

WVS*VORTEX(1),WCSCOM

1	*THIS IS A	COPYRIGHTED PROGRAM,COPYRIGHT 1975 BY VARIAN DATA MACHINE	00	00001
2	*		00	00002
3	*	THE FOLLOWING ARE SUPPLEMENTAL OPCODES	00	00003
4	*	FOR USE WITH THE MICRO ASSEMBLER	00	00004
5	*		00	00005
6	*		00	00006
7	ADD	EQU 9	00	00007
8	ALUC	EQU 8	00	00008
9	ALUO	EQU 6	00	00009
10	ALUS	EQU 7	00	00010
11	ALUZ	EQU 9	00	00011
12	AND	EQU X'B	00	00012
13	AONES	EQU 2	00	00013
14	AZERO	EQU 1	00	00014
15	ASGPR	EQU 0	00	00015
16	ASGPRL	EQU 2	00	00016
17	ASGPRR	EQU 3	00	00017
18	ASP	EQU 1	00	00018
19	ASSPEC	EQU 0	00	00019
20	BSSALU	EQU 7	00	00020
21	BSSMIR	EQU X'F	00	00021
22	BSSOVR	EQU 3	00	00022
23	BSSP	EQU X'B	00	00023
24	B\$GPR	EQU 0	00	00024
25	B\$SPEC	EQU 1	00	00025
26	CRY1	EQU 3	00	00026
27	DECA	EQU X'F	00	00027
28	DECB	EQU 9	00	00028
29	DECODE	EQU 4	00	00029
30	DECOD\$	EQU 5	00	00030
31	EOR	EQU 6	00	00031
32	FT	EQU 3	00	00032
33	GPROUT	EQU 1	00	00033
34	GPRS	EQU X'D	00	00034
35	IBR\$I	EQU 1	00	00035
36	IFSALU	EQU 4	00	00036
37	IFSMIR	EQU X'C	00	00037
38	IFSOVR	EQU 0	00	00038
39	IFSP	EQU 8	00	00039
40	INCA	EQU 0	00	00040
41	INCB	EQU 1	00	00041
42	INCP	EQU 4	00	00042
43	INCSC	EQU 5	00	00043
44	IOR	EQU 2	00	00044
45	IOSR	EQU 1	00	00045
46	KOUT	EQU 6	00	00046
47	LFT	EQU 0	00	00047
48	LIT	EQU 3	00	00048
49	LOG	EQU 1	00	00049
50	MEMC	EQU 1	00	00050
51	MEMC\$	EQU 2	00	00051
52	MIR	EQU 1	00	00052
53	MIRS	EQU X'B	00	00053
54	MSK	EQU 2	00	00054
55	NORM	EQU X'E	00	00055
56	NOTA	EQU 0	00	00056

57	NOTB	EQU	5
58	OF\$ALU	EQU	5
59	OF\$MIR	EQU	X'D
60	OF\$OVR	EQU	1
61	OF\$P	EQU	9
62	ORLZ	EQU	7
63	OLSE	EQU	5
64	ONES	EQU	3
65	OPR	EQU	0
66	OPROUT	EQU	3
67	OR	EQU	1
68	ORSE	EQU	4
69	ORZF	EQU	6
70	OS\$ALU	EQU	6
71	OS\$MIR	EQU	X'E
72	OS\$OVR	EQU	2
73	OS\$P	EQU	X'A
74	OVFL	EQU	0
75	PJMP	EQU	3
76	PJMPS	EQU	4
77	POUT	EQU	1
78	POPDEL	EQU	1
79	POPJMP	EQU	4
80	PUSH	EQU	2
81	QUOS	EQU	X'F
82	R0	EQU	0
83	R1	EQU	1
84	R2	EQU	2
85	R3	EQU	3
86	R4	EQU	4
87	R5	EQU	5
88	R6	EQU	6
89	R7	EQU	7
90	R8	EQU	8
91	R9	EQU	9
92	RA	EQU	X'A
93	RB	EQU	X'B
94	RC	EQU	X'C
95	RD	EQU	X'D
96	RE	EQU	X'E
97	RF	EQU	X'F
98	RGHT	EQU	1
99	ROVFL	EQU	4
100	SCOUT	EQU	2
101	SFTC	EQU	X'C
102	SHFA	EQU	X'C
103	SHFT	EQU	X'A
104	SHFTOP	EQU	1
105	SOVFL	EQU	2
106	SPEC	EQU	0
107	SSW1	EQU	4
108	SSW2	EQU	3
109	SSW3	EQU	2
110	STACK	EQU	X'D
111	STAT	EQU	3
112	SUB	EQU	6
113	S\$ALU	EQU	2

00	00057
00	00058
00	00059
00	00060
00	00061
00	00062
00	00063
00	00064
00	00065
00	00066
00	00067
00	00068
00	00069
00	00070
00	00071
00	00072
00	00073
00	00074
00	00075
00	00076
00	00077
00	00078
00	00079
00	00080
00	00081
00	00082
00	00083
00	00084
00	00085
00	00086
00	00087
00	00088
00	00089
00	00090
00	00091
00	00092
00	00093
00	00094
00	00095
00	00096
00	00097
00	00098
00	00099
00	00100
00	00101
00	00102
00	00103
00	00104
00	00105
00	00106
00	00107
00	00108
00	00109
00	00110
00	00111
00	00112
00	00113

```
114 S$OVFL EQU 6 00 00114
115 S$OVFS EQU 8 00 00115
116 S$SHFT EQU 1 00 00116
117 TCB EQU 2 00 00117
118 TESTT EQU 3 00 00118
119 TESTF EQU 2 00 00119
120 TFIR EQU 5 00 00120
121 TRNA EQU X'F 00 00121
122 TRNB EQU X'A 00 00122
123 TT EQU 2 00 00123
124 WAITMD EQU 1 00 00124
125 ZERO EQU 3 00 00125
126 EJEC 00 00126
127 * 00 00127
128 * FOLLOWING ARE ROM STANDARD STATE ADDRESSES 00 00128
129 * 00 00129
130 SS1M EQU X'13E RESTART PIPELINE @ P 00 00130
131 SS2M EQU X'092 MAINTAIN PIPELINE 00 00131
132 SS3M EQU X'02D DECODE NEXT INSTRUCTION (IN IBR) 00 00132
133 INT1 EQU X'D1 ROM INTERRUPT PROCESSOR 00 00133
134 EJEC 00 00134
135 * REMOVE ONE OF FOLLOWING TO SELECT PAGE NUMBER 00 00135
136 P EQU X'200 PAGE 1 00 00136
137 P EQU X'400 PAGE 2 00 00137
138 ALOC 0 00 00138
139 ***** 00 00139
140 * BCS ENTRY POINTS * 00 00140
141 ***** 00 00141
142 ***** 00 00142
143 ***** LOAD SAVE P IN RA* 00 00143
144 ***** 00 00144
145 * STORE REGISTER ENTRY 00 00145
146 ORG X'F 00 00146
147 RST010 GEN /N(RLS010),12(ASP),14(TRNA),15(LOG),17(GPROUT), 00 00147
148 C24(RA) 00 00148
149 ***** 00 00149
150 ***** LOAD SET MEM START ADDR* 00 00150
151 ***** 00 00151
152 * LOAD REGISTER ENTRY 00 00152
153 ORG X'10 00 00153
154 RLD010 GEN /N(RST010),11(B$GPR),23(R7),14(INCB),16(CRY1), 00 00154
155 C17(GPROUT),24(RB),12(AS$PEC),22(AZERO) 00 00155
156 * 00 00156
157 * DECODE ENTRY 00 00157
158 * FETCH NEXT OP/CODE THROUGH P$EUDO=P 00 00158
159 * 00 00159
160 ORG X'11 00 00160
161 DEC00 GEN /N(DEC0),12(AS$GPR),24(R2),14(TRNA), 00 00161
162 C15(LOG),10(OF$ALU),6(MEMC) 00 00162
163 * 00 00163
164 * DECIMAL MATH ENTRY 00 00164
165 * START SAVE OF P IN TEMP BLOCK 00 00165
166 * 00 00166
167 ORG X'12 00 00167
168 DE00 GMSK /N(DE01),6(MEMC),10(OS$ALU),12(AS$GPR),16(RO), 00 00168
169 C11(LIT),15(X'8000),14(AND),13(OPROUT) 00 00169
170 ***** 00 00170
```

```
171 ***** SAVE P IN RA AND SELECT OP 00 00171
172 ***** 00 00172
173 ORG X'13 00 00173
174 MCMOVE GEN /F(T100),FSB,2(1),12(ASP),14(TRNA),17(GPROUT),24(RA), 00 00174
175 C15(LOG) 00 00175
176 *COMPOP 00 00176
177 *COMPOP 00 00177
178 *COMPOP 00 00178
179 ORG X'2A 00 00179
180 COMPOP GEN IMD,LB3,AA4 00 00180
181 EJEC 00 00181
182 ORG X'60 00 00182
183 ***** 00 00183
184 * 00 00184
185 * TITLE: LOAD/STORE REGISTER SUBROUTINE * 00 00185
186 * 00 00186
187 * PURPOSE: LOAD OR STORE REGISTER BLOCK SPECIFIED BY * 00 00187
188 * P= MEM ADDR OF FIRST REG * 00 00188
189 * SC=COMPLIMENT(REG COUNT+1) * 00 00189
190 * CALL ROUTINES AS FOLLOWS: * 00 00190
191 * LDR0 = LOAD R0 * 00 00191
192 * LDRD = LOAD RD * 00 00192
193 * LDR4 = LOAD R4 * 00 00193
194 * STR0 = STORE R0 * 00 00194
195 * STRD = STORE RD * 00 00195
196 ***** 00 00196
197 * CAUTION!!! R8 IS MODIFIED! * 00 00197
198 ***** 00 00198
199 * 00 00199
200 * NOTE SAMPLES LAST REGISTER LOADED * 00 00200
201 ***** 00 00201
202 ***** 00 00202
203 ***** LDR0 SET FIRST REG (R0)* 00 00203
204 ***** AND LOAD FLAG * 00 00204
205 ***** 00 00205
206 LDR0 GEN /N(ST010),12(ASPEC),22(AZERO),14(DECA), 00 00206
207 C13(OPROUT) 00 00207
208 ***** 00 00208
209 ***** LDR4 SET FIRST REG (R4)* 00 00209
210 ***** AND LOAD FLAG * 00 00210
211 ***** 00 00211
212 LDR4 GMSK /N(LS010),11(LIT),15(X'FFFC),14(TRNB),13(OPROUT), 00 00212
213 C7(SSALU) 00 00213
214 ***** 00 00214
215 ***** LDRD SET FIRST REG (RD)* 00 00215
216 ***** AND LOAD FLAG * 00 00216
217 ***** 00 00217
218 LDRD GMSK /N(LS010),11(LIT),15(X'FFF3),14(TRNB),13(OPROUT), 00 00218
219 C7(SSALU) 00 00219
220 ***** 00 00220
221 ***** STR0 SET FIRST REG (R0)* 00 00221
222 ***** 00 00222
223 STR0 GEN /N(LS010),12(ASPEC),22(AZERO),14(DECA), 00 00223
224 C13(OPROUT),7(SSALU) 00 00224
225 ***** 00 00225
226 ***** STR4 SET FIRST REG (R4) 00 00226
227 ***** 00 00227
```

```
228 STR4 GMSK /N(STO10),11(LIT),15(X'FFFC'),14(TRNB),13(OPROUT) 00 00228
229 ***** 00 00229
230 ***** STRC SET FIRST REG RC 00 00230
231 ***** 00 00231
232 ORG X'DB 00 00232
233 STRC GMSK /N(STO10),11(LIT),15(X'FFF4'),14(TRNB),13(OPROUT) 00 00233
234 ORG STR4+1 00 00234
235 ***** 00 00235
236 ***** STRD SET FIRST REG (RD)* 00 00236
237 ***** 00 00237
238 STRD GMSK /N(STO10),11(LIT),15(X'FFF3'),14(TRNB),13(OPROUT) 00 00238
239 ***** 00 00239
240 ***** STORE SET STORE FLAG* 00 00240
241 ***** 00 00241
242 STO10 GEN /N(LSO10),11(BSSPEC),23(OPR),14(NOTB),15(LOG), 00 00242
243 C7(SSALU) 00 00243
244 ***** 00 00244
245 ***** COMBINE I AND REG # 00 00245
246 ***** MOVE TO IBR 00 00246
247 ***** 00 00247
248 LSCONT EQU * 00 00248
249 ORG X'26 00 00249
250 LSO10 GEN /N(LSO15),11(BSSPEC),23(OPR),12(ASPEC),22(AZERO), 00 00250
251 C14(INCB),16(CRY1),17(GPROUT),24(RB) 00 00251
252 ORG LSCONT 00 00252
253 LSO15 GMSK /N(LSO20),12(ASGPR),16(RB),11(MSK),15(X'F'),14(OR), 00 00253
254 CSFO,1M4 00 00254
255 LSO20 GMSK /N(LSO22),12(ASGPR),16(RB),11(MSK),15(X'F'),14(OR) 00 00255
256 ***** 00 00256
257 ***** IBR TO I 00 00257
258 ***** 00 00258
259 LSO22 GEN /N(LSO24),TF0,SFO,7(IBRS1) 00 00259
260 ***** 00 00260
261 ***** SELECT LOAD/STORE 00 00261
262 ***** IF LOAD, START FETCH OF NEXT REG 00 00262
263 ***** 00 00263
264 LSO24 GEN /T(LDO20,STO20),5(FT),7(ALUS),6(TESTF),10(OFSP) 00 00264
265 ***** 00 00265
266 ***** INCREMENT TO NEXT REGISTER* 00 00266
267 ***** TEST FOR COUNT SATISFIED * 00 00267
268 ***** 00 00268
269 LSO30 GMSK /T(COMPPOP,LSO10),7(SFTC),5(TT),11(MSK), 00 00269
270 C15(X'FFFO'),14(TRNB),RF7 00 00270
271 ***** 00 00271
272 ***** LOAD WAIT FOR FETCH AND* 00 00272
273 ***** SELECT REGISTER * 00 00273
274 ***** 00 00274
275 LDO20 GEN /N(LDO40),6(SPEC),10(WAITMD),AB2,TSO,MR1,23(MIR) 00 00275
276 ***** 00 00276
277 ***** LOAD TRANSFER MIR TO SELECTED * 00 00277
278 ***** REG AND INCREMENT REG COUNT * 00 00278
279 ***** 00 00279
280 LDO40 GEN /N(LSO30),11(BSSPEC),23(MIR),14(TRNB),15(LOG), 00 00280
281 C17(GPROUT),24(RF),13(INCSC) 00 00281
282 ***** 00 00282
283 ***** STORE SELECT STORE REG* 00 00283
284 ***** 00 00284
```

```

285 ST020 GMSK /N(ST030),SFO,AB2,TSO,MR1 00 00285
286 ***** 00 00286
287 ***** STORE MOVE REG CONTENTS TO OPR 00 00287
288 ***** BEGIN REG STORE * 00 00288
289 ***** 00 00289
290 ST030 GEN /N(ST040),12(ASGPR),24(RF),14(TRNA),15(LOG), 00 00290
291 C13(OPROUT),6(MEMC),10(OS$P) 00 00291
292 ***** 00 00292
293 ***** STORE COMPLETE STORE* 00 00293
294 ***** 00 00294
295 ST040 GEN /N(LS030),11(B$SPEC),23(OPR),14(TRNB),15(LOG), 00 00295
296 C6(SPEC),10(WAITMD),13(INCSC) 00 00296
