS LAST SKIP INDICATOR
4593 5762 1147 TAD BANK /PLACE CURRENT BANK
4594 5763 3073 DCA LSTBNK /IN LAST BANK INDICATOR
4595 5764 1147 TAD BANK /SAVE FOR PROTECTION DURING LISTING
4596 5765 3374 DCA BNKSAV
4597 5766 1072 TAD LSTSKP
4598 5767 3375 DCA SKPSAV
4599 5770 2065 ISE ILC /INCREMENT ILC
4600 5771 5772 JHP I ASM2IX /NO ... SET UP FOR CURRENT INSTRUCTION
4601 5772 5650 ASM2IX, ASM02=1
4602 5773 6600 GETBAP, GETBAS
4603 5774 0000 BNKSAV, 0
4604 5775 0000 SKPSAV, 0
4605 5776 4062 OPISUB, JMS OPIS
4606 5777 4055 OBISUB, JMS OBIS
4607

```



```

4608      6200      *6200
4609      /
4610      /
4611      /
4612      /      INDIRECT DUMMY ARGUMENT
4613      6200      3950      DUMSUB
4614      6201      1620      ASM12, TAD I , -1
4615      6202      3145      DCA      TEMP6
4616      6203      7240      CLA CMA
4617      6204      5210      JMP      ASME3      /SET BANK UNKNOWN
4618      /
4619      /      OFF BANK INDIRECT (INDIRECT COMMON)
4620      /
4621      6025      5777      OBISUB
4622      6026      1605      ASM11, TAD I , -1
4623      6027      3145      DCA TEMP6
4624      /
4625      /
4626      /
4627      6010      3147      TEMP6=EXP
4628      6011      1072      ASME3, DCA      BANK      /SET C(AC) IN BANK
4629      6012      7650      TAD      LSTSKP      /WAS LAST A SKIP?
4630      6013      5223      SNA CLA
4631      /
4632      /      //FOLLOWING 6 LINES HAVE BEEN
4633      6014      5615      JMP I      ,+1
4634      6015      3970      ASMEXT
4635      /
4636      //      JMS I      OUTSKP      /YES, OUTPUT SKP
4637      //      TAD      ILC      /GET PG.LOC,PTR.
4638      //      TAD      K4      /+4
4639      //      AND      K177
4640      //      TAD      K5200A      /OUTPUT JMP ,+4
4641      //      DCA      TEM1
4642      6016      4456      REEASM, JMS I      WRITEP
4643      6017      4436      JMS I      OUTBIN
4644      6020      0123      TEM1
4645      6021      0000      0
4646      6022      2065      ISZ      ILC      /INCREMENT PG.LOC,PTR.
4647      6023      4456      ASME4, JMS I      WRITEP
4648      6024      4436      JMS I      OUTBIN      /OUTPUT JMS TO
4649      6025      0145      TEMP6      /OBISUB,OPISUB, OR DUMSUB
4650      6026      0000      0
4651      6027      2065      ISZ      ILC
4652      6030      4653      JMS I      ASM12B      /PPAR3S
4653      6031      2065      ISZ      ILC
4654      6032      1140      TAD      OP
4655      6033      1256      TAD      K407
4656      6034      5635      JMP I      ASME2P
4657      6035      5741      ASME2P, ASME2
4658      6036      1001      ASM02A, TAD      USE      /AFS MST USE WORD
4659      6037      0157      AND      K20      /IS AFS A DUMMY ARGUMENT
4660      6040      7640      SZA CLA
4661      6041      5201      JMP      ASM10      /YES
4662      6042      1001      TAD      USE      /AFS MST USE WORD
4663      6043      0160      AND      K40      /IS AFS OFF BANK (COMMON)

```

4663	6044	7640	SZA	CLA	
4664	6045	5206	JMP	ASM11	/YES
4665	6046	1001	TAD	USE	/AFS MST USE WORD
4666	6247	2031	AND	K300A	/IS AFS ABSOLUTE
4667	6050	7650	SNA	CLA	
4668	6051	5654	JMP	I ASM12I	/YES
4669	6252	5655	JMP	I ASM13I	/NO
4670	6253	1253	ASM14B,	PPAR3S	
4671	6054	6272	ASM12I,	ASM12	
4672	6055	6313	ASM13I,	ASM13	
4673	6056	2407	K427,	2407	
4674	6057	7240	ASME5,	CLA CMA	
4675	6060	1147	TAD	BANK	
4676	6061	7650	SNA	CLA	
4677	6062	5304	JMP	ASME6*2	
4678	6063	1072	ASME7,	TAD LSTSKP	
4679	6064	7640	SZA	CLA	
4680	6065	5274	JMP	,*7	
4681	6066	4456	JMS	I WRITEP	
4682	6067	4436	JMS	I OUTBIN	
4683	6070	5662	ASMX4,	KCDF00	
4684	6071	2005	ASMX5,	5	
4685	6072	2065	ISZ	ILC	
4686	6073	5302	JMP	ASME6	
4687	6074	4456	JMS	I WRITEP	
4688	6075	4436	JMS	I OUTBIN	
4689	6076	5525	ASMX6,	CDFSKP	
4690	6077	2000	0		
4691	6100	2065	ISZ	ILC	
4692	6101	4437	JMS	I OUTSKP	
4693	6102	1103	ASME6,	TAD TEM7	
4694	6103	3147	DCA	BANK	
4695	6104	1154	TAD	K5	
4696	6105	3271	DCA	ASMX5	
4697	6106	1320	TAD	KCDFA	
4698	6107	3270	DCA	ASMX4	
4699	6110	1316	TAD	CDFSKI	
4700	6111	3276	DCA	ASMX6	
4721	6112	7001	IAC		
4722	6113	3103	DCA	TEM7	
4723	6114	1145	ASME8,	TAD TEMP6	
4724	6115	5717	JMP	I ASME1I	
4725	6116	5525	CDFSKI,	CDFSKP	
4726	6117	5740	ASME1I,	ASME1	
4727	6120	5662	KCDFA,	KCDF00	
4728					

```

4709 /
4710 / SYMBOL TABLE TYPEOUT FLAG TYPEOUT ROUTINE
4711 /
4712 /CALL SEQ.1 TAD USE /GET TYPE BITS
4713 / RTL
4714 / RTL
4715 / JMS STFT
4716 / RETURN
4717 /USED ONLY BY PRSYM
4718 /
4719 6121 0200 STFT, 0
4720 6122 0016 AND K3 /MASK OUT TYPE BITS
4721 6123 7450 SNA
4722 6124 5347 JMP STFT3 /ABSOLUTE SYM,
4723 6125 1172 TAD M3 /CK FOR NEW OPDEF
4724 6126 7450 SNA
4725 6127 5345 JMP STFT2 /YES
4726 6130 7001 IAC
4727 6131 7450 SNA
4728 6132 5372 JMP STFT5 /EXTERNAL
4729 6133 7630 SZL CLA
4730 6134 5341 JMP STFT1 /DEFINED
4731 6135 1367 TAD K2516 /"UN"
4732 6136 4423 JMS I CTYPE
4733 6137 1370 TAD K0406 /"DF"
4734 6140 4423 STFT0, JMS I CTYPE /TYPE FLAG
4735 6141 7200 STFT1, CLA /WE MUST HAVE A CLEAR AC
4736 6142 5721 JMP I STFT /RETURN
4737 6143 2560 K2560, 2560
4738 6144 1720 K1720, 1720
4739
4740 6145 1344 STFT2, TAD K1720 /TYPE "OP"
4741 6146 5340 JMP STFT0
4742 6147 1001 STFT3, TAD USE
4743 6150 0160 AND K40
4744 6151 7640 SZA CLA
4745 6152 5357 JMP STFT4 /COMMON
4746 6153 1363 TAD K0102 /"AB"
4747 6154 4423 JMS I CTYPE
4748 6155 1365 TAD K2300 /"S "
4749 6156 5340 JMP STFT0
4750 6157 1364 STFT4, TAD K0317 /"CO"
4751 6160 4423 JMS I CTYPE
4752 6161 1366 TAD K1500 /"M "
4753 6162 5340 JMP STFT0
4754
4755 6163 0102 K0102, 102
4756 6164 0317 K0317, 317
4757 6165 2300 K2300, 2300
4758 6166 1500 K1500, 1500
4759 6167 2516 K2516, 2516
4760 6170 0406 K0406, 406
4761 6171 0530 K0530, 530
4762 0047 K2400=SLITAB
4763

```

/SABR,815	FAL10	V135	23-SEP-69	11128	PAGE 73-1	
4764	6172	1371	STFT5,	TAD	K0530	/"EX"
4765	6173	4423		JMS I	CTYPE	
4766	6174	1047		TAD	K2400	
4767	6175	5340		JMP	STFT0	

```

4768                    6220      *622/
4769                    /
4772                    /      LOCAL DIRECT REFERENCE
4771                    /
4772      6220      4450      ASH07,    JMS I    SFSTAB      /IS AFS ON PST
4773      6221      5220                    JMP      ASH07A      /NO ... ROUTINE PLACES IT THERE
4774      6222      1514                    TAD I    PSTCPR      /PST CODE WORD
4775      6223      7720                    SMA CLA              /IS SYMBOL ON PAGE
4776      6224      5220                    JMP      ASH07A      /NO
4777      6225      1140                    TAD      OP          /OP CODE
4778      6226      1271                    TAD      M2402A      /IS IT A JMS
4779      6227      7640                    SZA CLA              /
4780      6210      5213                    JMP      ,+3        /NO...EXIT
4781      6211      7201                    CLA IAC              /YES... SET BANK TO CURRENT
4782      6212      3147                    DCA      BANK
4783
4784      6213      4333                    JMS      NSCHK
4785      6214      1002                    TAD      VAL          /AFS MST VALUE
4786      6215      0163                    AND      K177        /SAVE PAGE ADDRESS
4787      6216      1164                    TAD      K200        /ADD PAGE BIT
4788      6217      5660                    JMP I    AS0214      /
4789      6220      1521                    ASH07A, TAD I    PTSPR      /ACTUAL SYMBOL
4790      6221      3151                    DCA      S1          /TO 2 WORD LITERAL TABLE SEARCH LOCATION
4791      6222      4333                    JMS      NSCHK
4792      6223      7126                    CLL RTL
4793      6224      1014                    TAD      K2          /RELOCATABLE SEARCH
4794      6225      5234                    JMP      ASM05C+3
4795      6226      4333                    ASH09B, JMS      NSCHK
4796      6227      1002                    TAD      VAL          /DIRECT NON-PAGE 0 ABSOLUTE
4797      6230      7410                    SKP
4798
4799                    /
4800                    /
4801                    /      NON PAGE ZERO CONSTANT ADDRESS
4802                    /
4802      6231      1521                    ASH05C, TAD I    PTSPR      /ACTUAL BINARY CONSTANT
4803      6232      3151                    DCA      S1          /TO 2 WORD LITERAL TABLE SEARCH LOCATION
4804      6233      7201                    CLA IAC
4805      6234      3150                    DCA      S0          /ABSOLUTE SEARCH
4806      6235      4670                    JMS I    SERALT      /GET A PAGE ADDRESS
4807      6236      1045                    TAD      K400        /ADD INDIRECT BIT
4808      6237      3145                    DCA      TEMP6
4809      6240      5651                    JMP I    AS0011      /EXIT FOR SKIP CHECK
4810
4811                    /
4812                    /      DIRECT ABSOLUTE OR EXTERNAL
4813                    /
4813      6241      4333                    ASH09,    JMS      NSCHK
4814      6242      1002                    TAD      VAL          /ABSOLUTE SYMBOL VALUE
4815      6243      0166                    AND      K7600        /IS SYMBOL ON PAGE ZERO
4816      6244      7640                    SZA CLA
4817      6245      5226                    JMP      ASM09B      /NO
4818      6246      4333                    JMS      NSCHK
4819      6247      1002                    TAD      VAL          /ADD IN PAGE ZERO ADDRESS
4820      6250      5660                    JMP I    AS0014      /
4821      6251      6057                    AS0011, ASME5
4822                    /

```