297 EJEC 00 00297
298 ***** 00 00298
299 * 00 00299
300 * 00 00300
301 * TITLE: REGISTER LOAD/STORE * 00 00301
302 * * 00 00302
303 * PURPOSE: LOAD/STORE V75 REGISTERS ON A STACK. R7 CONTAINS * 00 00303
304 * STACK POINTER. 1 TO 8 REGISTERS BEGINNING WITH * 00 00304
305 * RO MAY BE LOADED OR STORED. * 00 00305
306 * * 00 00306
307 * REGISTER USAGE: * 00 00307
308 * R7 STACK POINTER: LOAD=RO-1 ADDR * 00 00308
309 * STORE=ADDR TO STORE HIGHEST REG * 00 00309
310 * INTERNAL * 00 00310
311 * RF - SAVE P * 00 00311
312 * P - PASS STACK POINTER TO LDRO OR STRO * 00 00312
313 * OPR- CALCULATE NUMBER OF REGISTERS * 00 00313
314 * SC - PASS NUMBER OF REGISTERS * 00 00314
315 * * 00 00315
316 * EXIT LOAD R7 POINTS TO LAST REG LOADED * 00 00316
317 * STORE R7 POINTS ADDR OF RO - 1 * 00 00317
318 * * 00 00318
319 * BCS VALUES: * 00 00319
320 * 0105020 LOAD RO * 00 00320
321 * 0105060 LOAD RO,R1 * 00 00321
322 * 0105120 LOAD RO,....,R2 * 00 00322
323 * 0105160 LOAD RO,....,R3 * 00 00323
324 * 0105220 LOAD RO,....,R4 * 00 00324
325 * 0105260 LOAD RO,....,R5 * 00 00325
326 * 0105320 LOAD RO,....,R6 * 00 00326
327 * 0105360 LOAD RO,....,R7 * 00 00327
328 * * 00 00328
329 * 0105017 STORE RO * 00 00329
330 * 0105057 STORE RO,R1 * 00 00330
331 * 0105117 STORE RO,....,R2 * 00 00331
332 * 0105157 STORE RO,....,R3 * 00 00332
333 * 0105217 STORE RO,....,R4 * 00 00333
334 * 0105257 STORE RO,....,R5 * 00 00334
335 * 0105317 STORE RO,....,R6 * 00 00335
336 * 0105357 STORE RO,....,R7 * 00 00336
337 * * 00 00337
338 * CALLED ROUTINES: * 00 00338
339 * LDRO - LOAD REG; P=ADDR SC=NUMBER TO LOAD (COMPLIMENT) * 00 00339
340 * STRO - STORE REG; P=ADDR SC=NUMBER TO STORE (COMPLIMENT) * 00 00340
341 * * 00 00341

```

```
342 ***** 00 00342
343 ***** 00 00343
344 ***** MASK REGISTER COUNT FROM* 00 00344
345 ***** INSTRUCTION * 00 00345
346 ***** 00 00346
347 RLS010 GMSK /N(RLS020),11(MSK),15(X'FF1F),14(TRNB), 00 00347
348 C13(OPROUT) 00 00348
349 ***** 00 00349
350 ***** SET SHIFT AMOUNT FOR FETCHING* 00 00350
351 ***** REGISTER COUNT * 00 00351
352 ***** 00 00352
353 RLS020 GMSK /N(RLS030),11(LIT),15(X'0002),14(TRNB), 00 00353
354 C13(SCOUT) 00 00354
355 ***** 00 00355
356 ***** POSITION OPR FOR REG COUNT* 00 00356
357 ***** 00 00357
358 RLS030 GEN /T(RLS030,RLS040),5(FT),7(SFTC),18(SHFTOP), 00 00358
359 C20(LFT),13(INCSC) 00 00359
360 ***** 00 00360
361 ***** SET REG COUNT TO REG COUNT +1 00 00361
362 ***** 00 00362
363 RLS040 GEN /N(RLS045),11(BSSPEC),23(OLSE),14(INCB),16(CRY1), 00 00363
364 C13(OPROUT),12(ASPEC),22(AZERO) 00 00364
365 ***** 00 00365
366 ***** COMPLIMENT REG COUNT TO SC* 00 00366
367 ***** 00 00367
368 RLS045 GEN /F(RLD020),2(X'1),F57,11(BSSPEC),23(OPR), 00 00368
369 C14(NOTB),15(LOG),13(SCOUT) 00 00369
370 ***** 00 00370
371 ***** LOAD CALL LOAD ROUTINE* 00 00371
372 ***** 00 00372
373 RLD020 GMSK /P(LDR0+P),IMD,LB3,15(RLD040+P),AKB,12(ASGPR), 00 00373
374 C14(TRNA),13(POUT) 00 00374
375 ***** 00 00375
376 ***** STORE SET STORE START ADDR* 00 00376
377 ***** 00 00377
378 RST020 GEN /N(RST030),12(ASGPR),24(R7),11(BSSPEC),23(OPR), 00 00378
379 C14(SUB),16(CRY1),13(POUT),17(GPROUT) 00 00379
380 RLD040 EQU *+1 00 00380
381 ORG RLD040 00 00381
382 ***** 00 00382
383 ***** LOAD SET NEW STACK POINTER* 00 00383
384 ***** 00 00384
385 RLD040 GEN /N(RLSXIT),12(ASP),14(DETA),17(GPROUT),24(R7) 00 00385
386 ***** 00 00386
387 ***** STORE CALL STORE ROUTINE* 00 00387
388 ***** 00 00388
389 RST030 GMSK /P(STRU+P),IMD,LB3,15(RLSXIT+P),AK2,13(INCP) 00 00389
390 ***** 00 00390
391 ***** EXIT EXIT LOAD/STORE* 00 00391
392 ***** 00 00392
393 DB00 EQU *+1 00 00393
394 ORG X'E6 00 00394
395 RLSXIT GEN /P(SS2M),6(MEMCS),7(PJMPs),TFO,10(IFSALU), 00 00395
396 C12(ASGPR),24(RA),14(TRNA),15(LOG),13(POUT) 00 00396
397 ORG DB00 00 00397
398 EJEC 00 00398
```



```
399 *****
400 *
401 * TITLE:      COMPUTE MEMORY ACC BYTE ADDR      (DB00)
402 *
403 * PURPOSE:    CONVERT PARAMETER OF BASE(4MSB)/DISPLACEMENT(12LSB),
404 *             BYTE ADDR.
405 *
406 * REGISTER USAGE:
407 *             ENTRY
408 *             MIR = BASE/DISPL BEING FETCHED
409 *             R1  = ADDR OF CONDIT
410 *             EXIT
411 *             P   = BYTE ADDR
412 *
413 *             INTERNAL
414 *             SC  = USED FOR BASE ISOLATION IN OPR
415 *             OPR = ISOLATE BASE
416 *             R8  = ISOLATE DISPLACEMENT
417 *             MIR = FETCH FWA OF MEM ACC
418 *
419 * NOTE: IF INVALID MEM ACC FWA (0 OR NEGATIVE) CONDIT (R1) IS
420 *       SET TO 5 AND OVERFLOW IS SET.
421 *****
422 *****
423 ***** BASE/DISPL BEING FETCHED. *
424 ***** INIT SHIFT FOR BASE ISOLATE*
425 *****
426 DB00  GMSK  /N(DB10),6(SPEC),10(WAITMD),11(LIT),15(X'3),
427          C14(TRNB),13(SCOUT)
428 *****
429 ***** BASE/DISPL TO R8*
430 *****
431 DB10  GEN   /N(DB30),11(B$SPEC),23(MIR),14(TRNB),15(LOG),
432          C17(GPROUT),24(R8),13(OPROUT)
433 *****
434 ***** FETCH BASE WORD ADDR*
435 *****
436 DB40  GEN   /N(DB50),12(A$GPR),24(R1),14(ADD),16(CRY1),
437          C11(B$SPEC),23(OLSE),6(MEMC),10(OF$ALU)
438 *****
439 ***** ISOLATE BASE IN OPR*
440 *****
441 DB30  GEN   /T(DB30,DB40),5(FT),7(SFTC),13(INCSC),18(SHFTOP),
442          C20(RGHT),XF3
443          ORG  *+1
444 *****
445 ***** COMPLETE FETCH*
446 ***** ISOLATE DISPL *
447 *****
448 DB50  GMSK  /N(DB60),12(A$GPR),16(R8),11(LIT),15(X'F000),
449          C14(AND),13(OPROUT),6(SPEC),10(WAITMD)
450 *****
451 ***** FORCE BASE ADDR TO PRECEDE SIGN*
452 *****
453 DB60  GEN   /N(DB70),11(B$SPEC),23(MIR),12(A$SPEC),22(AONES),
454          C14(DEC8),17(GPROUT),24(R8),7(SSALU)
455 *****
```

```
00 00399
00 00400
00 00401
00 00402
00 00403
00 00404
00 00405
00 00406
00 00407
00 00408
00 00409
00 00410
00 00411
00 00412
00 00413
00 00414
00 00415
00 00416
00 00417
00 00418
00 00419
00 00420
00 00421
00 00422
00 00423
00 00424
00 00425
00 00426
00 00427
00 00428
00 00429
00 00430
00 00431
00 00432
00 00433
00 00434
00 00435
00 00436
00 00437
00 00438
00 00439
00 00440
00 00441
00 00442
00 00443
00 00444
00 00445
00 00446
00 00447
00 00448
00 00449
00 00450
00 00451
00 00452
00 00453
00 00454
00 00455
```



```
456 ***** CREATE BYTE ADDR IN P * 00 00456
457 ***** TEST FOR NEGATIVE OP ADDR* 00 00457
458 ***** 00 00458
459 DB70 GEN /T(DB80,COMP0P),5(TT),7(ALUS),12(ASGPR),24(R8), 00 00459
460 C11(BSSPEC),23(OPR),14(ADD),16(CRY1),13(POUT),SH0 00 00460
461 ***** 00 00461
462 ***** INVALID ADDR= SET CONDIT* 00 00462
463 ***** @ (R1) AND OVERFLOW * 00 00463
464 ***** 00 00464
465 DBCONT EQU * 00 00465
466 ORG X'27 00 00466
467 DB80 GEN /N(DB90),6(MEMC),10(DSSALU),12(ASGPR),24(R1), 00 00467
468 C14(TRNA),15(LOG),TF0,GF2 00 00468
469 ORG DBCONT 00 00469
470 ***** 00 00470
471 ***** COMPLETE CONDIT SET TO 4* 00 00471
472 ***** 00 00472
473 DB90 GMSK /N(COMP0P),6(SPEC),10(WAITMD),11(LIT),15(X'FFFA), 00 00473
474 C14(TRNB) 00 00474
475 EJEC 00 00475
476 ***** 00 00476
477 * 00 00477
478 * TITLE: VALIDATE SIGN (DS00) * 00 00478
479 * * 00 00479
480 * PURPOSE: FETCH SIGN BYTE AND VERIFY -, +, OR BLANK. SET * 00 00480
481 * TO BLANK IF +. SET OVERFLOW AND CONDIT (@R1) TO * 00 00481
482 * 4 IF NON OF ABOVE. * 00 00482
483 * * 00 00483
484 * REGISTER USAGE: ENTRY * 00 00484
485 * R8 = BYTE ADDR OF SIGN * 00 00485
486 * R1 = ADDR OF CONDIT * 00 00486
487 * OPR = DISPL * 00 00487
488 * EXIT * 00 00488
489 * P = SIGN BYTE ( 0255 OR 0) * 00 00489
490 * * 00 00490
491 * CALLED ROUTINES: * 00 00491
492 * DFO0 FETCH BYTGE R8=BYTE ADDR/ BYTE RETURNED IN P * 00 00492
493 * * 00 00493
494 ***** 00 00494
495 ***** 00 00495
496 ***** FETCH SIGN BYTE* 00 00496
497 ***** 00 00497
498 DS00 GMSK /P(DFO0+P),1MD,LB3,15(DS10+P),AK2 00 00498
499 ***** 00 00499
500 ***** TEST FOR '-'+ 00 00500
501 ***** 00 00501
502 DS10 GMSK /N(DS20),11(LIT),15(X'FF52),14(EOR), 00 00502
503 C12(ASP),7(SSALU) 00 00503
504 ***** 00 00504
505 ***** IF - ITS LEGAL ELSE CONTINUE TEST* 00 00505
506 ***** 00 00506
507 DS20 GEN /T(DS30,COMP0P),5(FT),7(ALUZ) 00 00507
508 ***** 00 00508
509 ***** TEST FOR BLANK* 00 00509
510 ***** 00 00510
511 DSC1 EQU * 00 00511
512 ORG X'29 00 00512
```

```
513 DS30 GMSK /N(DS40),11(LIT),15(X'FF5F'),14(EOR), 00 00513
514 C12(ASP),7(SSALU) 00 00514
515 ORG DSC1 00 00515
516 ***** 00 00516
517 ***** IF BLANK ITS LEGAL ELSE TEST AGAIN* 00 00517
518 ***** 00 00518
519 DSC2 EQU * 00 00519
520 ORG X'2B 00 00520
521 DS40 GEN /T(DS60,DS50),5(TT),7(ALUZ) 00 00521
522 ORG DSC2 00 00522
523 ***** 00 00523
524 ***** TEST FOR PLUS* 00 00524
525 ***** 00 00525
526 DS50 GMSK /N(DS60),11(LIT),15(X'FF54'),14(EOR), 00 00526
527 C12(ASP),7(SSALU) 00 00527
528 ***** 00 00528
529 ***** FORCE ZERO IN CASE +* 00 00529
530 ***** IF + EXIT ELSE ERROR * 00 00530
531 ***** 00 00531
532 DS60 GEN /T(DS70,COMPOP),5(FT),7(ALUZ), 00 00532
533 C14(ZERO),15(LOG),13(POUT) 00 00533
534 ***** 00 00534
535 *****LL ILLEGAL SIGN- SET CONDIR* 00 00535
536 *****L (DR1) AND OVERFLOW * 00 00536
537 ***** 00 00537
538 DSC3 EQU * 00 00538
539 ORG X'2D 00 00539
540 DS70 GEN /N(DS80),6(MEMC),10(SSALU),12(ASGPR),24(R1), 00 00540
541 C14(TRNA),15(LOG),TFD,GF2 00 00541
542 ORG DSC3 00 00542
543 ***** 00 00543
544 ***** SET CONDIR TO 4* 00 00544
545 ***** 00 00545