```

/SABR,615      PAL10  V135  23-SEP-69      11126  PAGE 74-1
4823           /      LITERAL FOR AN ADDRESS FIELD SYMBOL
4824           /
4825      6252  1521  ASM06,  TAD I  PYSR      /ACTUAL LITERAL
4826      6253  3151          DCA   S1      /TO 2 WORD LITERAL TABLE SEARCH LOCATION
4827      6254  7201          CLA IAC
4828      6255  3150          DCA   S0      /ABSOLUTE SEARCH
4829      6256  4670          JMS I  SERALT  /GET A PAGE ADDRESS
4830      6257  5660          JMP I  AS0014 /EXIT FOR SKIP CHECK
4831      6260  5740  AS0014, ASME1
4832      6261  0000  OUTSK,  0
4833      6262  4456          JMS I  WRITEP
4834      6263  4436          JMS I  OUTBIN
4835      6264  3657          K7410
4836      6265  0000          0
4837      6266  2065          ISZ ILC
4838      6267  5661          JMP I  OUTSK
4839      6270  3443  SERALT,  SRALT
4840      6271  4000  M2402A,  =4000
4841           /
4842           /      INDIRECT ABSOLUTE
4843           /
4844      6272  1002  ASM12,  TAD   VAL      /AFS MST USE WORD
4845      6273  0166          AND   K7600  /IS IT ON PAGE ZERO
4846      6274  7640          SZA CLA
4847      6275  5304          JMP   ASM12F  /NO
4848           /
4849           /      INDIRECT PAGE ZERO ABSOLUTE SYMBOL
4850           /
4851      6276  4333          JMS   NSCHK
4852      6277  1002          TAD   VAL      /SAVE PAGE ZERO ADDRESS
4853      6300  1045  ASM12E,  TAD   K400  /ADD INDIRECT BIT
4854      6301  3145          DCA   TEMP6
4855      6302  5651          JMP I  AS0011
4856           /
4857           /      INDIRECT NON PAGE ZERO ABSOLUTE SYMBOL
4858           /
4859      6303  5776          OPISUB
4860      6304  1703  ASM12F,  TAD I  ,=1
4861      6305  3145          DCA   TEMP6
4862      6306  1147          TAD   BANK
4863      6307  5710          JMP I  AS0013  /EXIT FOR SKIP CHECK
4864      6310  6010  AS0013,  ASME3
4865      6311  5376  K5376,  5376
4866      6312  5377  K5377,  5377
4867           /
4868           /      LOCAL INDIRECT REFERENCE
4869           /
4870      6313  4450  ASM13,  JMS I  SPSTAB  /IS AFS ON PST
4871      6314  5327          JMP   ASM14  /NO ... MUST BE OFF PAGE
4872      6315  1514          TAD I  PSTCPR  /YES ... PST CODE WORD
4873      6316  7720          SMA CLA  /IS AFS ON PAGE
4874      6317  5327          JMP   ASM14  /NO
4875      6320  4333          JMS   NSCHK
4876      6321  1002          TAD   VAL      /AFS VALUE FROM MST
4877      6322  0163          AND   K177  /SAVE PAGE DISPLACEMENT

```

4878	6323	1027	TAD	K600	/ADD PAGE AND INDIRECT BIT
4879	6324	3145	DCA	TEMP6	/SAVE
4880	6325	5651	JMP I	AS0011	/GO OUTPUT INSTRUCTION
4881			/		
4882			/		
4883			/		
4884					
4885	6326	5776	OPISUB		
4886	6327	1726	ASM14,	TAD I ,=1	
4887	6330	3145		DCA TEMP6	
4888	6331	7201		CLA IAC	/SET BANK TO CURRENT
4889	6332	5710		JMP I AS0013	/EXIT FOR SKIP CHECK
4890	6333	0000	NSCHK,	?	
4891	6334	1517		TAD I PTCPR	
4892	6335	2043		AND K2000	
4893	6336	7640		SZA CLA	
4894	6337	7001		IAC	/ITS A #
4895	6340	5733		JMP I NSCHK	
4896					

```

4897
4898
4899      6341  4427      PIF,   JMS I   GETSYM
4900      6342  5357      JMP     PIFERR      /NOTHING THERE
4901      6343  5346      JMP     ,S3         /SYM
4902      6344  7000      NOP
4903      6345  5357      JMP     PIFERR      /CON
4904      6346  1061      TAD     CHR         /LIT
4905      6347  1167      TAD     M254        /CK FOR COMMA
4906      6350  7640      SZA CLA
4907      6351  5575      IERROR
4908      6352  2010      ISZ     X0          /NOT A COMMA
4909      6353  4427      JMS I   GETSYM     /PREVENT FLAGGING COMMA
4910      6354  7000      NOP
4911      6355  7410      SKP
4912      6356  7410      SKP
4913      6357  5575      PIFERR, IERROR    /YES, SET CTR TO SKIP N LINES
4914      6360  1001      TAD     USE
4915      6361  0045      AND     K400
4916      6362  7640      SZA CLA
4917      6363  3150      DCA     S0          /I WANT A NUMBER
4918      6364  4446      JMS I   SKIPL      /IS SYMBOL DEFINED?
4919      6365  1150      TAD     S0
4920      6366  7041      CIA
4921      6367  3371      DCA     IFCTR      /YES, CONTINUE NORMAL ASSMBLY
4922      6370  5434      JMP I   NULLP      /GET THE NUM.
4923
4924      6371  0000      IFCTR, 0
4925
4926
4927      /LAP & EAP PSUEDO=OPS
4928
4929      6372  4446      PLAP,   JMS I   SKIPL
4930      6373  7001      IAC
4931      6374  7410      SKP
4932      6375  4446      PEAP,   JMS I   SKIPL      /LEAVE AUTO-PAGING MODE
4933      6376  3102      DCA     APMSW
4934      6377  5434      JMP I   NULLP      /ENTER AUTO-PAGING MODE
4935

```



4936	6400	1411	LISTON,	1411
4937	6401	2324		2324
4938	6402	1116		1116
4939	6403	0700		0700
4940	6404	1716		1716
4941	6405	0000		0000
4942	6406	1011	HISP,	1011
4943	6407	7710		7710
4944	6410	0023		0023
4945	6411	2005		2005
4946	6412	0504		0504
4947	6413	0020		0020
4948	6414	2516		2516
4949	6415	0310		0310
4950	6416	7700		7700
4951	6417	0022	RDER,	0022
4952	6420	0501		0501
4953	6421	0405		0405
4954	6422	2277		2277
4955	6423	0000		0000

/SABR.815

PAL10

V135

23-SEP-69

11128

PAGE 77

```

4956 / PART OF MAIN PROGRAM
4957 / RECOUNT ROUTINE
4958 / FOLLOWING CODE CLEANS UP PST CODES BEFORE RECOUNTING
4959 6424 7041 CLNPST, CIA
4960 6425 3256 DCA IOINIT /SET COUNTER
4961 6426 1173 TAD PSTBSE
4962 6427 7001 IAC
4963 6430 3114 DCA PSTCPR /CODE POINTER
4964 6431 1514 TAD I PSTCPR /LOOP
4965 6432 0242 AND K4003K /KILL OBAC DUE TO THIS SYM.
4966 6433 3514 DCA I PSTCPR
4967 6434 2114 ISZ PSTCPR /MOVE PTR
4968 6435 2114 ISZ PSTCPR
4969 6436 2256 ISZ IOINIT
4970 6437 5231 JMP ,=6 /NOT DONE
4971 6440 5641 JMP I ,+1
4972 6441 0447 RECRET
4973 6442 4003 K4003K, 4003
4974 /
4975 /
4976 /COMMON EXIT FOR DATA-GENERATING PSUEDO-OPS
4977 /
4978 6443 1137 POPEX, TAD LFS /CK FOR TAG
4979 6444 7650 SNA CLA
4980 6445 5433 JMP I DCIL1 /NO TAG
4981 6446 7040 CMA /DECREMENT PTSZE
4982 6447 1122 TAD PTSZE
4983 6450 3122 DCA PTSZE
4984 6451 4444 JMS I RECTI /YES RECOUNT THE PAGE
4985 6452 2122 ISZ PTSZE /RESTORE PTSZE
4986 6453 4655 JMS I ISZPTX /RESTORE PT PTRS
4987 6454 5433 JMP I DCIL1 /RETURN FOR NEXT LINE
4988 6455 6701 ISZPTX, ISZPT

```

4989  
 4990  
 4991  
 4992  
 4993  
 4994  
 4995  
 4996  
 4997  
 4998  
 4999  
 5000  
 5001  
 5002  
 5003  
 5004  
 5005  
 5006  
 5007  
 5008  
 5009  
 5010  
 5011  
 5012  
 5013  
 5014  
 5015  
 5016  
 5017  
 5018  
 5019  
 5020  
 5021  
 5022  
 5023  
 5024  
 5025  
 5026  
 5027  
 5028  
 5029  
 5030  
 5031  
 5032  
 5033  
 5034  
 5035  
 5036  
 5037  
 5038  
 5039  
 5040  
 5041  
 5042  
 5043

4423  
 6456 2000  
 6457 6201  
 6460 1370  
 6461 3054  
 6462 5774  
 6463 4424  
 6464 1371  
 6465 3011  
 6466 1375  
 6467 4331  
 6470 6476  
 6471 1373  
 6472 3011  
 6473 1375  
 6474 4331  
 6475 6476  
 6476 4347  
 6477 1364  
 6500 1365  
 6501 3766  
 6502 4424  
 6503 1371  
 6504 3011  
 6505 1376  
 6506 4331  
 6507 6510  
 6510 4347  
 6511 5315  
 6512 1054  
 6513 3042  
 6514 5327  
 6515 1367  
 6516 3042  
 6517 4424  
 6520 1372  
 6521 3011  
 6522 1377  
 6523 4331  
 6524 6525  
 6525 4347  
 6526 7001  
 6527 3331  
 6530 5656  
 6531 2000  
 6532 3125

/ROUTINE TO INITIALIZE I/O DEVICES

C2=JMS I CTYPE

IOINIT, 0  
 CDF 00  
 TAD JL64  
 DCA TYPE  
 VN, JMP I VERSI  
 IOI, JMS I CRLF  
 TAD JHISP  
 DCA X1  
 TAD M5  
 JMS QUERY  
 RGO RGO  
 TAD JRDER  
 DCA X1  
 TAD M5  
 JMS QUERY  
 RGO RGO  
 JMS KSR  
 TAD JHSR  
 TAD JASR  
 DCA I INDEVP  
 JMS I CRLF  
 TAD JHISP  
 DCA X1  
 TAD M11  
 JMS QUERY  
 ,+1  
 JMS KSR  
 JMP ,+4  
 TAD TYPE  
 DCA PUNCH  
 JMP IOX  
 TAD JL63  
 DCA PUNCH  
 JMS I CRLF  
 TAD JLIST  
 DCA X1  
 TAD M17  
 JMS QUERY  
 ,+1  
 JMS KSR  
 IAC  
 IOX, DCA LSTDEV  
 JMP I IOINIT  
 QUERY, 0  
 DCA JCOUNT

/1 = PUNCH, 2 = TYPE

5044	6533	1411	TAD I	X1
5045	6534	4423	C2	
5046	6535	6031	KSF	
5047	6536	7410	SKP	
5048	6537	5344	JMP	,=5
5049	6540	2125	ISZ	JCOUNT
5050	6541	5333	JMP	,=6
5051	6542	2331	ISZ	QUERY
5052	6543	5731	JMP I	QUERY
5053	6544	1731	TAD I	QUERY
5054	6545	3331	DCA	QUERY
5055	6546	5731	JMP I	QUERY
5056	6547	0000	0	
5057	6550	6031	KSF	
5058	6551	5350	JMP	,=1
5059	6552	6036	KRB	
5060	6553	3123	DCA	TEM1
5061	6554	1123	TAD	TEM1
5062	6555	4454	JMS I	TYPE
5063	6556	1123	TAD	TEM1
5064	6557	1363	TAD	M331
5065	6560	7640	S2A	CLA
5066	6561	2347	ISZ	KSR
5067	6562	5747	JMP I	KSR
5068		6531	LSTDEV=	QUERY
5069	6563	7447	M331,	=331
5070	6564	7751	JMSR,	MSR=ASR
5071	6565	6637	JASR,	ASR
5072	6566	4702	INDEVP,	INDEV
5073	6567	3164	JL63,	L63
5074	6570	4772	JL64,	L64
5075		0125	JCOUNT=	TEM3
5076	6571	6405	JHISP,	HISP=1
5077	6572	6377	JLIST,	LISTON=1
5078	6573	6416	JRDER,	RDER=1
5079	6574	7000	VERSI,	VERNUM
5080	6575	7773	M5,	=5
5081	6576	7767	M11,	=11
5082	6577	7761	M17,	=17
5083				

KSR,

/0="YES"  
/NOT "YES"

```

5084          6620      *6620
5085          //
5086          //PART OF MAIN PROGRAM MOVED FOR V03
5087          //
5088          //ROUTINE TO RESTORE BANK AND LSTSKP FOR PASS2
5089          //WILL NOT FIT INTO ASM22 WHERE IT BELONGS
5090
5091          6620      2000      GETRAS, 0
5092          6621      1606          TAD I   BNKSAP
5093          6622      3147          DCA     BANK
5094          6623      1607          TAD I   SKPSAP
5095          6624      3072          DCA     LSTSKP
5096          6625      5600          JMP I   GETBAS
5097
5098          6626      5774          BNKSAP, BNKSAV
5099          6627      5775          SKPSAP, SKPSAV
5100
5101          //INPUT ROUTINES
5102
5103          6610      0000      HSR, 0
5104          6611      3255          DCA     TEM10      //CLR TIMER
5105          6612      6014          RFC
5106          6613      6011      HSR1, RSF
5107          6614      5217          JMP     HSR2
5108          6615      6012          RRB
5109          6616      5610          JMP I   HSR
5110          6617      3237      HSR2, DCA     ASR      //WASTE SOME TIME
5111          6620      2255          ISZ    TEM10      //CK TIMER
5112          6621      5213          JMP     HSR1      //KEEP TRYING
5113          6622      1013      REXIT, TAD     X3      //CK FOR EMPTY BUFFER
5114          6623      1257          TAD     BUFBEQ
5115          6624      7640          SZA   CLA
5116          6625      5231          JMP     ,+4      //NO, WE HAVE A PARTIAL BUFFER
5117          6626      6201          CDF   00
5118          6627      5630          JMP I   ,+1      //YES TAPE HAS ENDED WITH NO END STATMT
5119          6630      2701          ERRE
5120          6631      3413          DCA I  X3      //FILL END OF BUFFER WITH 0'S
5121          6632      1013          TAD     X3
5122          6633      1261          TAD     BUFEN
5123          6634      7640          SZA   CLA
5124          6635      5231          JMP     ,+4
5125          6636      5660          JMP I   RG3P      //NOW RET. FOR PROCESSING
5126
5127          6637      0000      ASR, 0
5128          6640      1254          TAD     M50
5129          6641      3256          DCA     TEM11
5130          6642      3255          DCA     TEM10
5131          6643      6031      ASR1, KSF
5132          6644      5247          JMP     ASR2
5133          6645      6036          KRB
5134          6646      5637          JMP I   ASR
5135          6647      2255      ASR2, ISZ    TEM10
5136          6650      5243          JMP     ASR1
5137          6651      2256          ISZ    TEM11
5138          6652      5242          JMP     ASR1=1

```

5139	6653	5222	JMP	REXIT
5140				
5141				
5142	6654	7730	MS0,	=50
5143	6655	0000	TEM10,	0
5144	6656	0000	TEM11,	0
5145	6657	6701	RUFREG,	1=DATA
5146	6660	4677	RG3P,	RG3
5147	6661	6133	RUFEN,	1=LINBUF

```

5148 /
5149 / ENTRY PSEUDO OPERATION
5150 /
5151 6662 4427 PENTRY, JMS I GETSYM /GET NEXT INPUT ITEM
5152 6663 7410 SKP /NOTHING THERE
5153 6664 5267 JMP ,43 /SYMBOL
5154 6665 7200 NOP /CONSTANT
5155 6666 5575 IERROR /LITERAL
5156 6667 4446 JMS I SKIPL
5157 6670 4451 JMS I SREST /PLACE SYMBOL ON EXTERNAL SYMBOL TABLE
5158 6671 7200 CLA
5159 6672 1001 TAD USE /AFS MST USE WORD
5160 6673 4320 AND K403 /SAVE SYMBOL LENGTH (& DEF. BIT FOR PASS 2)
5161 6674 1277 TAD K2220 /ADD IN PROPER BITS
5162 6675 3001 DCA USE /FOR NEW MST USE WORD
5163 6676 5434 JMP I NULLP /EXIT FOR NEXT LINE
5164 6677 2220 K2220, 2220
5165 6700 403 K403, 403
5166 /
5167 /
5168 /
5169 /INCREMENT PAGE TABLE POINTERS
5170 /
5171 6721 0000 ISZPT, ?
5172 6742 2117 ISZ PTCPR /INCREMENT PAGE TABLE CODE POINTER BY 2
5173 6743 2117 ISZ PTCPR
5174 6724 2121 ISZ PTSPR /INCREMENT PAGE TABLE SYMBOL POINTER BY 2
5175 6745 2121 ISZ PTSPR
5176 6726 2120 ISZ PTOPR /INCREMENT PT OP CODE POINTER
5177 6727 5701 JMP I ISZPT
5178 /
5179 /CK CONSTANT FOR BLOCK PSEUDO=OP
5180 /
5181 6710 1102 PBSS4, TAD APMSW /AUTOMATIC PAGING?
5182 6711 7650 SNA CLA
5183 6712 1014 TAD K2 /YES, 176 IS MAXIMUM
5184 6713 1166 TAD M200 /NO, 200 IS MAX
5185 6714 1150 TAD S0 /CHECK CONSTANT
5186 6715 7740 SMA SZA CLA
5187 6716 5575 IERROR /TOO BIG
5188 6717 1150 TAD S0 /IS CONSTANT 0?
5189 6720 7450 SNA
5190 6721 5723 JMP I PBSS2J /YES, EQUIVALENCE TAG
5191 6722 5724 JMP I PBSS5I /NO, CREATE BLOCK OF THIS SIZE
5192 6723 2435 PBSS2J, PBSS2
5193 6724 2610 PBSS5I, PBSS5
5194 /
5195 /
5196 /PATCH TO DELETE DEFINED BIT IN PST FOR A TAG
5197 /EQUIVALENCED TO A LINE THAT OVERFLOWED THE PAGE
5198 /
5199 6725 1121 EQVFIX, TAD EQVBIT /WAS THERE SUCH A TAG?
5200 6726 7650 SNA CLA
5201 6727 5736 JMP I L55CP /NO
5202 6730 1501 TAD I EQVBIT /YES, GET PST CODE FOR THIS TAG

```