546 DS80 GMSK /N(COMPOP),6(SPEC),10(WAITMD),11(LIT),15(X'FFFB'), 00 00546
547 C14(TRNB) 00 00547
548 ***** 00 00548
549 ***** COMPOP EXIT (MUST BE EVEN ADDR* 00 00549
550 ***** 00 00550
551 EJEC 00 00551
552 ***** 00 00552
553 * 00 00553
554 * TITLE: DIVIDE BY 10 (DV00) * 00 00554
555 * 00 00555
556 * PURPOSE: DIVIDE VALUE IN R5 BY 10. RETURN REMAINDER IN * 00 00556
557 * P, QUOTIENT IN R5. * 00 00557
558 * 00 00558
559 * REGISTER USAGE: ENTRY * 00 00559
560 * R5 =VALUE * 00 00560
561 * EXIT * 00 00561
562 * P =REMAINDER * 00 00562
563 * R5 =QUOTIENT * 00 00563
564 * 00 00564
565 ***** 00 00565
566 ***** 00 00566
567 ***** MOVE VALUE TO P AND* 00 00567
568 ***** TEST FOR NEGATIVE * 00 00568
569 ***** 00 00569
```

```
570 DV00 GEN /N(DV10),12(ASGPR),24(R5),14(TRNA),15(LOG), 00 00570
571 C13(POUT),7(SSALU) 00 00571
572 ***** 00 00572
573 ***** INIT QUOTIENT TO -1 * 00 00573
574 ***** IF DIVIDEND NEG- DIVIDE COMPLETE* 00 00574
575 ***** 00 00575
576 DV10 GEN /T(DV20,DV40),5(FT),7(ALUS),12(ASPEC),22(AONES), 00 00576
577 C14(TRNA),15(LOG),17(GPROUT),24(R5) 00 00577
578 ***** 00 00578
579 ***** EXTRACT NEXT 10* 00 00579
580 ***** 00 00580
581 DV20 GMSK /N(DV30),12(ASP),11(LIT),15(X'9),13(POUT), 00 00581
582 C14(ADD),7(SSALU) 00 00582
583 ***** 00 00583
584 ***** SET DIGIT BACK TO POSITIVE* 00 00584
585 ***** 00 00585
586 DV40 GMSK /N(COMPOP),11(LIT),15(X'FFF5),12(ASP),14(ADD), 00 00586
587 C13(POUT) 00 00587
588 ***** 00 00588
589 ***** INCREMENT QUOTIENT AND TEST * 00 00589
590 ***** FOR COMPLETE DIVIDE (NEGATIVE)* 00 00590
591 ***** 00 00591
592 DV30 GEN /T(DV20,DV40),5(FT),7(ALUS),12(ASGPR),24(R5), 00 00592
593 C14(INCA),16(CRY1),17(GPROUT) 00 00593
594 EJEC 00 00594
595 ***** 00 00595
596 * 00 00596
597 * TITLE: CHECK DIGIT (DH00) * 00 00597
598 * 00 00598
599 * PURPOSE: VALIDATE DIGIT IS ASCII 0-9. * 00 00599
600 * 00 00600
601 * REGISTER USAGE: ENTRY * 00 00601
602 * P CONTAINS DIGIT (ASCII) * 00 00602
603 * 00 00603
604 * EXIT * 00 00604
605 * P,OPR CONTAIN OCTAL DIGIT * 00 00605
606 * IF ERROR, OPR CONTAINS VALUE 4 * 00 00606
607 * 00 00607
608 ***** 00 00608
609 ***** 00 00609
610 ***** SAMPLE DIGIT - ASCII 0 00 00610
611 ***** 00 00611
612 DH00 GMSK /N(DH10),12(ASP),11(LIT),15(X'AF),14(ADD),7(SSALU), 00 00612
613 C13(POUT) 00 00613
614 ***** 00 00614
615 ***** SET ERROR IF ILLEGAL 00 00615
616 ***** 00 00616
617 DH10 GEN /T(DH50,DH20),5(TT),7(ALUS) 00 00617
618 ***** 00 00618
619 ***** SAMPLE FOR LESS THAN 10 00 00619
620 ***** 00 00620
621 DH20 GMSK /N(DH30),12(ASP),11(LIT),15(X'9),14(ADD),7(SSALU), 00 00621
622 ***** 00 00622
623 ***** SET ERROR IF ILLEGAL 00 00623
624 ***** 00 00624
625 DH30 GEN /T(DH50,DH40),5(FT),7(ALUS) 00 00625
626 ***** 00 00626
```

```
627 ***** SAVE DIGIT IN BOTH P AND OPR/EXIT 00 00627
628 ***** 00 00628
629 DH40 GEN 10(STACK),24(POPJMP),LB3,TF0,SF0,BB0,12(ASP),14(TRNA), 00 00629
630 C15(LOG),13(OPROUT),7(S$ALU) 00 00630
631 ***** 00 00631
632 ***** CLEAR STACK AND PLACE 4 IN OPR (INVALID DIGIT) 00 00632
633 ***** 00 00633
634 DH50 GMSK /P(EX00+P),10(STACK),AK4,11(LIT),15(X'FFFB), 00 00634
635 C14(TRNB),13(OPROUT) 00 00635
636 EJEC 00 00636
637 ***** 00 00637
638 * 00 00638
639 * TITLE: FETCH BYTE (DF00 DF10 IS ALTERNATE ENTRY) * 00 00639
640 * 00 00640
641 * PURPOSE: FETCH NEXT BYTE OR RETURN '0' IF LEFT SHIFT * 00 00641
642 * NON ZERO OR OPERAND EXHAUST CD. * 00 00642
643 * 00 00643
644 * REGISTER USAGE: DF00 DF10 * 00 00644
645 * ENTRY R8 = BASE BYTE R8= BYTE ADDR * 00 00645
646 * OPR= LEFT SHIFT COUNT/DISPLACEMENT * 00 00646
647 * 00 00647
648 * EXIT * 00 00648
649 * P = BYTE * 00 00649
650 * OPR= LEFT SHIFT COUNT-1/DISPL-1 * 00 00650
651 * R9 = ORIG DISPL * 00 00651
652 * 00 00652
653 * INTERNAL * 00 00653
654 * MIR= WORD FOR BYTE SELECTION * 00 00654
655 * 00 00655
656 ***** 00 00656
657 ***** 00 00657
658 ***** SAMPLE LEFT SHIFT COUNT* 00 00658
659 ***** 00 00659
660 DF00 GEN /N(DF05),11(B$SPEC),23(OLSE),14(TRNB),15(LOG), 00 00660
661 C7(S$ALU) 00 00661
662 ***** 00 00662
663 ***** SAMPLE BYTE DISPLACEMENT* 00 00663
664 ***** 00 00664
665 * ALTERNATE ENTRY 00 00665
666 DF10 GEN /N(DF15),11(B$SPEC),23(ORZF),14(TRNB),15(LOG), 00 00666
667 C7(S$ALU) 00 00667
668 ***** 00 00668
669 ***** BUILD BYTE ADDR (R8+OPR,RB) * 00 00669
670 ***** IF NON ZERO LEFT SHIFT RETUTN 0* 00 00670
671 ***** 00 00671
672 DF05 GEN /T(DF10,DF25),12(ASGPR),24(R8),11(B$SPEC), 00 00672
673 C23(ORZF),14(ADD),17(GPROUT),5(TT),7(ALUZ) 00 00673
674 ***** 00 00674
675 ***** FETCH WORD (L/R TO BYTA) IF* 00 00675
676 ***** DISPLACEMENT NON ZERO * 00 00676
677 ***** 00 00677
678 DF15 GEN /T(DF40,DF20),5(FT),7(ALUZ),12(ASGPRR),24(R8), 00 00678
679 C6(TESTF),10(OF$ALU),14(TRNA),15(LOG),SH4 00 00679
680 ***** 00 00680
681 ***** RETURN 0 BYTE AND EXIT* 00 00681
682 ***** 00 00682
683 DF20 GMSK /N(CPOP51),11(LIT),15(X'FF4F),14(TRNB), 00 00683
```

```
684          C13(POUT)                                00 00684
685          *****                                00 00685
686          *****  SAVE DISPLACEMENT IN R9*         00 00686
687          *****                                00 00687
688  DF40  GEN      /N(DF50),11(BSSPEC),23(OPR),17(GPROUT),24(R9), 00 00688
689          C14(TRNB),15(LOG),6(SPEC),10(WAITMD)         00 00689
690          *****                                00 00690
691          *****  PREPARE SHIFT COUNT FOR DECREMENT* 00 00691
692          *****                                00 00692
693  DF25  GEN      /N(DF30),11(BSSPEC),23(OPR),14(TRNB),15(LOG), 00 00693
694          C13(POUT)                                00 00694
695          *****                                00 00695
696          *****  DECREMENT SHIFT COUNT*            00 00696
697          *****                                00 00697
698  DF30  GMSK      /N(DF20),11(LIT),15(X'00FF'),12(ASP),14(ADD), 00 00698
699          C13(OPROUT)                                00 00699
700          *****                                00 00700
701          *****  ITS LEFT BYTE (MOVE LEFT TO RIGHT)* 00 00701
702          *****                                00 00702
703  DF60  GEN      /N(DF70),11(BSSPEC),23(OLSE),14(TRNB),15(LOG), 00 00703
704          C13(OPROUT)                                00 00704
705          *****                                00 00705
706          *****  SELECT RIGHT BYTE*                00 00706
707          *****                                00 00707
708  DF70  GEN      /N(DF80),11(BSSPEC),23(ORZF),14(TRNB),15(LOG), 00 00708
709          C13(POUT)                                00 00709
710          *****                                00 00710
711          *****  PLACE BYTE IN OPR*                00 00711
712          *****  SELECT RIGHT/LEFT *              00 00712
713          *****                                00 00713
714  DF50  GEN      /F(DF60),FS2,2(1),11(BSSPEC),23(MIR),      00 00714
715          C14(TRNB),15(LOG),13(OPROUT)                00 00715
716          *****                                00 00716
717          *****  PLACE NEW DISPLACEMENT IN OPR*     00 00717
718          *****                                00 00718
719  DF80  GEN      /N(CPOPS1),12(ASGPR),24(R9),14(DECA),13(OPROUT) 00 00719
720  EJECT                                              00 00720
721          *****                                00 00721
722  *                                              * 00 00722
723  * TITLE:      ZERO FILL      (DZ00)                * 00 00723
724  *                                              * 00 00724
725  * PURPOSE:    FILL REMAINDER OF MEM ACC WITH ZEROES  * 00 00725
726  *                                              * 00 00726
727  * REGISTER USAGE:  ENTRY                            * 00 00727
728  *              SC = -ZERO COUNT                     * 00 00728
729  *              OPR = INITIAL DISPLACEMENT           * 00 00729
730  *              R8 = BASE BYTE POINTER               * 00 00730
731  *              *                                     * 00 00731
732  *              INTERNAL                             * 00 00732
733  *              RC = CLEARED FOR ROUTINE TO ROUND    * 00 00733
734  *              P  = 0 FOR ZERO FILL                  * 00 00734
735  *              *                                     * 00 00735
736          *****                                00 00736
737          *****                                00 00737
738          *****  CLEAR ROUND/CARRY AND*             00 00738
739          *****  INIT 0 FILL VALUE *                 00 00739
740          *****                                00 00740
```

```
741 DZ00 GEN /N(DZ10),12(ASPEC),22(AZERO),14(TRNA),15(LOG), 00 00741
742 C17(GPROUT),24(RC),13(POUT) 00 00742
743 ***** 00 00743
744 ***** STORE ZERO BYTE* 00 00744
745 ***** 00 00745
746 DZCONT EQU * 00 00746
747 ORG X'2F 00 00747
748 DZ10 GMSK /P(DR00+P),IMD,LB3,15(DZ20+P),AK2 00 00748
749 ORG DZCONT 00 00749
750 ***** 00 00750
751 ***** IF STORE COMPLETE, EXIT,* 00 00751
752 ***** ELSE CONTINUE * 00 00752
753 ***** 00 00753
754 DZ20 GEN /T(DZ10,COMPOP),13(INCSC),7(SFTC),5(FT) 00 00754
755 EJEC 00 00755
756 ***** 00 00756
757 * 00 00757
758 * TITLE: SUPPRESS LEADING ZEROES * 00 00758
759 * 00 00759
760 * PURPOSE: RETURN COUNT OF LEADING ZEROES. * 00 00760
761 * 00 00761
762 * REGISTER USAGE: ENTRY * 00 00762
763 * OPR,RB = NO. BYTES OF VALUE * 00 00763
764 * R8 = BASE BYTE POINTER * 00 00764
765 * EXIT * 00 00765
766 * OPR = NO. OF ZERO BYTES * 00 00766
767 * 00 00767
768 * INTERNAL * 00 00768
769 * SC = -NO. BYTES OF VALUE * 00 00769
770 * P = ZERO COUNT * 00 00770
771 * NEXT BYTE * 00 00771
772 * R9 = CURRENT BYTE DISPLACEMENT * 00 00772
773 * 00 00773
774 * CALLED ROUTINE: * 00 00774
775 * DFOO FETCH BYTE OPR=DISPL/R8=BYTE BASE * 00 00775
776 * BYTE RETURNED IN P/OPR VALUE IN R9 * 00 00776
777 * 00 00777
778 ***** 00 00778
779 ***** 00 00779
780 *****LL MOVE LENGTH TO SHIFT COUNT. 00 00780
781 ***** 00 00781
782 DL00 GEN /N(DL10),11(BSPEC),23(ORZF),14(NTB),15(LOG), 00 00782
783 C13(SCOUT),7(S$ALU),TF0,SFO 00 00783
784 ***** 00 00784
785 ***** INIT ZERO COUNT * 00 00785
786 ***** IF LENGTH IS 0 GO EXIT* 00 00786
787 ***** 00 00787
788 DL10 GEN /T(DL20,DL60),5(FT),7(ALU0),12(ASPEC),22(AZERO), 00 00788
789 C14(INCA),16(CRY1),13(POUT) 00 00789
790 ***** 00 00790
791 ***** FETCH BYTE AND SAVE ZERO COUNT* 00 00791
792 ***** 00 00792
793 DL20 GMSK /P(DFOO+P),IMD,LB3,15(DL30+P),AK2,12(ASP),14(TRNA), 00 00793
794 C13(OPROUT) 00 00794
795 ***** 00 00795
796 ***** TEST BYTE FOR 0* 00 00796
797 ***** 00 00797
```

```
798 DL30 GMSK /N(DL40),12(ASP),14(SUB),16(CRY1),11(LIT), 00 00798
799 C15(X'FF4F),7(SSALU) 00 00799
800 ***** 00 00800
801 ***** RESTORE ORIG ZERO COUNT (NOW R9) IN P* 00 00801
802 ***** DONE IF NON ZERO CHARAC * 00 00802
803 ***** 00 00803
804 DL40 GEN /T(DL50,DL60),5(TT),7(ALUZ),12(ASGPR),24(R9), 00 00804
805 C14(TRNA),15(LOG),13(POUT) 00 00805
806 ***** 00 00806
807 ***** RESTORE R8 TO BASE BYTE/INCREMENT * 00 00807
808 ***** ZERO COUNT/TEST FOR VALUE ALL ZERO* 00 00808
809 ***** 00 00809
810 DL50 GEN /N(DL55),12(ASGPR),24(R8), 00 00810
811 C11(BSGPR),23(R9),14(SUB),16(CRY1),17(GPROUT), 00 00811
812 C13(INCSC) 00 00812
813 ORG X'31 00 00813
814 DL55 GEN /T(DL20,DL60),5(FT),7(SFTC),13(INCP) 00 00814
815 ORG DL50+1 00 00815
816 ***** 00 00816
817 ***** EXIT= SET TRUE ZERO COUNT* 00 00817
818 ***** 00 00818
819 DL60 GMSK IMD,LB3,AK4,12(ASP),15(X'0),14(ADD), 00 00819
820 C13(OPROUT) 00 00820
821 EJEC 00 00821
822 ***** 00 00822
823 * 00 00823
824 * TITLE: STORE RESULT BYTE (DROO DR30 ALTERNATE ENTRY) * 00 00824
825 * 00 00825
826 * PURPOSE: STORE BYTE IN P AT BYTE ADDR FORMED IN OPR AND R8. * 00 00826
827 * 00 00827
828 * REGISTER USAGE: ENTRY * 00 00828
829 * P =ASCII BYTE * 00 00829
830 * OPR=RIGHT SHIFT COUNT/DISPLACEMENT * 00 00830
831 * R8 =BYTE BASE ADDR * 00 00831
832 * RC =ROUND CARRY * 00 00832
833 * 00 00833
834 * EXIT * 00 00834
835 * P =INTACT * 00 00835
836 * OPR=RT COUNT-1/DISPL-1 * 00 00836
837 * R8 =INTACT * 00 00837
838 * RC =NEW CARRY * 00 00838
839 * 00 00839
840 ***** 00 00840
841 ***** 00 00841
842 ***** TEST RIGHT SHIFT COMPLETE 00 00842
843 ***** 00 00843
844 DROO GEN /N(DRO2),11(BSSPEC),23(OLSE),14(DECB),12(ASPEC), 00 00844
845 C22(AONES),7(SSALU) 00 00845
846 ***** 00 00846
847 ***** MASK DIGIT FROM ASCII 00 00847
848 ***** IF SHIFT COMPLETE, GO STORE DIGIT 00 00848
849 ***** 00 00849
850 DRO2 GMSK /T(DRO4,DR16),5(FT),7(ALUS),12(ASP),11(LIT), 00 00850
851 C15(X'FFF0),14(AND),13(POUT) 00 00851
852 ***** 00 00852
853 ***** RTSHFT PREPARE SHIFT COUNT DECREMENT (OPR TO R9) 00 00853
854 ***** IF COMPLETE TEST FOR ROUND 00 00854
```


855	*****			00 00855
856	DR04	GEN	/T(DR06,DR14),5(TT),7(ALUZ),11(B\$SPEC),23(OPR),	00 00856
857			C14(TRNB),15(LOG),17(GPROUT),24(R9)	00 00857
858	*****			00 00858
859	*****	RTSHFT	ROUNDING SPECIFIED IF OVERFLOW SET	00 00859
860	*****			00 00860
861	DR06	GEN	/T(DR08,DR14),5(TT),7(OVFL)	00 00861
862	*****			00 00862
863	*****	ROUND	TEST FOR CARRY GENERATED BY ROUND	00 00863
864	*****			00 00864
865	DR08	GMSK	/N(DR10),12(ASP),11(LIT),15(X'4),14(ADD),7(S\$ALU),	00 00865
866	*****			00 00866
867	*****	RTSHFT	DECREMENT RIGHT SHIFT COUNT AND EXIT	00 00867
868	*****			00 00868
869	DR14	GMSK	/N(CPOP51),12(ASGPR),16(R9),11(LIT),15(X'00FF),	00 00869
870			C14(ADD),13(OPROUT)	00 00870
871	*****			00 00871
872	*****	ROUND	CLEAR NEXT CARRY AND TEST FOR CARRY GENERATED	00 00872
873	*****			00 00873
874	DR10	GEN	/T(DR12,DR14),12(AS\$SPEC),22(AZERO),14(TRNA),	00 00874
875			C15(LOG),17(GPROUT),24(RC),5(FT),7(ALUS)	00 00875
876	*			00 00876
877	*		ADD CARRY TO MASKED DIGIT	00 00877
878	*			00 00878
879	DR16	GEN	/N(DR18),12(ASP),11(B\$GPR),23(RC),14(ADD),13(POUT)	00 00879
880	*****			00 00880
881	*****	ROUND	CARRY GENERATED	00 00881
882	*****			00 00882
883	DR12	GEN	/N(DR14),12(AS\$SPEC),22(AZERO),14(INCA),16(CRY1),	00 00883
884			C17(GPROUT),24(RC)	00 00884
885	*****			00 00885
886	*****		SAMPLE FOR PROPAGATED CARRY	00 00886
887	*****			00 00887
888	DR18	GMSK	/N(DR20),12(ASP),11(LIT),15(X'9),14(ADD),7(S\$ALU),	00 00888
889	*****			00 00889
890	*****		INIT CARRY AND TEST FOR PROPAGATION	00 00890
891	*****			00 00891
892	DR20	GEN	/T(DR22,DR26),5(FT),7(ALUS),12(AS\$SPEC),22(AZERO),	00 00892
893			C14(TRNA),15(LOG),17(GPROUT),24(RC)	00 00893
894	*****			00 00894
895	*****		PROPAGATE BY DEDUCTING 10 FROM CURRENT	00 00895
896	*****			00 00896
897	DR22	GMSK	/N(DR24),12(ASP),11(LIT),15(X'9),14(ADD),13(POUT),	00 00897
898	*****			00 00898
899	*****		SET CARRY FOR NEXT DIGIT	00 00899
900	*****			00 00900
901	DR24	GEN	/N(DR26),12(AS\$SPEC),22(AZERO),14(INCA),16(CRY1),	00 00901
902			C17(GPROUT),24(RC)	00 00902
903	*****			00 00903
904	*****		TEST FOR DIGITS EXHAUSTED	00 00904
905	*****			00 00905
906	DR26	GEN	/N(DR27),11(B\$SPEC),23(ORZF),14(TRNB),15(LOG),7(S\$ALU)	00 00906
907	*****			00 00907
908	*****		SAMPLE FOR NON ZERO DIGIT WHEN STORE EXHAUSTED	00 00908
909	*****			00 00909
910	DR48	GEN	/N(DR50),12(ASP),14(TRNA),15(LOG),7(S\$ALU)	00 00910
911	*****			00 00911


```
912 ***** NOT EXHAUSTED- BUILD ASCII DIGIT 00 00912
913 ***** 00 00913
914 DR28 GMSK /N(DR29),12(ASP),11(LIT),15(X'FF4F'),14(ADD),13(POUT) 00 00914
915 ***** 00 00915
916 ***** PLACE BYTE IN OPR WITH DISPLACEMENT 00 00916
917 ***** 00 00917
918 DR29 GEN /N(DR30),12(ASP),11(B$SPEC),23(ORLZ),14(OR),13(OPROUT) 00 00918
919 ***** 00 00919
920 ***** ALT BUILD WORD ADDR (SAVE BYTE POSITION- BYTA) 00 00920
921 ***** ENTRY 00 00921
922 ***** 00 00922
923 DR30 GEN /N(DR32),12(ASGPRR),24(R8),14(TRNA),15(LOG),SH4 00 00923
924 ***** 00 00924
925 ***** BEGIN BYTE STORE- SELECT BYTE 00 00925
926 ***** 00 00926
927 DR32 GEN /F(DR36),FS2,2(1),12(ASGPRR),24(R8), 00 00927
928 C14(TRNA),15(LOG),SH4,6(MEMC),10(B$ALU) 00 00928
929 ***** 00 00929
930 ***** STORE LEFT BYTE 00 00930
931 ***** 00 00931
932 DR36 GEN /N(DR38),11(B$SPEC),23(ORLZ),14(TRNB),15(LOG),6(SPEC), 00 00932
933 C10(WAITMD) 00 00933
934 ***** 00 00934
935 ***** STORE RIGHT BYTE 00 00935
936 ***** 00 00936
937 DR34 GEN /N(DR38),11(B$SPEC),23(ORZF),14(TRNB),15(LOG),6(SPEC), 00 00937
938 C10(WAITMD) 00 00938
939 ***** 00 00939
940 ***** DECREMENT BYTE STORE COUNT AND SAMPLE 00 00940
941 ***** 00 00941
942 DR38 GEN /N(DR40),11(B$SPEC),23(OLSE),12(AS$SPEC),22(AONES), 00 00942
943 C14(DEC8),13(OPROUT),7(S$ALU) 00 00943
944 ***** 00 00944
945 ***** RESTORE BASE IN R8 AND TEST FOR COMPLETE STORE 00 00945
946 ***** 00 00946
947 DR40 GEN /T(DR42,DR52),5(TT),7(ALUZ),12(ASGPR),24(R8), 00 00947
948 C11(B$SPEC),23(ORZF),14(SUB),17(GPROUT) 00 00948
949 ***** 00 00949
950 ***** COMPLETE STORE- SAMPLE FOR MORE CARRY 00 00950
951 ***** 00 00951
952 DR42 GEN /N(DR44),12(ASGPR),24(RC),14(TRNA),15(LOG),7(S$ALU) 00 00952
953 ***** 00 00953
954 ***** IF CARRY WE OVERFLOWED, ELSE EXIT 00 00954
955 ***** 00 00955
956 DR44 GEN /T(DR46,DR52),5(FT),7(ALUZ) 00 00956
957 ***** 00 00957
958 ***** CLEAN UP STACK AND ANNOUNCE ERROR (CONDIT=3) 00 00958
959 ***** 00 00959
960 DR46 GEN /P(ER03+P),10(STACK),23(POPDEL),LB3,TF0,SF0,AA4 00 00960
961 ***** 00 00961
962 ***** BUILD BYTE ADDRESS AND 00 00962
963 ***** TEST FOR DIGIT EXHAUSTION 00 00963
964 ***** 00 00964
965 DR27 GEN /T(DR48,DR28),12(ASGPR),24(R8),11(B$SPEC),23(ORZF), 00 00965
966 C14(ADD),17(GPROUT),5(TT),7(ALUZ) 00 00966
967 ***** 00 00967
968 ***** EXIT SUBROUTINE 00 00968
```

```
969 ***** 00 00969
970 DR52 EQU * 00 00970
971 CPOP51 GEN IMD, LB3, AA4 00 00971
972 ***** 00 00972
973 ***** IF NON ZERO, ANNOUNCE ERROR (CONDIT=3) 00 00973
974 ***** 00 00974
975 DR50 GEN /T(CPOP51, DR46), 5(TT), 7(ALUZ) 00 00975
976 EJECE 00 00976
977 ***** 00 00977
978 * 00 00978
979 * TITLE: STORE SIGN (DN00) * 00 00979
980 * 00 00980
981 * PURPOSE: STORE SIGN IN MEMORY ACCUMULATOR * 00 00981
982 * 00 00982
983 * REGISTER USAGE: ENTRY * 00 00983
984 * P =SIGN (0=+, NON ZERO=-) * 00 00984
985 * RF =DISPL FOR BYTE * 00 00985
986 * RA =BYTE POINTER * 00 00986
987 * 00 00987
988 * CALLED ROUTINES: * 00 00988
989 * DR00: STORE BYTE IN P AT ADDR R8+OPR * 00 00989
990 * 00 00990
991 ***** 00 00991
992 ***** 00 00992
993 ***** SAMPLE SIGN FOR + (P=0) OR - (P NOT 0) 00 00993
994 ***** 00 00994
995 DN00 GEN /N(DN10), 12(ASP), 14(TRNA), 15(LOG), 7(SSALU) 00 00995
996 ***** 00 00996
997 ***** SET BYTE ADDR (RD) IN R8 AND COMPLETE SIGN TEST 00 00997
998 ***** 00 00998
999 DN10 GEN /T(DN30, DN20), 5(FT), 7(ALUZ), 11(BSGPR), 23(RD), 00 00999