```

/SABR,815      PAL12      V135      23-SEP-69      11128      PAGE 80-1
5203      6731      1041      TAD      K4000      /CANCEL DEFINED BIT
5204      6732      3501      DCA I    EQVBIT
5205      6733      1164      TAD      K2000      /SET EQUIVALENCE BIT FOR NEXT LINE
5206      6734      3101      DCA      EQVBIT      /WHEN NEXT PAGE GETS GOING
5207      6735      5736      JMP I    L55CP      /RETURN TO ASSEMBLE THE PAGE WE HAVE
5208      6736      2405      L55CP,   L55C
5209
5210
5211
5212
5213      /
5214      /CK FOR TYPE OF SYMBOL
5215      /CALL SEQ1      JMS WHATYP
5216      /
5217      /
5218      /
5219      /
5220      /
5221      /
5222      /
5223      /
5224      /
5225      /
5226      /
5227      /
5228      /EXECUTE ISZ GTSYM (MOVE RETURN POINTER) ONLY IF IFCTR .GE. 0
5229      /OTHERWISE MOVE LINE PTR TO NEXT SLASH, SEMI-COLON OR CAR,RET.
5230      /
5231      /
5232      /
5233      /
5234      /
5235      /
5236      /
5237      /
5238      /
5239      /
5240      /
5241      /
5242      /
5243      /
5244      /
5245      /
5246      /
5247      /
5248      /
5249      /
5250      /
5251      /
5252      /
5253      /
5254      /

```



5255 6776 \*6776

5256 /PAGE SYMBOL TABLE (224 WORDS)  
5257 /DOUBLE WORD ENTRIES  
5258 /REQUIRE FOR EACH CORE PAGE OF CODE  
5259 /EVERY SYMBOL DEFINED OR REFERENCED ON  
5260 /GIVEN PAGE IS ENTERED  
5261 /TYPICAL ENTRY\*1 WD1=SYMBOL ID  
5262 / WD2=CODE BITS  
5263 /  
5264 /SYMBOL ID=ADDRESS OF SYMBOL ENTRY IN MAIN SYM. TAB  
5265 /CODE1 BIT0=1 IF SYM. DEF. ON CUR. PAGE  
5266 / BIT11=1 IF SYM REFERENCED NORMALLY BY A MRI ON THE PG.  
5267 / BIT10=1 IF SYM. REF'D, WITH A #  
5268 / BITS 1-9 USED FOR COUNTING AMOUNT OF OBACK  
5269 / WHICH IS DUE TO THIS SYMBOL  
5270 /SYMBOLS ARE ENTERED ON PST IN ORDER OF APPEARANCE  
5271 /IN SOURCE  
5272 /NO MORE THAN 64 (DEC) SYMBOLS MAY BE REF'D,  
5273 /ON ANY PAGE.  
5274 /NOTE! THE SIZE OF THIS TABLE SHOULD NOT BE  
5275 /INCREASED UNLESS LFS TABLE IS ALSO INCREASED.  
5276

5277 7176 \*7176

5278 /PAGE TABLE (402 WORDS)  
5279 /DOUBLE WORD ENTRIES  
5280 /ONE ENTRY FOR EACH INSTRUCTION TO BE ASSEMBLED  
5281 /ROOM FOR 1 EXTRA ENTRY TO COVER PAGE CVERFLOW  
5282 /A NEW TABLE FOR EACH PAGE OF CODE  
5283 /TYPICAL ENTRY: WD1=CODE BITS  
5284 / WD2=SYMBOL WORD  
5285 /  
5286 /CODE1 BIT1=1 IF # REF  
5287 / BIT2=1 IF CDF TO CUR BANK  
5288 / BIT3=1 IF INDIRECT  
5289 / BIT4=1 IF BLOCK 0 (FOR EQUIVALENCED TAGS)  
5290 / BIT5=1 IF SPECIAL CALL CONST  
5291 / BIT6=1 IF SKIP INST.  
5292 / BIT7=1 IF AFS IS CONST  
5293 / BIT8=1 IF PARAMETER  
5294 / BIT9=1 IF OPR OR IOT INST.  
5295 / BIT10=1 IF AFS IS LITERAL  
5296 / BIT11=1 IF LFS OCCURS  
5297 / BIT0 UNUSED  
5298 /  
5299 /THE SYMBOL WORD=0 IF CODE BIT9=1  
5300 / =THE ACTUAL CONST OR LITERAL IF BITS2,5,7 OR 10=1  
5301 / =THE SYMBOL ID (MST ENTRY ADDR.) FOR AN ADDR, PARAMETER  
5302 / OR FOR THE AFS OF AN MRI

5323  
5324  
5325  
5326  
5327  
5328  
5329  
5310 7151  
5311 7151 1122  
5312 7152 6211  
5313 7153 3775  
5314 7154 1071  
5315 7155 7106  
5316 7156 7006  
5317 7157 7006  
5318 7160 2375  
5319 7161 1111  
5320 7162 3775  
5321 7163 1074  
5322 7164 2375  
5323 7165 3775  
5324 7166 1075  
5325 7167 2375  
5326 7170 3775  
5327 7171 2375  
5328 7172 6201  
5329 7173 5774  
5330 7174 2220  
5331 7175 4000

/THIS ROUTINE IS INCLUDED FOR DEBUGGING PURPOSES ONLY  
/IMPLEMENT WITH  
/CPGES+1, JMP I ,+1  
/  
/AND SET "DUMP" ABOVE HIGHEST LOC OF SYM TAB EXPECTED TO BE USED  
/(DUMP AREA MUST STAY BETWEEN MST AREA AND OCC, TAB)  
/ THIS ROUTINE FITS INTO TOP OF PST (USUALLY OK)  
\*7151

TAD PTSZE  
CDF 10  
DCA I DUMP  
TAD LTSZE  
CLL RTL  
RTL  
RTL  
ISZ DUMP  
TAD PGEESC  
DCA I DUMP  
TAD OBACKR  
ISZ DUMP  
DCA I DUMP  
TAD OPSCTR  
ISZ DUMP  
DCA I DUMP  
ISZ DUMP  
CDF 00  
JMP I ,+1  
CPGES+3  
DUMP, 4000

5332  
5333  
5334  
5335  
5336  
5337  
5338  
5339 7000  
5340  
5341 7000 4424  
5342 7001 1212  
5343 7002 3011  
5344 7073 1214  
5345 7004 4613  
5346 7025 6476  
5347 7006 1216  
5348 7077 3615  
5349 7010 5611  
5350 7011 6463  
5351 7012 7016  
5352 7013 6531  
5353 7014 7763  
5354 7015 6462  
5355 7016 7000  
5356 7017 2004  
5357 7020 2055  
5358 7021 7040  
5359 7022 2301  
5360 7023 2222  
5361 7024 4004  
5362 7025 2503  
5363 7026 5560  
5364 7027 7055  
5365 7030 2162  
5366 7031 2462  
5367 7032 5561  
5368 7033 6640

/  
/TYPE VERSION NUMBER  
/(THIS IS ONCE ONLY CODE)  
/(OVERWRITTEN BY P.S.T.)  
/

\*7000  
/  
VERNUM, JMS I CRLF  
TAD JVERS  
DCA X1  
TAD M26  
JMS I MTYPE  
RGO  
TAD K7000X  
DCA I VNOP  
JMP I ,\*1  
IOI

JVERS, VERSN=1  
MTYPE, QUERY  
M26, =15  
VNOP, VN  
K7000X, NOP  
VERSN, 2004

/PDP=8 SABR DEC=08=A2C2=V#

/\* VERSION # (1ST DIGIT)  
/2ND DIGIT

/SABR,815

PAL10

V135

23-SEP-69

11128

PAGE 84

5369  
5370  
5371  
5372  
5373  
5374  
5375

/SABR BANK 1 SECTION

/TABLES





5376 0001 FIELD 1

5377

5378

5379

5380

0000 0000

\*0  
EQUFB, 0

5381

5382

5383

5384

5385

5386

5387

5388

5389

5390

5391

5392

5393

5394

5395

5396

5397

5398

0100 \*100

5399

5400

/BSEEST, 0

5401

5402

5403

5404

5405

5406

5407

5408

5409

5410

5411

0200 \*200

5412

5413

/PTOPTB, 0

5414

5415

5416

5417

5418

5419

5420

5421

5422

5423

5424

5425

/EQUIVALENCE TABLE  
 /100 WORDS  
 /TABLE IS REINITIALIZED BEFORE EACH PAGE BEGINS  
 /COLLECTION, IF NO EQUIV, IS LEFT FROM PREVIOUS PAGE  
 /MULTIPLE WORD ENTRIES  
 /ONE ENTRY IS MADE FOR  
 /EACH LOC. TAG WHICH HAS  
 /EQUIVALENTS  
 /1ST WORD OF EACH ENTRY  
 /CONTAINS NO. OF OTHER WORDS  
 /IN THE ENTRY  
 /OTHER WORDS ARE SYMBOL ID'S  
 / (MST ADDRESSES) OF SYMBOLS  
 /EQUIVALENT TO THE PARTICULAR  
 /LOCATION TAG

/EXTERNAL SYMBOL TABLE  
 /100 WORDS  
 /SINGLE WORD ENTRIES  
 /CONSISTING OF THE SYMBOL ID (MST ADDRESS)  
 /EACH EXT, SYM, IS ENTERED IN  
 /THE TABLE WHEN IT FIRST  
 /OCCURS IN THE SOURCE AND  
 /ASSIGNED A LOCAL EXT, NUMBER  
 /ACCORDING TO ITS PLACE IN THE  
 /TABLE.

/PAGE OP CODE TABLE  
 /200 WORDS  
 /SINGLE WORD ENTRIES  
 /ONE FOR EACH ENTRY IN PAGE TABLE  
 /ENTRY=ACTUAL OP CODE FOR  
 /ALL MRI, OPR OR IOT'S  
 /OR ? FOR ALL PARAMETERS  
 /NEW TABLE FOR EACH PAGE OF CODE  
 /NOTE! THIS TABLE MAY OVERFLOW BY 1 WORD DURING COLLECTION  
 /OVERFLOW CAUSED BY PUTTING INFO ON TABLE BEFORE CK FOR OVERFLI  
 /NO HARM IF ASSEM, PHASE LIT, TAB FOLLOWS

5426  
5427  
5428  
5429  
5430  
5431  
5432  
5433  
5434  
5435  
5436  
5437  
5438  
5439  
5440  
5441  
5442  
5443  
5444  
5445  
5446  
5447  
5448  
5449  
5450  
5451  
5452  
5453  
5454  
5455  
5456  
5457  
5458  
5459  
5460  
5461  
5462  
5463  
5464  
5465  
5466  
5467  
5468  
5469  
5470  
5471  
5472  
5473  
5474  
5475  
5476

2400

\*400

/LITBSE, 0

/ASSEMBLY PHASE LITERAL TABLE  
/200 WORDS  
/DOUBLE WORD ENTRIES  
/MUST BE SEPARATE FROM COLL.  
/PHASE LIT, TAB, BECAUSE BOTH  
/GOING AT ONCE IN PASS 2.  
/THIS TABLE CONTAINS NOT  
/ONLY LITERALS BUT ALSO  
/OFF PAGE POINTERS  
/1ST WORD OF ENTRY = 1 OR 2 OR 6  
/1 MEANS LITERAL &  
/2ND WORD CONTAINS ACTUAL VALUE  
/2 MEANS OFF PAGE SYMBOL PTR  
/3 2ND WORD CONTAINS SYMBOL ID,  
/6 MEANS OFF PAGE SYM, PTR  
/WHERE SYMBOL REFERENCED BY A #  
/2ND WORD AS FOR 2  
/TABLE BUILT ANEW FOR EACH  
/PAGE OF CODE.

0600

\*600

/LFSBSE, 0

/LOC FIELD SYMBOL TABLE  
/100 WORDS  
/SINGLE WORD ENTRIES  
/EACH SYMBOL ID (MST ADDRESS)  
/OF THE GIVEN LFS  
/LFS'S ARE ENTERED IN ORDER  
/OF THEIR APPEARANCE IN SOURCE  
/TABLE REBUILT FOR EACH PAGE OF CODE  
/NOTE! THIS TABLE MUST BE AT LEAST AS LONG  
/AS THE PST TO PREVENT LFS OVERFLOW

2700

\*700

0700 2000

PEBSE, 0

/PAGE ESCAPE PUSH DOWN LIST  
/40 WORDS  
/SINGLE WORD ENTRIES  
/EACH ENTRY IS 0,2, OR 4  
/BEING THE VALUE OF THE  
/PAGE ESCAPE (0,2,OR 4 WORDS)  
/OF EACH PAGE ASSEMBLED  
/THESE NOS, ARE SAVED DURING  
/PASS 1 & USED DURING  
/PASS 2