1000 C14(TRNB), 15(LOG), 17(GPROUT), 24(R8) 00 01000
1001 ***** 00 01001
1002 ***** SET + SIGN (BLANK) 00 01002
1003 ***** 00 01003
1004 DN20 GMSK /N(DN40), 11(LIT), 15(X'FF5F), 14(TRNB), 13(OPROUT) 00 01004
1005 ***** 00 01005
1006 ***** SET - SIGN 00 01006
1007 ***** 00 01007
1008 DN30 GMSK /N(DN40), 11(LIT), 15(X'FF52), 14(TRNB), 13(OPROUT) 00 01008
1009 ***** 00 01009
1010 ***** STORE SIGN AND EXIT 00 01010
1011 ***** 00 01011
1012 DN40 GEN /N(DN40P), 12(ASGPR), 24(R8), 00 01012
1013 C11(BSGPR), 23(RF), 14(ADD), CF3, 17(GPROUT) 00 01013
1014 ORG X'55 00 01014
1015 DN40P GMSK /P(DR30+P), 10(STACK), 16(PUSH), 15(COMPOP+P), 00 01015
1016 CLB3, SFO, TFO 00 01016
1017 ORG DN40+1 00 01017
1018 EJECE 00 01018
1019 ***** 00 01019
1020 * 00 01020
1021 * TITLE: MOVE OPA TO TEMP (DM00) * 00 01021
1022 * 00 01022
1023 * PURPOSE: BYTE BY BYTE MOVE OF OPA TO TEMP AREA * 00 01023
1024 * 00 01024
1025 * REGISTER USAGE: ENTRY * 00 01025
```

```
1026 *      RO  =TEMP ADDR *      00 01026
1027 *      RA  =OPA ADDR  (BYTE) *      00 01027
1028 *      RB  =OPB LENGTH *      00 01028
1029 * *      00 01029
1030 *      EXIT *      00 01030
1031 *      RA  =NEW OPA ADDR (IN TEMP) (BYTE) *      00 01031
1032 * *      00 01032
1033 *      INTERNAL *      00 01033
1034 *      OPR  =LENGTH (BYTE) *      00 01034
1035 *      SC   =-LENGTH FOR MOVE *      00 01035
1036 *      R5   BUILD BYTE ADDR IN TEMP *      00 01036
1037 *      R8   PASS BYTE ADDR TO DFOO/DR00 *      00 01037
1038 * *      00 01038
1039 * CALLING SEQUENCE;      NORMAL BRANCH TO DM10; *      00 01039
1040 * *      00 01040
1041 * *      00 01041
1042 * CALLED ROUTINES; *      00 01042
1043 *      DFOO  FETCH BYTE  ADDR IN R8, DISPL IN OPR/ RETURNED IN P *      00 01043
1044 *      DR00  STORE BYTE  ADDR IN R8, DISPL IN OPR, BYTE IN P *      00 01044
1045 * *      00 01045
1046 * ***** *      00 01046
1047 *      ORG    **1 *      00 01047
1048 * ***** *      00 01048
1049 * ***** SET LENGTH IN SHIFT COUNT *      00 01049
1050 * ***** *      00 01050
1051 * DM10  GEN    /N(DM20),11(B$SPEC),23(ORZF),14(TCB),16(CRY1), *      00 01051
1052 *      C13(SCOUT),12(A$SPEC),22(AZERO) *      00 01052
1053 * ***** *      00 01053
1054 * ***** TRANSFER TEMP BLOCK ADDR IN R0 TO R5 *      00 01054
1055 * ***** *      00 01055
1056 * DM20  GEN    /N(DM30),11(B$GPR),23(R0),14(TRNB),15(LOG), *      00 01056
1057 *      C17(GPROUT),24(R5) *      00 01057
1058 * ***** *      00 01058
1059 * ***** BUILD BYTE ADDR IN R5 *      00 01059
1060 * ***** *      00 01060
1061 * DM30  GEN    /N(DM40),12(A$GPRL),24(R5),14(INCA),16(CRY1), *      00 01061
1062 *      C17(GPROUT),SHO *      00 01062
1063 * ***** *      00 01063
1064 * ***** FETCH BYTE *      00 01064
1065 * ***** *      00 01065
1066 * DM50  GMSK    /P(DFOO+P),IMD,LB3,15(DM60+P),AK2 *      00 01066
1067 * ***** *      00 01067
1068 * ***** RESET DISPLACEMENT *      00 01068
1069 * ***** *      00 01069
1070 * DM60  GEN    /N(DM70),11(B$SPEC),23(OPR),14(INCB),16(CRY1), *      00 01070
1071 *      C13(OPROUT),12(A$SPEC),22(AZERO) *      00 01071
1072 * ***** *      00 01072
1073 * ***** PLACE STORE ADDR IN R8 *      00 01073
1074 * ***** *      00 01074
1075 * DM70  GEN    /N(DM80),11(B$GPR),23(R5),14(TRNB),15(LOG), *      00 01075
1076 *      C17(GPROUT),24(R8) *      00 01076
1077 * ***** *      00 01077
1078 * ***** STORE BYTE *      00 01078
1079 * ***** *      00 01079
1080 * DM80  GMSK    /P(DR00+P),IMD,LB3,15(DM90+P),AK2 *      00 01080
1081 * ***** *      00 01081
1082 * ***** TEST FOR MOVE COMPLETE *      00 01082
```

```
1083 *****
1084 DM90 GEN /T(DM95,DM40),5(TT),7(SFTC),13(INCSC)
1085 *****
1086 ***** PLACE OPERAND A ADDR IN R8
1087 *****
1088 DM40 GEN /N(DM50),11(BSGPR),23(RA),14(TRNB),15(LOG),
1089 C17(GPROUT),24(R8)
1090 *****
1091 ***** RETURN NEW BYTE ADDR IN RA
1092 *****
1093 DM95 GEN /N(DA05),11(BSGPR),23(R5),14(TRNB),15(LOG),
1094 C17(GPROUT),24(RA)
1095 EJEC
1096 *****
1097 *
1098 * TITLE: BYTE BLOCK MOVE AND COMPARE
1099 *
1100 * PURPOSE: MOVE BYTE BLOCK OR COMPARE TWO BYTE BLOCKS.
1101 * OPTIONALLY MAY REQUEST NO INCREMENT OF OP1
1102 * ADDRESS (STORE OR COMPARE 1 BYTE WITH A BLOCK
1103 * OF MEMORY).
1104 * FOR COMPARE SET CONDIT TO:
1105 * 0 IF OP1 < OP2
1106 * 1 IF OP1 = OP2
1107 * 2 IF OP1 > OP2
1108 *
1109 * REGISTER USAGE:
1110 * RD = ADDR OF DESCRIPTIVE PARAMETER BLOCK
1111 * R1 = ADDR OF CONDIT
1112 *
1113 * INTERNAL
1114 * RD = OP1 ADDR (BYTE)
1115 * RE = OP2 ADDR (BYTE)
1116 * RF = LOOP COUNT
1117 * RA = SAVE P
1118 * R9 = COMPARE FLAG
1119 * R8 = PASS ADDR TO BYTE FETCH/STORE ROUTINES
1120 * P = RETURN FETCH BYTE
1121 * OPR= PASS STORE BYTE
1122 * RB = PASS STORE BYTE
1123 *
1124 *
1125 *
1126 * CALLED ROUTINES
1127 * LDRD = LOAD REGISTERS ) SHIFT COUNT=4
1128 * STRD = STORE REGISTERS ) LOAD/STORE 3 REGS RD=RF
1129 * DF10 = FETCH BYTE R8=ADDR(BYTE)/ RETURN IN P
1130 * (OPR NON 0) R9 RETURNS ORIG OPR
1131 * DR30 = STORE BYTE R8=ADDR(BYTE)/ OPR=BYTE
1132 * PARAMETER BLOCK:
1133 * WORD 0 = OP1 BYTE ADDR
1134 * WORD 1 = OP2 BYTE ADDR
1135 * WORD 2 = BYTE COUNT
1136 *
1137 * BCS VALUES:
1138 * 105223 = MOVE WITHOUT INCREMENT
1139 * 105263 = MOVE WITH INCREMENT
```

```
00 01083
00 01084
00 01085
00 01086
00 01087
00 01088
00 01089
00 01090
00 01091
00 01092
00 01093
00 01094
00 01095
00 01096
00 01097
00 01098
00 01099
00 01100
00 01101
00 01102
00 01103
00 01104
00 01105
00 01106
00 01107
00 01108
00 01109
00 01110
00 01111
00 01112
00 01113
00 01114
00 01115
00 01116
00 01117
00 01118
00 01119
00 01120
00 01121
00 01122
00 01123
00 01124
00 01125
00 01126
00 01127
00 01128
00 01129
00 01130
00 01131
00 01132
00 01133
00 01134
00 01135
00 01136
00 01137
00 01138
00 01139
```

```
1140 * 0105323 - COMPARE WITHOUT INCREMENT * 00 01140
1141 * 0105363 - COMPARE WITH INCREMENT * 00 01141
1142 * * 00 01142
1143 * NOTE: ON EXIT * 00 01143
1144 * RO,R1 INTACT * 00 01144
1145 * PARAMETER BLOCK INTEGRITY NOT MAINTAINED * 00 01145
1146 * * 00 01146
1147 * ..... * 00 01147
1148 * ..... * 00 01148
1149 * ..... COMPARE=COMPLETE RESULT STORE * 00 01149
1150 * ..... * 00 01150
1151 * ORG X'DA * 00 01151
1152 MCC070 GEN /N(MCC080),10(WAITMD),6(SPEC),12(ASGPR),24(R9), * 00 01152
1153 C14(TRNA),15(LOG) * 00 01153
1154 MC030 EQU *+1 * 00 01154
1155 * ..... * 00 01155
1156 * ..... PREPARE FOR PARAMETER BLOCK STORE * 00 01156
1157 * ..... * 00 01157
1158 * ORG X'DF * 00 01158
1159 MCC080 GEN /N(MCLO70),12(ASGPR),24(RD),14(TRNA),15(LOG), * 00 01159
1160 C13(POUT) * 00 01160
1161 * ..... * 00 01161
1162 * ..... SET NUMBER OF REGS TO LOAD * 00 01162
1163 * ..... * 00 01163
1164 * ORG X'D9 * 00 01164
1165 MC000 GMSK /N(MC030),11(LIT),15(X'0003),14(TRNB), * 00 01165
1166 C13(SCOUT) * 00 01166
1167 * ..... * 00 01167
1168 * ..... PLACE LOAD ADDR IN P * 00 01168
1169 * ..... * 00 01169
1170 * ORG MC030 * 00 01170
1171 MC030 GEN /N(MC040),12(ASGPR),24(RD),14(TRNA),15(LOG), * 00 01171
1172 C13(POUT) * 00 01172
1173 * ..... * 00 01173
1174 * ..... CALL LOAD ROUTINE * 00 01174
1175 * ..... * 00 01175
1176 MC040 GMSK /P(LDRD+P),IMD,LB3,15(MC050+P),AK2 * 00 01176
1177 * ORG X'80 * 00 01177
1178 * ..... * 00 01178
1179 * ..... TEST FOR ZERO LENGTH MOVE * 00 01179
1180 * ..... * 00 01180
1181 MC050 GEN /N(MCLO10),12(ASGPR),24(RF),14(TRNA),15(LOG),7(S$ALU) * 00 01181
1182 * ORG X'EO * 00 01182
1183 * ..... * 00 01183
1184 * ..... NO INTERRUPT= SELECT BLOCK * 00 01184
1185 * ..... MOVE OR FILL * 00 01185
1186 * ..... * 00 01186
1187 MCNOT1 GEN /F(MCLO40),2(X'1),FS9,12(ASGPR),24(RB),14(TRNA), * 00 01187
1188 C15(LOG),13(OPROUT) * 00 01188
1189 * ..... * 00 01189
1190 * ..... GO STORE CONTEXT * 00 01190
1191 * ..... * 00 01191
1192 MCLO80 GMSK /P(STRD+P),IMD,LB3,15(RLSXIT+P),AK2 * 00 01192
1193 * ..... * 00 01193
1194 * ..... PLACE OP2 ADDR IN R8 * 00 01194
1195 * ..... SELECT OPERATION * 00 01195
1196 * ..... * 00 01196
```

```
1197 MCL040 GEN /F(MCM010),11(BSGPR),23(RE),14(TRNB),15(LOG), 00 01197
1198 C17(GPROUT),24(RB),2(X'1),FSA 00 01198
1199 ***** 00 01199
1200 ***** INCREMENT OP1 BYTE ADDR* 00 01200
1201 ***** 00 01201
1202 MCL00P GEN /N(MCL010),12(ASGPR),24(RD),14(INCA),16(CRY1), 00 01202
1203 C17(GPROUT),7(SSALU) 00 01203
1204 ***** 00 01204
1205 ***** PASS OP1 ADDR IN R8* 00 01205
1206 ***** 00 01206
1207 MCL010 GEN /T(MCL020,RLSXIT),5(FT),7(ALUZ),11(BSGPR),23(RD), 00 01207
1208 C14(TRNB),15(LOG),17(GPROUT),24(RB) 00 01208
1209 ***** 00 01209
1210 ***** CALL OP1 BYTE FETCH* 00 01210
1211 ***** 00 01211
1212 MCL020 GMSK /P(DF10+P),IMD,LB3,15(MCL025+P),AK2 00 01212
1213 ***** 00 01213
1214 ***** PLACE P BYTE IN RB * 00 01214
1215 ***** 00 01215
1216 MCINT EQU *+1 00 01216
1217 ORG X'7B 00 01217
1218 MCL025 GEN /N(MCL040),12(ASP),14(TRNA),15(LOG),17(GPROUT), 00 01218
1219 C24(RB),13(OPROUT) 00 01219
1220 ORG MCINT 00 01220
1221 ***** 00 01221
1222 ***** INTRPT -SET NUMBER OF REGS TO STORE* 00 01222
1223 ***** 00 01223
1224 MCINT GMSK /N(MCL010),11(LIT),15(X'0003),14(TRNB), 00 01224
1225 C13(SCOUT) 00 01225
1226 ***** 00 01226
1227 ***** MOVE CALL STORE ROUTINE* 00 01227
1228 ***** 00 01228
1229 MCM010 GMSK /P(DR30+P),IMD,LB3,15(MCL050+P),AK2 00 01229
1230 ***** 00 01230
1231 ***** OPR HAS PHONY BYTE DISPLACEMENT FOR DF00 00 01231