```

5477      .740      *740
5478      /SORT LISTS
5479
5480      /SORT LIST FOR INITIAL CHAR, OF LITERAL
5481      0740      0242      SL3,      242      /QUOTE
5482      0741      0255      255      /MINUS
5483      0742      0304      304      /D
5484      0743      0313      313      /K
5485      0744      7777      -1      /SORT LIST MUST BE FOLLOWED BY A NEGATIVE
5486
5487
5488
5489      /BRANCH LISTS
5490      0745      4025      BL6,      RLN15
5491      0746      4007      RLN2
5492      0747      4007      RLN2
5493      0750      4014      RLN3
5494      0751      4014      RLN3
5495
5496      /SORT LIST FOR BEGINNING OF INPUT ITEM
5497      0752      0255      SL2,      255      /MINUS
5498      0753      0250      250      /LEFT PARIN
5499      0754      0242      242      /QUOTE
5500      0755      0273      SL6,      273      /SEMI-COLON
5501      0756      0257      257      /SLASH
5502      0757      0240      SL1,      240      /SPACE
5503      0760      0211      211      /TAB
5504      0761      0000      000      /CR
5505      0762      7777      -1      /SORT LIST MUST BE FOLLOWED BY A NEGATIVE
5506
5507      /BRANCH LIST FOR BEGINNING OF INPUT ITEM
5508      0763      0607      BL2,      ITM4      /NEGATIVE
5509      0764      0732      ITM8      /LITERAL
5510      0765      0722      ITM7      /ALPHA CONSTANT
5511      0766      0751      ITM15     /NULL ITEM
5512      0767      0751      ITM15     /NULL ITEM
5513      0770      0610      ITM2      /IGNORE SPACE
5514      0771      0610      ITM2      /IGNORE TAB
5515      0772      0751      ITM15     /NULL ITEM
5516      /BRANCH LIST FOR INITIAL CHAR, OF LITERL
5517      0773      0722      BL3,      ITM7      /GET ASCII VALUE FOR LITERAL
5518      0774      0742      ITM10     /SET NEG, SW,
5519      0775      0744      ITM11     /SET MODE TO DECIMAL
5520      0776      0745      ITM12     /SET MODE TO OCTAL

```

5521  
5522  
5523  
5524  
5525  
5526  
5527  
5528  
5529  
5530  
5531  
5532  
5533  
5534  
5535  
5536  
5537

1000 \*1000

/LTBSE, 2

/COLLECTION PHASE LITERAL TABLE  
/120 WORDS  
/SINGLE WORD ENTRIES  
/CONTAINING ACTUAL VALUES  
/TABLE CONTAINS NOT ONLY  
/LITERALS BUT ALSO  
/POINTERS TO CONSTANT  
/AND ABSOLUTE ADDRESSES.  
/TABLE BUILT ANEW FOR  
/EACH PAGE OF CODE,



5565		1757	*1757	
5566	1757	2215	SL7,	215
5567	1760	2214		214
5568	1761	2212		212
5569	1762	2240		240
5570	1763	2211		211
5571	1764	7777		=1
5572	1765	1657	BL1,	PTEXT /SPACE
5573	1766	1657		PTEXT /TAB
5574	1767	1676		TEXERR /000
5575	1770	2207	BL7,	L72S
5576	1771	2211		L72X
5577	1772	2202		L72*2
5578	1773	2202		L72+2
5579	1774	2211		L72X
5580				
5581				
5582				

/SORT LIST MUST BE FOLLOWED BY A NEGATIVE