1232 ***** 00 01232
1233 MCC010 GMSK /N(MCC015),11(LIT),15(X'FFF1),14(TRNB),13(OPROUT), 00 01233
1234 MCL050 ORG *+1 00 01234
1235 ***** 00 01235
1236 ***** DECREMENT LOOP COUNT* 00 01236
1237 ***** 00 01237
1238 MCL050 GEN /N(MCL060),12(ASGPR),24(RF),14(DECA),17(GPROUT), 00 01238
1239 C7(SSALU) 00 01239
1240 ***** 00 01240
1241 ***** TEST FOR COMPLETE* 00 01241
1242 ***** INCREMENT OP2 ADDR* 00 01242
1243 ***** 00 01243
1244 MCL060 GEN /T(MCINT1,MCEXIT),5(FT),7(ALUZ),12(ASGPR),24(RE), 00 01244
1245 C14(INCA),16(CRY1),17(GPROUT) 00 01245
1246 ***** 00 01246
1247 ***** INTRPT -FIRST TEST FOR INTERRUPTS* 00 01247
1248 ***** 00 01248
1249 MCINT1 GEN /N(MCNOT1),TS7,GF4,TF0,SF0,IMC 00 01249
1250 ***** 00 01250
1251 ***** EXIT SELECT EXIT(MOVE) OR* 00 01251
1252 ***** FLAG STORE (COMPARE)* 00 01252
1253 ***** 00 01253
```

```
1254 MCEXIT GEN      /F(MCLO70),2(X'1),FSA,12(ASGPR),24(RD),14(TRNA),
1255                C15(LOG),13(POUT)
1256 *****
1257 ***** INTRPT -PLACE STORE ADDR IN RP*
1258 *****
1259 MCIO10 GEN      /F(MCIO20),12(ASGPR),24(RD),14(TRNA),15(LOG),
1260                C13(POUT),2(X'1),FS9
1261 *****
1262 ***** INTRPT -CALL STORE ROUTINE*
1263 *****
1264 MCIO20 GMSK      /P(STRD+P),IMD,LB3,15(MCIO30+P),AK2
1265 *****
1266 ***** IF BLOCK MOVE, GET TO NEXT BYTE
1267 *****
1268 MCIO15 GEN      /N(MCIO20),12(ASGPR),24(RD),14(INCA),16(CRY1),
1269                C17(GPROUT)
1270 *****
1271 ***** RESTORE P AND EXIT*
1272 *****
1273 MCIO30 GEN      /N(IEEXIT),11(BSGPR),23(RA),14(DECB),
1274                C17(GPROUT),24(RE),12(ASPEC),22(AONES)
1275 *****
1276 ***** COMPARE-MAKE COMPARISON (OPR-RB)*
1277 *****
1278 MCCO20 GEN      /N(MCCO30),12(ASP),11(BSGPR),23(RB),14(SUB),
1279                C7(SSALU),16(CRY1)
1280 *****
1281 ***** COMPARE-CLEAR RESULT FLAG*
1282 *****
1283 MCCO50 GEN      /N(MCCO60),12(ASPEC),22(AZERO),14(TRNA),15(LOG),
1284                C17(GPROUT),24(R9)
1285 *****
1286 ***** COMPARE-INITIALIZE RESULT TEST FLAG (R9)*
1287 ***** TO 1, TEST FOR UNEQUAL *
1288 *****
1289 MCCO30 GEN      /T(MCLO50,MCCO40),5(TT),7(ALUZ),12(ASPEC),
1290                C22(AZERO),14(INCA),16(CRY1),17(GPROUT),24(R9)
1291 *****
1292 ***** SET CONTEXT STORE AMOUNT
1293 *****
1294 MCLO70 GMSK      /N(MCLO80),11(LIT),15(X'0003),14(TRNB),13(SCOUT)
1295 *****
1296 ***** COMPARE-STORE RESULT FLAG*
1297 *****
1298 MCCO60 GEN      /N(MCCO70),6(MEMC),10(SSALU),12(ASGPR),24(R1),
1299                C14(TRNA),15(LOG)
1300 *****
1301 ***** COMPARE-SET FLAG TO 2, TEST*
1302 ***** FOR GREATER *
1303 *****
1304 MCCO40 GEN      /T(MCCO60,MCCO50),5(TT),7(ALUS),12(ASGPR),24(R9),
1305                C17(GPROUT),14(INCA),16(CRY1)
1306 *****
1307 ***** COMPARE-CALL FETCH ROUTINE*
1308 *****
1309 MCCO15 GMSK      /P(DF10+P),IMD,LB3,15(MCCO20+P),AK2
1310 EJECE
```

```
00 01254
00 01255
00 01256
00 01257
00 01258
00 01259
00 01260
00 01261
00 01262
00 01263
00 01264
00 01265
00 01266
00 01267
00 01268
00 01269
00 01270
00 01271
00 01272
00 01273
00 01274
00 01275
00 01276
00 01277
00 01278
00 01279
00 01280
00 01281
00 01282
00 01283
00 01284
00 01285
00 01286
00 01287
00 01288
00 01289
00 01290
00 01291
00 01292
00 01293
00 01294
00 01295
00 01296
00 01297
00 01298
00 01299
00 01300
00 01301
00 01302
00 01303
00 01304
00 01305
00 01306
00 01307
00 01308
00 01309
00 01310
```



```
1311 *****00 01311
1312 * TITLE: COBOL RUNTIME DECODER * 00 01312
1313 * * 00 01313
1314 * PURPOSE: DECODE A COBOL RUN-TIME INSTRUCTION BY GETTING THE * 00 01314
1315 * INSTRUCTION AND JUMPING TO THE APPROPRIATE * 00 01315
1316 * INSTRUCTION PROCESSOR * 00 01316
1317 * CALLING SEQUENCE: * 00 01317
1318 * BSC 021 * 00 01318
1319 * DATA INSTR 0 PROCESSOR ADDR * 00 01319
1320 * DATA INSTR 1 PROCESSOR ADDR * 00 01320
1321 * DATA INSTR 32 PROCESSOR ADDR * 00 01321
1322 *****00 01322
1323 * 00 01323
1324 * COBOL RUN-TIME DECODER 00 01324
1325 * 00 01325
1326 * 00 01326
1327 * 00 01327
1328 * 00 01328
1329 * OPR = RO = 0 00 01329
1330 * 00 01330
1331 DECD GEN /N(DECD1),12(ASPEC),22(AZERO),14(TRNA),15(LOG), 00 01331
1332 C17(GPROUT),24(RO),13(OPROUT) 00 01332
1333 * 00 01333
1334 * SHIFT COUNTER =-5; WAIT FOR FETCH COMPLETE 00 01334
1335 * 00 01335
1336 DECD1 GMSK /N(DECD2),11(LIT),15(X'D004),14(TRNB),13(SCOUT), 00 01336
1337 C6(SPEC),10(WAITMD) 00 01337
1338 * 00 01338
1339 * OPCODE WORD TO R1 00 01339
1340 * 00 01340
1341 DECD2 GEN /N(DECD3),11(BSSPEC),23(MIR),14(TRNB),15(LOG), 00 01341
1342 C17(GPROUT),24(R1) 00 01342
1343 * 00 01343
1344 * 00 01344
1345 * LEFT SHIFT OPR AND R1 WITH R1(15) TO OPR(0); COUNT DOWN, LOOP, 00 01345
1346 * AND EXIT WHEN 5 BITS SHIFTED 00 01346
1347 * 00 01347
1348 DECD3 GEN /T(DECD3,DECD4),5(FT),7(SFTC),13(INCSC), EXIT ON DONE 00 01348
1349 C12(ASGPR),24(R1),14(SHFA), SHIFT R1 00 01349
1350 C18(SHFTOP),20(LFT),21(1),17(GPROUT) SHIFT OPR w/C1ROO 01350
1351 * 00 01351
1352 * CALCULATE INSTRUCTION PROCESSOR VECTOR POINTER AND FETCH 00 01352
1353 * 00 01353
1354 DECD4 GEN /N(DECD5),12(ASP),11(BSSPEC),23(OPR),14(ADD), 00 01354
1355 C10(OF$ALU),6(MEMC) 00 01355
1356 * 00 01356
1357 * WAIT FOR FETCH; INCREMENT PSEUDO=P 00 01357
1358 * 00 01358
1359 DECD5 GEN /N(DECD6),6(SPEC),10(WAITMD) 00 01359
1360 * 00 01360
1361 * SET P TO INSTRUCTION PROCESSOR AND RETURN TO PIPELINE 00 01361
1362 * 00 01362
1363 * ORG X'35 00 01363
1364 DECD6 GEN /P(SS2M),7(PJMP$),TFO,11(BSSPEC),23(MIR),14(TRNB), 00 01364
1365 C15(LOG),13(POUT),10(1F$ALU),6(MEMC$) 00 01365
1366 * EJECE 00 01366
1367 *****00 01367
```


1368	*			*00 01368
1369	*	TITLE:	DECIMAL OPERATION SETUP	*00 01369
1370	*			*00 01370
1371	*	PURPOSE:	THIS SECTION PERFORMS ALL SETUP AND	*00 01371
1372	*		PRE-PROCESSING NECESSARY FOR THE DECIMAL	*00 01372
1373	*		OPERATORS.	*00 01373
1374	*			*00 01374
1375	*	DESCRIPTION:	THE FIRST THREE WORDS OF THE PARAMETER	*00 01375
1376	*		BLOCK ARE FETCHED AND USED TO OBTAIN	*00 01376
1377	*		BYTE POINTERS TO OPERANDS A AND B. THE	*00 01377
1378	*		LENGTHS OF THE OPERANDS ARE ALSO SETUP	*00 01378
1379	*		IN THE APPROPRIATE REGISTERS. THE CORRECT	*00 01379
1380	*		SIGN OF EACH OPERAND IS DETERMINED BY	*00 01380
1381	*		CHECKING THE FIRST WORD OF THE PARAMETER	*00 01381
1382	*		BLOCK AND THEN FETCHING THE SIGN FROM	*00 01382
1383	*		MEMORY IF NECESSARY.	*00 01383
1384	*			*00 01384
1385	*		IF A FOURTH PARAMETER BLOCK WORD IS	*00 01385
1386	*		PRESENT, IT IS FETCHED AND THE SHIFT	*00 01386
1387	*		COUNTS OBTAINED FOR THE FIRST TWO	*00 01387
1388	*		OPERANDS. RIGHT SHIFTING IS HANDLED BY	*00 01388
1389	*		ADJUSTING THE OPERAND LENGTH. LEFT SHIFT	*00 01389
1390	*		IS HANDLED BY SETTING THE COUNT IN THE	*00 01390
1391	*		APPROPRIATE REGISTER FOR LATER USE BY	*00 01391
1392	*		THE FETCH BYTE ROUTINE.	*00 01392
1393	*			*00 01393
1394	*		IF THE FIFTH AND SIXTH PARAM BLOCK WORDS	*00 01394
1395	*		ARE PRESENT, THE DESTINATION BYTE PTR	*00 01395
1396	*		AND LENGTH ARE FETCHED AND BUILT. IN THIS	*00 01396
1397	*		CASE, LEFT SHIFTING IS HANDLED BY ZERO	*00 01397
1398	*		FILL AND LENGTH ADJUSTMENT. RIGHT SHIFTING	*00 01398
1399	*		IS HANDLED BY SETTING UP THE COUNT IN THE	*00 01399
1400	*		APPROPRIATE REGISTER FOR USE BY THE STORE	*00 01400
1401	*		BYTE SUBROUTINE.	*00 01401
1402	*			*00 01402
1403	*		WHEN ALL PROCESSING OF THE PARAMETER BLOCK	*00 01403
1404	*		IS FINISHED, CONTROL IS TRANSFERED TO THE	*00 01404
1405	*		APPROPRIATE SERVICE ROUTINE TO PERFORM	*00 01405
1406	*		THE ACTUAL DECIMAL OPERATION.	*00 01406
1407	*			*00 01407
1408	*	ENTRY POINT:	DE00	*00 01408
1409	*			*00 01409
1410	*	CALLING SEQUENCE:	ENTERED THROUGH BCS ENTRY POINT;	*00 01410
1411	*		EXITS TO DECIMAL OPERATION PROCESSOR;	*00 01411
1412	*			*00 01412
1413	*	EXCEPTION CONDITIONS:	IF AN ERROR IS DISCOVERED IN THE PARAMETER	*00 01413
1414	*		BLOCK, AN EXIT IS MADE WITH AN ERROR	*00 01414
1415	*		CODE IN CONDT. THIS EXIT WILL OCCUR JUST	*00 01415
1416	*		BEFORE THE DECIMAL OPERATION SO THAT THE	*00 01416
1417	*		PARAM BLOCK POINTER, R2, WILL BE ADVANCED	*00 01417
1418	*		AS NECESSARY.	*00 01418
1419	*			*00 01419
1420	*	REGISTER USAGE:	R0- POINTER TO TEMP AREA	*00 01420
1421	*		R1- PTR TO CONDT AND BASE REG BLOCK	*00 01421
1422	*		R2- PTR TO PARAMETER BLOCK	*00 01422
1423	*		R3- OP A SIGN	*00 01423
1424	*		R4- OP B SIGN	*00 01424

```
1425 * R5= SAVED PTR TO PARAMETER BLOCK *00 01425
1426 * R6= OP B BYTE PTR *00 01426
1427 * R7= LEFT BYTE; OP B LEFT SHIFT COUNT *00 01427
1428 * RIGHT BYTE; OP B LENGTH *00 01428
1429 * R8,R9 = TEMPS *00 01429
1430 * RA= OP A BYTE PTR *00 01430
1431 * RB= OP A LEFT SHIFT COUNT AND LENGTH *00 01431
1432 * RC= FLAGS RESULT IN THIRD OP;(ALSO ROUND *00 01432
1433 * CARRY FOR DR00). *00 01433
1434 * RD= RESULT BYTE PTR *00 01434
1435 * RE= RESULT RIGHT SHIFT COUNT AND LENGTH *00 01435
1436 * RF= DISPLACEMENT FROM RESULT BYTE PTR TO *00 01436
1437 * THE RESULT SIGN. *00 01437
1438 * OPR AND P = TEMPS *00 01438
1439 * QUOS= RESULT SIGNED *00 01439
1440 * OVFL= 1) FLAGS PARAMETER BLOCK ERROR. *00 01440
1441 * 2) FLAGS ROUNDING. *00 01441
1442 * SHFT = FLAGS MOVE OPERATION *00 01442
1443 * *00 01443
1444 * *00 01444
1445 * *00 01445
1446 * *00 01446
1447 ***** *00 01447
1448 * 00 01448
1449 * FINISH SAVE OF P 00 01449