```
5583 /MAIN SYMBOL TABLE
5584
5585 2000 *2070
5586
5587
5588 /ENTRIES ARE COMPOSED OF THE FOLLOWING:
5589 / FIRST A 1 WORD HEADER CODE
5590 / THEN THE SYMBOL ITSELF IN PACKED 6BIT ASCII (1*3 WORDS)
5591 / FINALLY THE 1 WORD BINARY VALUE OF THE SYMBOL
5592
5593 /THE HEADER CODE IS LAID OUT AS FOLLOWS:
5594 / (A) FOR OP CODE SYMBOLS:
5595 / BIT0=1 AFTER THE SYMBOL HAS BEEN PRINTED BY PRSYM
5596 / BITS1&2=3 (THESE ARE THE SYMBOL TYPE BITS)
5597 / BIT3=1 FOR MEMORY REFERENCE INSTRUCTIONS
5598 / BITS4&5=THE MICRO-GROUP FOR OPR INSTRUCTIONS (0 FOR MRI AND IOT INSTS.)
5599 / (NOTE! MICRO-GROUP IS SET TO 0 FOR CLA)
5600 / BIT6=1 IF THE SYMBOL IS A PSEUDO-OP
5601 / BIT7=1 IF THE INST. IS A SKIP TYPE INST.
5602 / BIT8=1
5603 / BIT9=0
5604 / BITS10&11=THE NUMBER OF PACKED ASCII SYMBOL WORDS IN THE ENTRY
5605
5606 / (B) FOR OTHER SYMBOL TYPES:
5607 / BIT0 AS ABOVE
5608 / BITS1&2=0 FOR ABSOLUTE AND COMMON SYMBOLS
5609 / =1 FOR RELOCATABLE SYMBOLS
5610 / =2 FOR EXTERNAL SYMBOLS
5611 / BIT3=1 AFTER THE SYMBOL HAS BEEN DEFINED
5612 / BIT4=1 FOR ENTRY SYMBOLS
5613 / BIT5=1 IF THE SYMBOL IS EVER REFERENCED BY A #
5614 / BIT6=1 IF THE SYMBOL IS IN COMMON
5615 / BIT7=1 IF THE SYMBOL IS A DUMMY SYMBOL
5616 / BITS8=11 AS ABOVE
```

5617			/MST#,		
5618	2000	3053		3053	/ABSYM
5619	2021	0102		0102	
5620	2022	2331		2331	
5621	2003	1500		1500	
5622	2004	3000		PABSYM	
5623	2025	3052		3052	/ARG
5624	2006	0122		0122	
5625	2027	0700		0700	
5626	2010	1406		PARG	
5627	2011	3412		3412	/AND
5628	2012	0116		0116	
5629	2013	0400		0400	
5630	2014	0000		AND 0	
5631	2015	3053		3053	/BLOCK
5632	2016	0214		0214	
5633	2017	1703		1703	
5634	2020	1300		1300	
5635	2021	2000		PBSS	
5636	2022	3052		3052	/CALL
5637	2023	0301		0301	
5638	2024	1414		1414	
5639	2025	1400		PCALL	
5640	2026	3053		3053	/COMMN
5641	2027	0317		0317	
5642	2030	1515		1515	
5643	2031	1600		1600	
5644	2032	1600		PCOMMN	
5645	2033	3112		3112	/CIA
5646	2034	0311		0311	
5647	2035	0100		0100	
5648	2036	7041		CIA	
5649	2037	3012		3012	/CLA
5650	2040	0314		0314	
5651	2041	0100		0100	
5652	2042	7200		CLA	
5653	2043	3112		3112	/CLL
5654	2044	0314		0314	
5655	2045	1400		1400	
5656	2046	7100		CLL	
5657	2047	3112		3112	/CMA
5658	2050	0315		0315	
5659	2051	0100		0100	
5660	2052	7040		CMA	
5661	2053	3112		3112	/CML
5662	2054	0315		0315	
5663	2055	1400		1400	
5664	2056	7020		CML	

5665	2057	3053	3053	/DECIM
5666	2060	2405	0405	
5667	2061	0311	0311	
5668	2062	1500	1500	
5669	2063	4735	PDEC	
5670	2064	3053	3053	/DUMMY
5671	2065	0425	0425	
5672	2066	1515	1515	
5673	2067	3100	3100	
5674	2070	5545	PDUMMY	
5675	2071	3412	3412	/DCA
5676	2072	0403	0403	
5677	2073	0100	0100	
5678	2074	3000	DCA 0	
5679	2075	3052	3052	/EAP
5680	2076	0501	0501	
5681	2077	2000	2000	
5682	2100	6375	PEAP	
5683	2101	3052	3052	/END
5684	2102	0516	0516	
5685	2103	0400	0400	
5686	2104	0534	PEND	
5687	2105	3053	3053	/ENTRY
5688	2106	0516	0516	
5689	2107	2422	2422	
5690	2110	3100	3100	
5691	2111	6662	PENTRY	
5692	2112	3053	3053	/FORTR
5693	2113	0617	0617	
5694	2114	2224	2224	
5695	2115	2200	2200	
5696	2116	3363	PFORT	
5697	2117	3212	3212	/HLT
5698	2120	1014	1014	
5699	2121	2400	2400	
5700	2122	7402	HLT	
5701	2123	3051	3051	/IF
5702	2124	1106	1106	
5703	2125	6341	PIF	
5704	2126	3053	3053	/CPAGE
5705	2127	0320	0320	
5706	2130	0107	0107	
5707	2131	0500	0500	
5708	2132	2633	PIFF	
5709	2133	3432	3432	/ISZ
5710	2134	1123	1123	
5711	2135	3200	3200	
5712	2136	2000	ISZ 0	
5713	2137	3412	3412	/INC (NON-SKIP ISZ)
5714	2140	1116	1116	
5715	2141	0300	0300	
5716	2142	2000	ISZ 0	

5717	2143	3112	3112	/IAC
5718	2144	1101	1101	
5719	2145	0300	0300	
5720	2146	7001	IAC	
5721	2147	3012	3012	/IOF
5722	2150	1117	1117	
5723	2151	0600	0600	
5724	2152	6002	IOF	
5725	2153	3012	3012	/ION
5726	2154	1117	1117	
5727	2155	1600	1600	
5728	2156	6001	ION	
5729	2157	3412	3412	/JMP
5730	2160	1215	1215	
5731	2161	2000	2000	
5732	2162	5000	JMP 0	
5733	2163	3412	3412	/JMS
5734	2164	1215	1215	
5735	2165	2300	2300	
5736	2166	4000	JMS 0	
5737	2167	3012	3012	/KRB
5738	2170	1322	1322	
5739	2171	0200	0200	
5740	2172	6036	KRB	
5741	2173	3032	3032	/KSF
5742	2174	1323	1323	
5743	2175	0600	0600	
5744	2176	6031	KSF	
5745	2177	3052	3052	/LAP
5746	2200	1401	1401	
5747	2201	2000	2000	
5748	2202	6372	PLAP	



5749	22.3	3112	3112	/OP
5750	22.4	1617	1617	
5751	22.5	2020	2020	
5752	22.6	7220	7020	
5753	22.7	3053	3053	/NOTAL
5754	2210	1723	1723	
5755	2211	2401	2401	
5756	2212	1420	1420	
5757	2213	4740	POCT	
5758	2214	3053	3053	/OPDEF
5759	2215	1720	1720	
5760	2216	0405	0405	
5761	2217	0600	0600	
5762	2220	3605	OPDEX	
5763	2221	3212	3212	/OSR
5764	2222	1723	1723	
5765	2223	2200	2200	
5766	2224	7404	OSR	
5767	2225	3052	3052	/PAGE
5768	2226	2001	2001	
5769	2227	0705	0705	
5770	2230	0334	PPAGE	
5771	2231	3053	3053	/PAUSE
5772	2232	2001	2001	
5773	2233	2523	2523	
5774	2234	0500	0500	
5775	2235	4316	PPAUSE	

5776	2236	3012	3012	/PLS
5777	2237	2014	2014	
5778	2240	2300	2300	
5779	2241	6026	PLS	
5780	2242	3032	3032	/PSF
5781	2243	2023	2023	
5782	2244	0600	0600	
5783	2245	6021	PSF	
5784	2246	3053	3053	/REORG
5785	2247	2205	2205	
5786	2250	1722	1722	
5787	2251	0700	0700	
5788	2252	0345	PRORG	
5789	2253	3053	3053	/RETRN
5790	2254	2205	2205	
5791	2255	2422	2422	
5792	2256	1600	1600	
5793	2257	4257	PRTN	
5794	2260	3112	3112	/RAL
5795	2261	2201	2201	
5796	2262	1400	1400	
5797	2263	7004	RAL	
5798	2264	3112	3112	/RAR
5799	2265	2201	2201	
5800	2266	2200	2200	
5801	2267	7010	RAR	
5802	2270	3012	3012	/RFC
5803	2271	2206	2206	
5804	2272	0300	0300	
5805	2273	6014	RFC	
5806	2274	3012	3012	/RRB
5807	2275	2222	2222	
5808	2276	0200	0200	
5809	2277	6012	RRB	
5810	2300	3032	3032	/RSF
5811	2301	2223	2223	
5812	2302	0600	0600	
5813	2303	6011	RSF	

5814	2304	3112	3112	/RTL
5815	2305	2224	2224	
5816	2306	1400	1400	
5817	2307	7006	RTL	
5818	2310	3112	3112	/RTR
5819	2311	2224	2224	
5820	2312	2200	2200	
5821	2313	7412	RTR	
5822	2314	3232	3232	/SKP
5823	2315	2313	2313	
5824	2316	2000	2000	
5825	2317	7410	SKP	
5826	2320	3053	3053	/SKPDF
5827	2321	2313	2313	
5828	2322	2004	2004	
5829	2323	2600	0600	
5830	2324	3604	SKPDEX	
5831	2325	3232	3232	/SMA
5832	2326	2315	2315	
5833	2327	0100	0100	
5834	2330	7500	SMA	
5835	2331	3232	3232	/SNA
5836	2332	2316	2316	
5837	2333	0100	0100	
5838	2334	7450	SNA	
5839	2335	3232	3232	/SNL
5840	2336	2316	2316	
5841	2337	1400	1400	
5842	2340	7420	SNL	
5843	2341	3232	3232	/SPA
5844	2342	2320	2320	
5845	2343	0100	0100	
5846	2344	7510	SPA	
5847	2345	3112	3112	/STA
5848	2346	2324	2324	
5849	2347	0100	0100	
5850	2350	7240	STA	
5851	2351	3112	3112	/STL
5852	2352	2324	2324	
5853	2353	1400	1400	
5854	2354	7120	STL	
5855	2355	3232	3232	/SEA
5856	2356	2332	2332	
5857	2357	0100	0100	
5858	2360	7440	SEA	
5859	2361	3232	3232	/SEL
5860	2362	2332	2332	
5861	2363	1400	1400	
5862	2364	7430	SEL	

/SABR.815		PAL10	V135	23-SEP-69	11128	PAGE 97
5863	2365	3232		3232		/SPC=SPA*CLA (USED BY COMPILER)
5864	2366	2320		2320		
5865	2367	0300		0300		
5866	2370	7710		SPA CLA		
5867	2371	3412		3412		/TAD
5868	2372	2401		2401		
5869	2373	0400		0400		
5870	2374	1000		TAD 0		
5871	2375	3052		3052		/TEXT
5872	2376	2405		2405		
5873	2377	3024		3024		
5874	2400	1657		PTEXT		
5875	2401	3012		3012		/TLS
5876	2402	2414		2414		
5877	2403	2300		2300		
5878	2404	6046		TLS		
5879	2405	3032		3032		/TSF
5880	2406	2423		2423		
5881	2407	0600		0600		
5882	2410	6041		TSF		
5883						
5884						
5885		2020	ACH=20			
5886		0021	ACH=21			
5887		0022	ACL=22			
5888						
5889	2411	0452		0452		/ACH
5890	2412	0103		0103		
5891	2413	1000		1000		
5892	2414	0020		ACH		
5893	2415	0452		0452		/ACM
5894	2416	0103		0103		
5895	2417	1500		1500		
5896	2420	0021		ACH		
5897	2421	0452		0452		/ACL
5898	2422	0103		0103		
5899	2423	1400		1400		
5900	2424	0022		ACL		
5901						
5902	2425	0451	II,	0451		/I
5903	2426	1100		1100		
5904	2427	0400		0400		
5905						
5906		2430	STTP=,			

5947  
5948  
5949  
5910  
5911  
5912  
5913  
5914  
5915  
5916  
5917  
5918  
5919  
5920  
5921  
5922  
5923  
5924  
5925  
5926  
5927  
5928  
5929  
5930  
5931  
5932  
5933  
5934  
5935  
5936  
5937  
5938  
5939  
5940  
5941  
5942  
5943

7577 \*CORE1=1

/THE OCCURRENCE TABLE EXTENDS DOWNWARD FROM HERE  
/TOWARD THE MAIN SYMBOL TABLE  
/ & SHARING THE SAME SPACE WITH IT.  
/THIS TABLE IS VARIABLE, BEING COLLAPSED  
/AS MUCH AS POSSIBLE DURING USE. THE ONLY  
/THING LEFT ON IT AT THE END ARE UNDEFINED  
/SYMBOLS.  
/THE OCC. TAB. CONTAINS AN ENTRY FOR EVERY  
/REF. TO AN AS YET UNDF. SYMBOL. EACH  
/TIME A SYMBOL IS DEFINED THE TABLE IS SEARCHED  
/TO SEE IF FORWARD REFERENCES TO IT EXIST,  
/IF SO THEY ARE OUTPUT & THE TABLE  
/CONDENSED.  
/ENTRIES CONSIST OF 2 OR 3 WORDS  
/STRUCTURED AS BELOW:  
/HIGH WORD: LOCATION OF REFERENCE  
/LOW WORD: SYMBOL I.D.  
/OPTIONAL WORD: # FLAG  
/THE LOC. WORD CONTAINS THE PROG. ADDR. WHERE  
/THE VALUE OF THE SYM. MUST BE ASSEMBLED  
/THE # FLAG\*1 IF IT EXISTS. IT WILL  
/EXIST ONLY FOR THOSE ENTRIES WHERE THE  
/SYM. WAS REF'D. BY A #.  
/THE TABLE IS ALWAYS SEARCHED IN REVERSE  
/FROM LOW CORE UPWARD  
/THE O.T. PTR (OTP) ALWAYS PTS. TO THE NEXT FREE  
/LOCATION BELOW THE TABLE  
/THE TABLE HAS NO IMPORTANCE DURING PASS 2.

7600 CORE1=7600

/THE TOP PAGE OF UPPER  
/MEMORY IS NOT USED

```

2020 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2200
2300
2400
2500
2600 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2700 10000000 00000000 00000000 00000000 11111111 11111111 11111111 11111110

1000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1100 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1200
1300
1400
1500
1600 00000000 00000000 00000000 00000000 00000010 00000000 00000000 00000000
1700 00000000 00000000 00000000 00000000 00000000 00000001 11111111 11111000

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2400 11111111 11111111 11111111 00000000 00000000 00000000 00000000 00000000
2500 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2600
2700