1450 * RESET SHIFT FLAG 00 01450
1451 * 00 01451
1452 * ORG X'100 00 01452
1453 DE01 GEN /T(DTRO0,DE02),5(TT),7(GPRS),12(ASP), 00 01453
1454 C14(TRNA),15(LOG),6(SPEC),10(WAITMD),VF1,24(R1) 00 01454
1455 * 00 01455
1456 * SAVE PARAM BLOCK ADDR AND RESET QUOS AND OVFL. 00 01456
1457 * START FETCH OF FIRST PARAM BLOCK WORD; 00 01457
1458 * 00 01458
1459 * ORG X'14 00 01459
1460 DE02 GEN /N(DE03),6(MEMC),10(IFSALU),11(BSGPR),23(R2), 00 01460
1461 C14(TRNB),15(LOG),17(GPROUT),24(R5), 00 01461
1462 C7(ROVFL),SCO,WF1 00 01462
1463 * 00 01463
1464 * START FETCH OF SECOND PARAM BLOCK WORD; 00 01464
1465 * 00 01465
1466 DE03 GEN /N(DE04),6(MEMC),10(OF$ALU),12(ASGPR),24(R2), 00 01466
1467 C14(INCA),16(CRY1),17(GPROUT) 00 01467
1468 * 00 01468
1469 * CONVERT SECOND PARAM BLK WORD INTO OP A BYTE PTR 00 01469
1470 * 00 01470
1471 DE04 GMSK /P(DB00+P),10(STACK),16(PUSH),15(P+DE05), 00 01471
1472 CLB3,TFO,SFO,7(IBR$1) 00 01472
1473 * 00 01473
1474 * START FETCH OF THIRD PARAM BLOCK WORD 00 01474
1475 * 00 01475
1476 DE05 GEN /N(DE06),6(MEMC),10(OF$ALU),12(ASGPR),24(R2), 00 01476
1477 C14(INCA),16(CRY1),17(GPROUT) 00 01477
1478 * 00 01478
1479 * SAVE OP A BYTE PTR 00 01479
1480 * 00 01480
1481 DE06 GEN /N(DE07),12(ASP),14(TRNA),15(LOG), 00 01481
```

1482		C17(GPROUT),24(RA)	00 01482
1483	*		00 01483
1484	*	CONVERT THIRD PARAM BLOCK WORD INTO OP B BYTE PTR	00 01484
1485	*		00 01485
1486	DE07	GMSK /P(DB00+P),10(STACK),16(PUSH),15(P+DE08),	00 01486
1487		CLB3,TFO,SFO	00 01487
1488	*		00 01488
1489	*	SAVE OP B BYTE PTR	00 01489
1490	*	ALSO TEST IF ERROR OCCURED IN EITHER CONVERSION	00 01490
1491	*		00 01491
1492	DE08	GEN /T(DE09,DE25),7(OVFL),5(FT),12(ASP),14(TRNA),15(LOG),	00 01492
1493		C17(GPROUT),24(R6)	00 01493
1494	*		00 01494
1495	*	GET OP A LENGTH FROM FIRST PARAM BLOCK WORD	00 01495
1496	*	DELETE A STACK ENTRY TO INSURE STACK IS EMPTY	00 01496
1497	*		00 01497
1498		ORG X'4F	00 01498
1499	DE09	GMSK /P(DE10+P),11(MSK),15(X'FC1F),14(TRNB),13(OPROUT),	00 01499
1500		C10(STACK),AK4,TFO,SFO	00 01500
1501	*		00 01501
1502	*	LOAD SHIFT COUNTER FOR OP A LENGTH RIGHT JUSTIFICATION	00 01502
1503	*		00 01503
1504		ORG DE08+1	00 01504
1505	DE10	GMSK /N(DE11),11(LIT),15(X'2),	00 01505
1506		C14(TRNB),13(SCOUT)	00 01506
1507	*		00 01507
1508	*	RIGHT JUSTIFY OP LENGTH IN OPR LEFT BYTE	00 01508
1509	*	ALSO INIT RESULT IN OP C FLAG	00 01509
1510	*		00 01510
1511	DE11	GEN /T(DE12,DE11),5(TT),7(SFTC),18(SHFTOP),20(LFT),XF3,	00 01511
1512		C14(ONES),17(GPROUT),24(RC),13(INCSC)	00 01512
1513	*		00 01513
1514	*	SET OP A LENGTH	00 01514
1515	*		00 01515
1516	DE12	GEN /N(DE13),11(BSSPEC),23(OLSE),14(TRNB),15(LOG),	00 01516
1517		C17(GPROUT),24(RB)	00 01517
1518	*		00 01518
1519	*	SETUP OP B LENGTH	00 01519
1520	*		00 01520
1521	DE13	GMSK /N(DE14),11(MSK),15(X'FFE0),14(TRNB),13(OPROUT)	00 01521
1522		ORG X'B	00 01522
1523	DE14	GEN /N(DE15),11(BSSPEC),23(ORZF),14(TRNB),15(LOG),	00 01523
1524		C17(GPROUT),24(R7)	00 01524
1525	*		00 01525
1526	*	INIT OP A AND OP B SIGNS	00 01526
1527	*		00 01527
1528	DE15	GEN /*,14(ZERO),15(LOG),17(GPROUT),24(R3)	00 01528
1529		GEN /*,14(ZERO),15(LOG),17(GPROUT),24(R4)	00 01529
1530	*		00 01530
1531	*	INIT RESULT BYTE PTR TO OP A BYTE PTR	00 01531
1532	*		00 01532
1533		GEN /N(DE16),11(BSGPR),23(RA),14(TRNB),15(LOG),	00 01533
1534		C17(GPROUT),24(RD)	00 01534
1535	*		00 01535
1536	*	INIT RESULT LENGTH TO OP A LENGTH	00 01536
1537	*	ALSO TEST IF RESULT (IE OP A) IS SIGNED	00 01537
1538	*		00 01538

```
1539      ORG      DE01+1                      00 01539
1540 DE16  GEN      /F(DE17),FSF,2(1),11(B$GPR),23(RB), 00 01540
1541          C14(TRNB),15(LOG),17(GPROUT),24(RE)      00 01541
1542      *                      00 01542
1543      *      RESULT SIGNED SO FLAG IT BY SETTING QUOS 00 01543
1544      *                      00 01544
1545      ORG      X'36                      00 01545
1546 DE17  GEN      /N(DE18),14(ONES),SC0,WFI      00 01546
1547      *                      00 01547
1548      *      INIT DISPLACEMENT TO RESULT SIGN      00 01548
1549      *                      00 01549
1550      ORG      DE16+1                      00 01550
1551 DE18  GEN      /N(DE19),11(B$GPR),23(RB),14(TRNB),15(LOG), 00 01551
1552          C17(GPROUT),24(RF)                  00 01552
1553      *                      00 01553
1554      *      SETUP BYTE PTR TO FETCH OP A SIGN      00 01554
1555      *                      00 01555
1556 DE19  GEN      /N(DE20),12(AS$PEC),22(AZERO),11(B$GPR),23(RA), 00 01556
1557          C14(INCB),16(CRY1),17(GPROUT),24(RB)    00 01557
1558      *                      00 01558
1559      *      FETCH SIGN OF OP A                      00 01559
1560      *                      00 01560
1561 DE20  GMSK      /P(DS00+P),10(STACK),15(DE21+P),TF0,SFO,LB3, 00 01561
1562          C12(AS$GPR),16(RB),14(TRNA),13(OPROUT)  00 01562
1563      *                      00 01563
1564      *      SAVE OP A SIGN                          00 01564
1565      *                      00 01565
1566 DE21  GEN      /N(DE22),12(ASP),14(TRNA),15(LOG), 00 01566
1567          C17(GPROUT),24(R3)                  00 01567
1568      *                      00 01568
1569      *      IF OPERATION IS A MOVE, SET SHFT TO FLAG IT. 00 01569
1570      *      ALSO, IF MOVE, SKIP OP B SIGN CHECK AND SET QUOS ACCORDINGLY. 00 01570
1571      *                      00 01571
1572      ORG      X'37                      00 01572
1573 DE22  GMSK      /N(DE22D),11(MSK),15(X'1FFF),14(TRNB),13(POUT) GET OP 00 01573
1574      ORG      DE21+1                      00 01574
1575 DE22D GMSK      /*,11(LIT),15(X'9FFF),12(ASP),14(EOR),      OP=3? 00 01575
1576          CTF0,SFO,7(SS$ALU)                  00 01576
1577      GEN      /T(DE22N,DE22H),5(FT),7(ALUZ)      00 01577
1578 DE22H GEN      /F(DE22K),FSE,2(1),24(RC),VF1,14(ZERO),15(LOG),SC0,WFI 00 01578
1579      ORG      X'4A                      00 01579
1580 DE22K GEN      /N(DE22K3),14(ONES),SC0,WFI      00 01580
1581      *                      00 01581
1582      *      IF MOVE, DO SPECIAL REGISTER SETUP      00 01582
1583      *                      00 01583
1584      ORG      X'4B                      00 01584
1585 DE22K3 GEN      /N(DE22K5),11(B$GPR),23(R6), 00 01585
1586          C14(TRNB),15(LOG),17(GPROUT),24(RD)    00 01586
1587      ORG      X'159                      00 01587
1588 DE22K5 GEN      /N(DE22K7),11(B$GPR),23(R7),14(TRNB),15(LOG), 00 01588
1589          C17(GPROUT),24(RE)                  00 01589
1590      ORG      X'15B                      00 01590
1591 DE22K7 GEN      /N(DE25),11(B$GPR),23(R7), 00 01591
1592          C14(TRNB),15(LOG),17(GPROUT),24(RF)    00 01592
1593      *                      00 01593
1594      *      SETUP BYTE PTR FOR OP B SIGN FETCH;      00 01594
1595      *      CHECK IF OP B SIGNED                  00 01595
```

1596	*			00 01596
1597		ORG	DE22H+1	00 01597
1598	DE22N	GEN	/F(DE23),FSD,2(2),12(ASPEC),22(AZERO),	00 01598
1599			C11(BSGPR),23(R6),14(INCB),16(CRY1),	00 01599
1600			C17(GPROUT),24(R8)	00 01600
1601	*			00 01601
1602	*			00 01602
1603	**		FETCH OP B SIGN	00 01603
1604	*		(THIS MICRO ADDR MUST BE EVEN)	00 01604
1605		ORG	X'4C	00 01605
1606	DE23	GEN	/N(DE23P),12(ASGPR),24(R7),14(TRNA),15(LOG),	00 01606
1607			C13(OPROUT)	00 01607
1608		ORG	X'21	00 01608
1609	DE23P	GMSK	/P(DS00+P),10(STACK),16(PUSH),15(DE24+P),	00 01609
1610			CLB3,SFO,TFO	00 01610
1611	*			00 01611
1612	*			00 01612
1613	*		SAVE OP B SIGN	00 01613
1614		ORG	DE22N+1	00 01614
1615	DE24	GEN	/N(DE25),12(ASP),14(TRNA),15(LOG),	00 01615
1616			C17(GPROUT),24(R4)	00 01616
1617	*			00 01617
1618	*			00 01618
1619	*		ADVANCE PARAM BLOCK PTR;	00 01619
1620	*		CHECK IF FOURTH PARAM BLOCK WORD PRESENT	00 01620
1621		ORG	X'4E	00 01621
1622	DE25	GEN	/F(DE40),FSF,2(2),12(ASGPR),24(R2),	00 01622
1623			C14(INCA),16(CRY1),17(GPROUT)	00 01623
1624	*			00 01624
1625	*		START FETCH OF FOURTH PARAM BLOCK WORD	00 01625
1626	*			00 01626
1627		ORG	X'42	00 01627
1628	DE26	GEN	/N(DE27),6(MEMC),10(IFSALU),12(ASGPR),24(R2),	00 01628
1629			C14(TRNA),15(LOG)	00 01629
1630	*			00 01630
1631	*		START FETCH OF FIFTH PARAM BLOCK WORD	00 01631
1632	*			00 01632
1633		ORG	DE24+1	00 01633
1634	DE27	GEN	/N(DE28),6(MEMC),10(IFSALU),12(ASGPR),24(R2),	00 01634
1635			C14(INCA),16(CRY1),17(GPROUT)	00 01635
1636	*			00 01636
1637	*		PUT FOURTH PARAM BLOCK WORD INTO I AND GPR	00 01637
1638	*			00 01638
1639	DE28	GEN	/N(DE29),11(BSSPEC),23(MIR),14(TRNB),15(10G),	00 01639
1640			C17(GPROUT),24(R8),TFO,SFO,7(IBR\$1)	00 01640
1641	*			00 01641
1642	*		ALIGN OP A SHIFT COUNT	00 01642
1643	*			00 01643
1644	DE29	GEN	/N(DE30),12(ASGPRL),24(R8),14(SHFA),17(GPROUT)	00 01644
1645	*			00 01645
1646	*		ALIGN OP A SHIFT COUNT AND PLACE IN OPR;	00 01646
1647	*		CHECK IF RIGHT OR LEFT SHIFT	00 01647
1648	*			00 01648
1649	DE30	GMSK	/F(DE31),FSF,2(1),12(ASGPRL),16(R8),	00 01649
1650			C11(LIT),15(X'EUFF),14(AND),13(OPROUT)	00 01650
1651	*			00 01651
1652	*		RIGHT SHIFT, SO ADJUST OP A LENGTH ACCORDINGLY;	00 01652

1653	*			00 01653
1654		ORG	*+1	00 01654
1655	DE31	GEN	/N(DE32),12(ASGPR),24(RB),11(B\$SPEC),23(OLSE),	00 01655
1656			C14(SUB),16(CRY1),17(GPROUT),TFO,SFO,7(S\$ALU)	00 01656