3000
3100
3200
3300
3400
3500
3600
3700

```

4000  
4100  
4200  
4300  
4400  
4500  
4600  
4700

5000  
5100  
5200  
5300  
5400  
5500  
5600  
5700

6000  
6100  
6200  
6300  
6400  
6500  
6600  
6700

7000  
7100  
7200  
7300  
7400  
7500  
7600  
7700

A1	5526	ASM09I	5736	BL3	0773	CPGES1	0022
A1P	0442	ASM10	6001	BL6	0745	CPLFS	5310
A2	5424	ASM10B	6053	BL7	1770	CPLFS3	5354
A2NONA	5511	ASM11	6006	BLEN	0137	CP5W	0104
A2P	0441	ASM12	6272	BNKSAP	6606	CRLF	0024
AACTR	0141	ASM12E	6300	BNKSAV	5774	CSUM	0062
AAUM7	2523	ASM12F	6304	BNKSV	1036	CTPTR	2473
ACH	0020	ASM12I	6054	BPTR	0003	CTYPE	0023
ACL	0022	ASM13	6313	BSEEST	0162	CURSKP	0146
ACM	0021	ASM13I	6055	BSSSW	0060	DATA	1100
ACTR	0057	ASM14	6327	BUF	4703	QCIL	2000
ADDRES	0135	ASM2AI	5731	BUFBEQ	6657	QCIL1	0033
AERROR	5760	ASM2IX	5772	BUFEN	6661	DEP1	3626
AFLG	0133	ASM5AI	5614	BUFEND	4704	DEP2	3634
AFS	0142	ASM5CI	5615	BVAL	0002	DEP3	3636
ALEN	0150	ASMBL	5400	BYTE	1771	DEPERR	3660
ANCHK	5737	ASME1	5740	C2	4423	DEFSUB	3642
ANUM7	2514	ASME1I	6117	C3774	5363	DELIN	1771
ANUMCK	2476	ASME1X	5745	C3777	5576	DISPL	0010
APMSW	0102	ASME2	5741	CALERR	1420	DORTN	4312
APTR	0005	ASME2P	6035	CALLFS	0041	DSW	0105
ARGCT	1562	ASME3	6010	CALLSP	0761	DUM	3706
ARGPP0	1527	ASME4	6023	CALLSW	1563	DUMMY	0025
ARGPP2	1544	ASME5	6057	CDPCHG	1570	DUMP	7175
ARGPP4	1535	ASME6	6102	CDFSK	0045	DUMS	0067
ARGPP5	1531	ASME7	6063	CDFSKI	6116	DUMSUB	3550
AS0	0005	ASME7I	5644	CDFSKP	5525	EFLG	0131
AS00I3	6310	ASME8	6114	CDZSK	0051	ENDEND	0570
AS00I4	6260	ASMEXT	3570	CDZSKI	5645	EQSAV	1022
AS00I1	6251	ASMIF	2672	CDZSKP	2577	EQUVB	0000
ASAV	0143	ASMIF1	1552	CERROR	5576	EQUVBT	0101
ASIF	1767	ASMX4	6070	CHARCT	0127	EQVFIX	6725
ASM00I	5730	ASMX4I	5641	CHR	0061	EQVPR	0064
ASM01	5746	ASMX5	6071	CKCLS	0771	EQVOPR	0063
ASM01A	5460	ASMX5I	5640	CKCLSP	1565	ERR1	2773
ASM01B	5477	ASMX6	6076	CKCSW	0762	ERR11	2751
ASM02	5651	ASMX6I	5643	CKCSWP	0333	ERR2	2767
ASM02A	6036	ASR	6637	CKIF	6746	ERRA	2712
ASM02I	5514	ASR1	6643	CKIF2	6757	ERRC	2711
ASM02S	1560	ASR2	6647	CKIF3	6771	ERRE	2721
ASM03I	5732	ASSMBL	0432	CKIFP	0757	ERREX	2756
ASM04I	5733	AT	2771	CLENUP	0531	ERRI	2710
ASM05	5600	ATEM1	3355	CLNPST	6424	ERRM	2705
ASM05C	6231	ATEM2	3356	CODE	0134	ERRS	2702
ASM06	6252	ATEM3	3357	COMMN0	1637	ERRS1	4471
ASM06I	5734	ATEM4	3360	COMMN1	1641	ESTSIZ	0100
ASM07	6200	AVAL	0142	COMMN2	1645	EXP	0145
ASM07A	6220	BANK	0147	COMP	0270	FATAL	2700
ASM07I	5735	BCODE	0001	COMPGO	0264	FATALP	4373
ASM08	5616	BCTR	0140	CORE1	7600	FETCH	4641
ASM09	6241	HL1	1765	COUNT	0152	FIXI	0332
ASM09B	6226	HL2	0763	CPGES	2215	FIXIL	2676



FIXILC	5561	IPOPIN	2672	K2400	2047	L40A	3465
FORFLG	4126	IPSHIN	2667	K2516	6167	L40B	3526
FOUND	0126	ISSI	5423	K2560	6143	L40C	3532
GETBAP	5773	ISZPT	6721	K260	3705	L40D	3477
GETBAS	6640	ISZPT1	0533	K3	0016	L41	3511
GETCHR	0026	ISZPT2	3567	K30	0020	L51	3062
GETSYM	2027	ISZPTX	6455	K300	2167	L51E	3101
GNEG	2524	ITM10	0742	K3000	0031	L51FLG	3172
GTSYM	0600	ITM11	0744	K3010	3641	L51G	3124
HCRPS	3537	ITM12	0745	K3403	5560	L51J	3145
HICOM	0076	ITM13	0747	K377	3441	L52	3237
HISP	6406	ITM14	0750	K3777	4176	L52A	3265
HSR	6610	ITM15	0751	K3777A	4256	L53	3271
HSR1	6613	ITM2	0610	K4	0015	L53A	3315
HSR2	6617	ITM3	0617	K40	0160	L53B	3321
IB	0141	ITM4	0607	K400	0045	L55	0400
IBTI	2163	ITM5	0664	K4000	0041	L55A	0433
ICALSW	0326	ITM6	0714	K4003K	6442	L55B	0434
ICPGES	0022	ITM7	0722	K401	3642	L55C	0405
ICPLFS	0021	ITM8	0732	K403	6700	L55CP	6736
IERROR	5575	ITM9	0733	K407	6056	L55I	0045
IFCT	6775	JASR	6565	K440	1656	L55L	0436
IFCTP	0756	JCOUNT	0125	K4440	2166	L56	0430
IFCTR	6371	JHISP	6571	K5	0154	L61	1323
IFFS	1770	JHSR	6564	K5204	3577	L61A	1335
IFFSUB	2646	JL63	6567	K5376	6311	L61B	1344
II	2425	JL64	6570	K5377	6312	L61C	3367
ILC	0065	JLIST	6572	K60	2576	L61CP	1346
INC	1561	JRDER	6573	K600	0027	L61D	1342
INCOBA	5367	JVERS	7012	K6200	2677	L61DP	3377
INCPT	3551	K0102	6163	K6201	1551	L62	3664
INCPTI	0327	K0317	6164	K7	0155	L62A	3671
INDEV	4702	K0406	6170	K7000	3647	L62A1	3774
INDEVP	6566	K0530	6171	K7000X	7016	L63	3164
INDEX	0151	K1	3776	K7377	3161	L64	4772
INDX1	5515	K10	0156	K7410	3657	L65	4705
INI	0030	K100	0162	K7577X	2475	L66	3713
INILPT	1000	K1000	0030	K7600	0166	L66A	3765
INIS	5516	K130	0017	K77	0161	L66A1	3741
INISS	0530	K1500	6166	K777	4360	L66A2	3747
INISUB	1041	K17	3442	KCDF00	5662	L66A3	3723
INITA	4327	K1720	6144	KCDF10	5660	L66B	2367
INITAI	0574	K176	5523	KCDF1A	5642	L66D	2370
INITAP	0325	K177	0163	KCDFA	6120	L66E	2357
INITIO	0377	K2	0014	KSR	6547	L67	3032
INITMP	4377	K20	0157	L31	2267	L68	2315
INITMR	2371	K200	0164	L31A	2305	L68I	3160
INITR	4574	K2000	0043	L31B	2273	L72	2200
INITRP	4372	K201	0532	L32	2232	L72S	2207
IOI	6463	K2220	6677	L32A	2265	L72X	2211
IOINIT	6456	K2300	6165	L39	3452	L73	4726
IOX	6527	K240	0165	L40	3504	LASMP	0324

LEAD	0575	M260A	0752	OUTSKP	0037	PPARX	1302
LEADER	4714	M3	0172	PARSYM	3600	PPARY	1234
LEADI	4375	M3000	0052	PAG	0077	PPAUSE	4316
LFLG	0130	M331	6563	PARG	1466	PRORG	0345
LFS	0137	M37A	4573	PARG2	1564	PRS1	4010
LFSBSE	0027	M4	3704	PARG5	1552	PRS2	4025
LFSBSI	2474	M5	6575	PARG6	1566	PRS3	4037
LFSBSS	3017	M50	6654	PARLIT	1241	PRS4	4062
LFSCCHK	0031	M7	0004	PASS	0110	PRS5	4101
LFSCK	3000	M7000	0164	PBC	2165	PRS6	4132
LFSCK1	3042	MBE	0170	PBSS	2600	PRS7	4137
LFSPTR	0066	MERROR	5763	PBSS1	2621	PRS8	4171
LINAX	0170	MGRP	0143	PBSS2	2435	PRSYM	4000
LINBUF	1646	MM1	5132	PBSS2I	2630	PRSYM P	0041
LINE	0067	MSCTR	0153	PBSS2J	6723	PRTN	4257
LINEB	1164	MST	0043	PBSS4	6710	PRTNO	4313
LINEB2	4763	MTYPE	7013	PBSS4I	2632	PRTN1	4314
LINEND	4640	MUL1	0702	PBSS5	2610	PRTN3	4315
LINK	0033	NSCHK	6333	PBSS5I	6724	PSHIN2	4213
LISTON	6400	NSCHKI	1322	PCALL	1400	PSHINI	0331
LITBSE	0045	NSCHKP	5647	PCALL1	1570	PST	4175
LITPTR	3362	NSGN	0144	PCOMMN	1600	PSTB	6776
LITSIZ	0070	NULL	4764	PDEC	4735	PSTBSE	0173
LITSZE	0070	NULLP	0034	PDMY	5545	PSTCPR	0114
LLFS	5366	NUMSGN	5173	PEAP	6375	PSTD	2522
LLFSI	2770	NUMSGP	5364	PEB	4376	PSTDEF	5570
LLFSP	2377	OAPLT	3200	PEBSE	0700	PSTDEF	5365
LSTBNK	0073	OACTR	0074	PEND	0534	PSTMAX	2314
LSTDEP	4177	OBFLG	5134	PENTRY	6662	PSTSPR	0115
LSTDEV	6531	OBFLGP	5277	PEPTR	5522	PSTSE	0116
LSTSKK	1037	OBIS	0055	PEPTRP	4374	PTR	7176
LSTSKP	0072	OBISUB	5777	PFLG	0125	PTBSE	0174
LTBSE	0030	OBNSYM	4507	PFORT	3363	PTCPR	0117
LTSZE	0071	OBSYM	0032	PGEESC	0111	PTEXT	1657
M10	1567	OLDOBA	5276	PHASE	0113	PTOPR	0120
M100	3775	OLDOBP	5163	PIF	6341	PTOPTB	0164
M11	6576	OP	0140	PIFERR	6397	PTSIZ	0122
M12	4572	OPDEX	3605	PIFF	2633	PTSPR	0121
M12A	4255	OPICTR	2532	PLAP	6372	PTSZE	0122
M14	6774	OPIS	0062	POCT	4740	PUNCH	0042
M17	6577	OPISUB	5776	POP	1766	PUPGE	0112
M177	4655	OPSCTR	0075	POPEX	6443	PUSH	1765
M2	0171	OTP	0006	POPEXP	0040	PUSHER	4237
M200	0166	OTPR	4356	POPIN	4217	PUSHIN	4200
M201	2675	OTYPE	0035	POPINI	0330	QUERY	6531
M211	1040	OUAPLT	5517	PPAGE	0334	R	4656
M2402A	6271	OUT1	0141	PPAR1	1200	R1	4664
M243	2171	OUT2	0127	PPAR3	1236	RC	0754
M254	0167	OUTBIN	0036	PPAR3S	1253	RCH	4546
M257	6773	OUTBN	3400	PPAR4	1304	RCH3	4565
M26	7014	OUTEX	3430	PPAR5	1316	RCH4	4567
M260	4571	OUTSK	6261	PPAR6	1276	RDER	6417

RDIL	0243	RSTR5	2242	SETSX	5160	TEM10	6655
RDL1	2001	RSTR6	0316	SIGN	0755	TEM11	6656
RDL10	2066	RSTR7	2673	SK	2146	TEM12	2631
RDL11	2172	RSTRX	0375	SKIPL	0046	TEM2	0124
RDL12	2075	RTN	0040	SKPDEX	3604	TEM3	0125
RDL13	2117	RUSVL	4535	SKPSAP	6607	TEM4	0126
RDL14	2125	S0	0150	SKPSAV	5775	TEM5	0127
RDL15	2141	S1	0151	SL1	0757	TEM55	0435
RDL16	2143	S2	0152	SL2	0752	TEM7	0103
RDL17	2161	S3	0153	SL3	0742	TEMP6	0145
RDL18	2174	SAVEIT	4762	SL6	0755	TEST	0053
RDL1X	1035	SAVLIN	4743	SL7	1757	TEX1	1674
PDL2	2022	SAVLNI	3361	SLITAB	0047	TEX2	1704
RDL3	2020	SAVLNP	5520	SLITB1	2416	TEX4	1742
RDL4	2105	SCOLON	0107	SLITB2	2423	TEX5	1726
RDL5	2024	SERALI	1240	SLTAB	2400	TEXCTR	0151
RDL7	2027	SERALP	5646	SME1X	5544	TEXERR	1676
RDL9	2046	SERALY	6270	SPPLUS	2772	TEXP	4233
RDS1	0665	SERROR	5577	SPSTAB	0050	TEXPTR	0152
RECRET	0447	SET00A	5263	SPSTB	2226	TEXSUB	1791
RECT	0443	SET00B	5122	SPTR	0127	TIB	4227
RECT1	0400	SET00I	5115	SRALT	3443	TLFS	4225
RECT2	0454	SET00J	5121	SREST	0051	TNSCN	4232
RECT3	0514	SET01A	5111	SRS	0753	TOP	4226
RECTI	0044	SET01B	5104	SRS1	4403	TOPCOR	3162
REDUCE	5524	SET02I	5116	SRS2	4450	TS2	4236
REDUCP	3536	SET04I	5117	SRS3	4431	TSCHR	1347
REE	0573	SET05I	5120	SRS4	4442	TSCHR2	1354
REEASM	6016	SET06I	5123	SRS5	4444	TSCHR3	1375
FELINE	4754	SET08I	5124	SRSYM	4400	TSK	4234
RELNP	5521	SET09I	5125	START	0200	TSUM	0144
REXIT	6622	SET10I	5126	STCE	0052	TUMIC	4231
RG	4667	SET11I	5127	STFT	6121	TXSV	0153
RG1	4671	SET12I	5130	STFT0	6140	TYPE	0004
RG3	4677	SET6P1	5131	STFT1	6141	UDPAGE	3172
RG3P	6660	SETC00	5251	STFT2	6145	UDPG	0376
RG0	6476	SETC01	5074	STFT3	6147	UMIC	0143
RLN	4600	SETC02	5306	STFT4	6157	UPDATE	2674
PLN15	4625	SETC04	5300	STFT5	6172	USE	0001
RLN2	4607	SETC05	5200	STFTI	4174	VAL	0002
RLN3	4614	SETC06	5231	STORE	5164	VALPTP	2376
RLN4	4634	SETC07	5050	STOREP	4637	VALPTR	4545
RLNP	2164	SETC08	5217	STT	0007	VALUF	0136
PORG1	0374	SETC09	5221	STTP	2430	VERNUM	7000
PORGX	0370	SETC10	5212	STTR	4397	VERSI	6574
RSM1	0651	SETC11	5220	SUM	3432	VERSN	7017
RSM2	0626	SETC12	5224	SUSVL	4520	VFLG	0132
RSM3	0642	SETC13	5236	SYMBOL	0003	VN	6462
RSTRT	0205	SETCAL	1165	T8	2226	VNOP	7015
RSTRT1	0207	SETCAP	5133	TAFS	4230	WHATPP	2170
RSTRT2	0252	SETCT	5000	TBANK	4235	WHATYP	6737
RSTRT4	0245	SETSUB	5134	TEM1	4123	WLIF	4771

/SABR,815

PAL10 V135

23-SEP-69

11120 PAGE 98-7

WLN	1060
WLN1	1125
WLN2	1143
WLN3	1153
WLNIF	2671
WLNIF1	1772
WLNIF1	4326
WLNIP	0055
WRITE	2532
WRITE2	2567
WRITEP	0056
X0	0010
X1	0011
X2	0012
X3	0013

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 61 SECONDS

6K CORE USED

















L32	1990	2033						
L32A	2006	2014	2017					
L39	2830	2884						
L40	2829	2863						
L40A	2846	2880						
L40B	2873	2881						
L40C	2849	2885						
L40D	2856	2888						
L41	2868	2883						
L51	2526	2543						
L51E	2541	2577	2593					
L51FLG	2526	2532	2557	2570	2626			
L51G	2566							
L51J	2583	2592						
L52	2664	2687						
L52A	2686	2709	2713					
L53	2678	2690						
L53A	2695	2710						
L53B	1395	2712	2717	2744				
L55	449	769	792	804	890	910		
L55A	774	796						
L55B	773	797						
L55C	774	5208						
L55CP	5271	5207	5208					
L55I	449	547	568	668	698	720	2361	
L55L	772	802						
L56	775	777	784	786	793			
L61	431	1405	1414					
L61A	1411	1413	1416	1422				
L61B	1419	1423						
L61C	1425	2761						
L61CP	1420	1425						
L61D	1421	1424	2769					
L61DP	2768	2769						
L62	441	3019	3034					
L62A	3024	3033						
L62A1	3127	3110						
L63	446	2605	2610	5073				
L64	456	2104	3817	3822	5074			
L65	434	3726	3732					
L66	453	3058	3098	3109				
L66A	3079	3103						
L66A1	3065	3080						
L66A2	3086	3095						
L66A3	3066	3096						
L66B	2061	2103						
L66D	2088	2104						
L66E	2095	2102	2103					
L67	2489	2495						
L68	2057	2060	2090	2596	3110			
L68I	2525	2596						
L72	450	1946	1958	5577	5578			
L72S	1953	5575						



















TYPE	456	1213	1223	1239	1243	1421	2062	2069	2424	3031	3134	3754	3756	4999
UDPAGE	5024	5062												
UDPG	729	2371	2619	2624	2626									
UDPG	669	699	729											
UMIC	523	617	955	1277	1402	1928	1911	3257						
UPDATE	2362	2371												
USE	47	1367	1371	1560	1639	1642	1834	1838	1842	1876	1880	1899	2070	2490
	2495	2528	2511	2692	2974	3275	3078	3229	3231	3245	3264	3570	3571	3583
	3926	3930	3951	3955	3959	4430	4433	4555	4559	4657	4661	4665	4742	4914
	5159	5162	5218											
VAL	478	1376	1656	1659	1666	1845	1894	1897	1926	2009	2507	2556	2703	2976
	2977	3263	3577	3585	4493	4098	4107	4186	4481	4785	4796	4814	4819	4844
	4852	4876												
VALPTP	2111	2115												
VALPTR	2115	3575	3576	3586	3590									
VALUE	516	1210	2254	2255	2256	3807	3808							
VERNUM	5079	5341												
VERSI	5000	5079												
VERSN	5351	5356												
VFLG	512	1204	1250	2257	3667	3809								
VN	5000	5354												
VNOP	5348	5354												
WHATPP	1821	1832	1861	1874	1916									
WHATYP	1916	5217	5222	5223										
WLIF	3810	3812												
WLN	457	1190	1257											
WLN1	1227	1240	1248											
WLN2	1236	1241	1247											
WLN3	1193	1220	1231	1249										
WLNIF	2328	2368												
WLNIF1	1771	1775	2368	3397	3812									
WLNIFI	3386	3397												
WLNP	457	926	1774	2281										
WRITE	458	2248	2251	2252	2253	2258	2259	2282	2283	2285				
WRITE2	2263	2277												
WRITEP	458	1326	1370	2682	2705	4341	4346	4361	4372	4577	4641	4646	4681	4687
	4833													
X0	416	965	966	1485	1695	1721	1749	1753	1805	1826	1953	3670	3728	4908
X1	417	1225	1228	1446	1449	1457	2067	2077	2145	2147	2728	2731	2733	2867
	2869	2874	2881	3085	3088	3141	3143	3168	3178	3198	3199	3200	3201	3203
	3210	3235	3238	3337	3341	3489	3496	3525	3530	5003	5008	5018	5031	5044
	5343													
X2	418	990	1001	1016	2069	2075	2081	2084	2644	3160	3181	3196	3208	3340
	3342	3491	3493	3523	3532	3535	3538	3648	3655	3660	3661	4066		
X3	419	3634	3700	3704	3708	3710	3711	3716	5113	5120	5121			