1657	*			00 01657
1658	*		LEFT SHIFT,SO SAVE SHIFT COUNT FOR LATER USE	00 01658
1659	**		(THIS MICRO MUST FOLLOW DE31)	00 01659
1660	*			00 01660
1661		GEN	/N(DE34),12(ASGPR),24(RB),11(B\$SPEC),23(OPR),	00 01661
1662			C14(OR),17(GPROUT)	00 01662
1663	*			00 01663
1664	*		IF OP A LENGTH AFTER RIGHT SHIFT ADJUSTMENT IS NEGATIVE,	00 01664
1665	*		RESET IT TO ZERO.	00 01665
1666	*			00 01666
1667		ORG	DE30+1	00 01667
1668	DE32	GEN	/T(DE34,DE33),5(FT),7(ALUS)	00 01668
1669		ORG	DE31+2	00 01669
1670	DE33	GEN	/N(DE34),14(ZERO),15(LOG),17(GPROUT),24(RB)	00 01670
1671	*			00 01671
1672	*		IF OPERATION IS MOVE, SKIP FURTHER SETUP!	00 01672
1673	*			00 01673
1674	DE34	GEN	/T(DE34N,DE34M),5(FT),7(SHFT)	00 01674
1675	*			00 01675
1676	*		PERFORM SHIFT/ROUND INITIALIZATION FOR MOVE	00 01676
1677	*			00 01677
1678		ORG	X'3C	00 01678
1679	DE34M	GMSK	/P(DX50+P),10(STACK),16(PUSH),15(DE40+P),	00 01679
1680			CLB3,TFO,SFO	00 01680
1681	*			00 01681
1682	*		GET OP B SHIFT COUNT;	00 01682
1683	*		CHECK IF LEFT OR RIGHT SHIFT	00 01683
1684	*			00 01684
1685		ORG	X'38	00 01685
1686	DE34N	GMSK	/F(DE35),FSE,2(1),11(MSK),15(X'FFED),	00 01686
1687			C14(TRNB),13(OPROUT)	00 01687
1688	*			00 01688
1689	*		RIGHT SHIFT, SO ADJUST OP B LENGTH ACCORDINGLY	00 01689
1690	*			00 01690
1691		ORG	DE34+1	00 01691
1692	DE35	GEN	/N(DE36),12(ASGPR),24(R7),11(B\$SPEC),23(OPR),	00 01692
1693			C14(SUB),16(CRY1),17(GPROUT),TFO,SFO,7(S\$ALU)	00 01693
1694	*			00 01694
1695	*		LEFT SHIFT, SO SAVE SHIFT COUNT FOR LATER USE	00 01695
1696	**		(THIS MICRO MUST FOLLOW DE35)	00 01696
1697	*			00 01697
1698		GEN	/N(DE38),12(ASGPR),24(R7),11(B\$SPEC),23(ORLZ),	00 01698
1699			C14(OR),17(GPROUT)	00 01699
1700	*			00 01700
1701	*		IF OP B LENGTH NEGATIVE AFTER RIGHT SHIFT ADJUSTMENT,	00 01701
1702	*		RESET IT TO ZERO.	00 01702
1703	*			00 01703
1704	DE36	GEN	/T(DE37,DE38),5(TT),7(ALUS)	00 01704
1705	DE37	GEN	/N(DE38),14(ZERO),15(LOG),17(GPROUT),24(R7)	00 01705
1706	*			00 01706
1707	*		CHECK IF FIFTH PARAM BLOCK WORD IS PRESENT	00 01707
1708	*			00 01708
1709	DE38	GEN	/F(DE40),FSE,2(4),6(SPEC),10(WAITMD)	00 01709

```
1710 * 00 01710
1711 * FIFTH AND SIXTH PARAM BLOCK WORDS ARE PRESENT SO PERFORM SFTUP 00 01711
1712 * 00 01712
1713 ORG X'44 00 01713
1714 DE39 GEN /N(DX01),6(MEMC),10(OF$ALU),12(ASGPR),24(R2), 00 01714
1715 C14(INCA),16(CRY1),17(GPROUT) 00 01715
1716 * 00 01716
1717 * SAVE RD IN TEMP AREA FOR MULTIPLY 00 01717
1718 * IF MOVE, EXIT TO MOVE ROUTINE 00 01718
1719 * 00 01719
1720 ORG X'40 00 01720
1721 DE40 GMSK /N(DE40H),12(ASGPR),16(RD),11(LIT),15(X'FFAF), 00 01721
1722 C14(ADD),6(MEMC),10(OS$ALU) 00 01722
1723 ORG X'3A 00 01723
1724 DE40H GEN /T(DE40N,EX3D),5(FT),7(OVFL), 00 01724
1725 C12(ASGPR),24(RD),14(TRNA),15(LOG),6(SPEC),10(WAITMD) 00 01725
1726 * 00 01726
1727 * SETUP OP A BYTE PTR FOR LEADING ZERO SUPPRESSION; 00 01727
1728 * CHECK IF PARAM ERROR HAS BEEN DETECTED. 00 01728
1729 * 00 01729
1730 ORG X'53 00 01730
1731 DE40N GEN /T(DY00,DE41),5(TT),7(SHFT), 00 01731
1732 C11(B$GPR),23(RA), 00 01732
1733 C14(TRNB),15(LOG),17(GPROUT),24(R8) 00 01733
1734 * 00 01734
1735 * SUPPRESS OP A LEADING ZEROES 00 01735
1736 * 00 01736
1737 ORG X'32 00 01737
1738 DE41 GMSK /P(DL00+P),10(STACK),15(DE42+P),LB3,TFO,SFO, 00 01738
1739 C12(ASGPR),16(RB),14(TRNA),13(OPROUT) 00 01739
1740 * 00 01740
1741 * UPDATE OP A LENGTH FOR ZERO SUPPRESS 00 01741
1742 * CHECK IF NEW LENGTH ZERO 00 01742
1743 * 00 01743
1744 ORG DE38+1 00 01744
1745 DE42 GEN /T(DE44,DE43),5(FT),7(SFTC),12(ASGPR),24(RB), 00 01745
1746 C11(B$SPEC),23(OPR),14(SUB),16(CRY1),17(GPROUT) 00 01746
1747 * 00 01747
1748 * NEW LENGTH ZERO SO CLEAR SHIFT COUNT ALSO 00 01748
1749 * 00 01749
1750 DE43 GEN /N(DE44),14(ZERO),15(LOG),17(GPROUT),24(RB) 00 01750
1751 * 00 01751
1752 * ADJUST OP A BYTE POINTER FOR ZERO SUPPRESS; 00 01752
1753 * ALSO CHECK IF ROUNDING REQUESTED. 00 01753
1754 * 00 01754
1755 DE44 GEN /F(DE45),FSF,2(2),12(ASGPR),24(RA), 00 01755
1756 C11(B$SPEC),23(OPR),14(ADD),17(GPROUT) 00 01756
1757 * 00 01757
1758 * REFETCH FIRST PARAM BLOCK WORD; 00 01758
1759 * FLAG WHETHER ROUNDING OR NOT THRU OVERFLOW. 00 01759
1760 * 00 01760
1761 DE45 GEN /N(DE46),6(MEMC),10(IFSALU),12(ASGPR),24(R5), 00 01761
1762 C14(TRNA),15(LOG),7(ROVFL) 00 01762
1763 ORG *+1 ALLIGN FOR FSEL 00 01763
1764 GEN /N(DE46),6(MEMC),10(IFSALU),12(ASGPR),24(R5), 00 01764
1765 C14(TRNA),15(LOG),7(SOVFL) 00 01765
1766 * 00 01766
```



```
1767 *      SETUP OP B BYTE PTR FOR ZERO SUPPRESS      00 01767
1768 *
1769 *      ORG      *-2      00 01768
1770 DE46 GEN      /N(DE47),11(B$GPR),23(R6),14(TRNB),15(LOG), 00 01769
1771          C17(GPROUT),24(R8),6(SPEC),10(WAITMD)      00 01770
1772 *      ORG      *+1      00 01771
1773 *
1774 *      SUPPRESS OP B LEADING ZEROES      00 01772
1775 *      ALSO SETUP FIRST PARAM BLOCK WORD.      00 01773
1776 *
1777 DE47 GEN      /N(DE47P),12(A$GPR),24(R7),14(TRNA),15(LOG), 00 01774
1778          C13(OPROUT)      00 01775
1779 *      ORG      X'45      00 01776
1780 DE47P GMSK     /P(DLOO+P),10(STACK),16(PUSH),15(DE48+P),7(1BR$1), 00 01777
1781          CLB3,SFO,TFO      00 01778
1782 *      ORG      DE47+1      00 01779
1783 *
1784 *      UPDATE OP B LENGTH FOR ZERO SUPPRESS;      00 01780
1785 *      CHECK IF NEW LENGTH ZERO.      00 01781
1786 *
1787 DE48 GEN      /T(DE49,DE50),5(TT),7(SFTC),12(A$GPR),24(R7), 00 01782
1788          C11(B$SPEC),23(OPR),14(SUB),16(CRY1),17(GPROUT) 00 01783
1789 *
1790 *      NEW LENGTH ZERO SO CLEAR SHIFT COUNT ALSO      00 01784
1791 *
1792 DE49 GEN      /N(DE50),14(ZERO),15(LOG),17(GPROUT),24(R7) 00 01785
1793 *
1794 *      ADJUST OP B BYTE PTR FOR ZERO SUPPRESS;      00 01786
1795 *      BRANCH TO OPERATION PROCESSOR.      00 01787
1796 *
1797 DE50 GEN      /F(DA00),FSF,2(X'C),MT1,12(A$GPR),24(R6), 00 01788
1798          C11(B$SPEC),23(OPR),14(ADD),17(GPROUT)      00 01789
1799 *      EJEC      00 01790
1800 *****00 01791
1801 *
1802 *      TITLE:      ADD/SUBTRACT PROCESSOR      00 01792
1803 *
1804 *      PURPOSE:      THIS ROUTINE PERFORMS ADDITION OR      00 01793
1805 *                      SUBTRACTION OF TWO DECIMAL DIGIT STRINGS. 00 01794
1806 *
1807 *      DESCRIPTION:  IF OPERAND A IS BOTH THE DESTINATION AND      00 01795
1808 *                      LEFT SHIFTED ON INPUT, IT IS MOVED TO      00 01796
1809 *                      THE TEMP AREA TO AVOID OVERLAY DURING      00 01797
1810 *                      THE OPERATION. NEXT, THE RESULT SIGN IS      00 01798
1811 *                      DETERMINED BY CHECKING THE SIGN AND      00 01799
1812 *                      ABSOLUTE VALUE OF THE TWO OPERANDS.      00 01800
1813 *                      FINALLY, THE OPERAND BYTES ARE FETCHED.      00 01801
1814 *                      ADDED OR SUBTRACTED, AND THE RESULT      00 01802
1815 *                      STORED. THIS IS DONE ONE BYTE AT A TIME,      00 01803
1816 *                      FROM LEAST SIGNIFICANT TO MOST SIGNIFICANT. 00 01804
1817 *
1818 *      ENTRY POINT:  ADD - EA00      00 01805
1819 *                      SUBTRACT - ES00      00 01806
1820 *
1821 *      CALLING SEQUENCE: RECEIVES CONTROL FROM SETUP ROUTINE, DE00. 00 01807
1822 *                      EXITS TO MOVE ROUTINE FOR ZERO FILL OF      00 01808
1823 *                      REST OF RESULT.      00 01809
```



```
1824 * *00 01824
1825 * EXCEPTION CONDITIONS: SUBTRACT IS IMPLEMENTED BY COMPLEMENTING *00 01825
1826 * THE SIGN OF OP B AND PERFORMING AN ADD. *00 01826
1827 * *00 01827
1828 * REGISTER USAGE: SAME AS FOR DEOD EXCEPT: *00 01828
1829 * R3= SCRATCH *00 01829
1830 * R5= CARRY *00 01830
1831 * SHFT = FLAGS ADD(0) OR SUBTRACT(1) *00 01831
1832 * RC= ROUND CARRY (FOR DR00) *00 01832
1833 *.....*00 01833
1834 * ORG *+1 ***** 00 01834
1835 DA01 EQU * SAVE LOCATION 00 01835
1836 * 00 01836
1837 * SUBTRACT 00 01837
1838 * COMPLEMENT OP B SIGN AND BRANCH TO ADD 00 01838
1839 * 00 01839
1840 * ORG X'24 PLACE FOR FIELD SEL. 00 01840
1841 * GMSK /N(DA005),11(LIT),15(X'FF52),14(TRNB),13(OPROUT) 00 01841
1842 * ORG X'33 00 01842
1843 DA005 GEN /N(DA00),12(ASGPR),24(R4),11(BSSPEC),23(OPR), 00 01843
1844 * C14(EOR),15(LOG),17(GPROUT) 00 01844
1845 * 00 01845
1846 * ADD 00 01846
1847 * CHECK IF RESULT IN OP A 00 01847
1848 * 00 01848
1849 * ORG X'20 PLACE FOR FIELD SEL. 00 01849
1850 DA00 GEN /N(DA01),12(ASGPR),24(RC),14(TRNA),15(LOG), 00 01850
1851 * CTFO,SFO,7(SSALU) 00 01851
1852 * 00 01852
1853 * IF RESULT NOT IN OP A, OP A DOES NOT HAVE TO BE MOVED TO TEMP 00 01853
1854 * 00 01854
1855 * ORG DA01 RESTORE LOC. 00 01855
1856 DA01 GEN /T(DA02,DA05),5(FT),7(ALUZ) 00 01856
1857 * 00 01857
1858 * CHECK IF OP A LEFT SHIFTED 00 01858
1859 * 00 01859
1860 DA02 GMSK /N(DA03),12(ASGPR),16(RB),11(LIT),15(X'80FF), 00 01860
1861 * C14(AND),TF0,SFO,7(SSALU) 00 01861
1862 * 00 01862
1863 * IF OP A LEFT SHIFTED AND GETS RESULT, MOVE TO TEMP TO AVOID 00 01863
1864 * OVERLAY. 00 01864
1865 * 00 01865
1866 DA03 GEN /T(DA04,DA05),5(FT),7(ALUZ), 00 01866
1867 * C14(ZERO),15(LOG),17(GPROUT),24(RC) 00 01867
1868 DA04 GMSK /N(DM10),12(ASGPR),16(RB),11(LIT),15(X'FF80),14(AND), 00 01868
1869 * C13(OPROUT) 00 01869
1870 * 00 01870
1871 * CHECK IF OPERANDS HAVE THE SAME SIGN 00 01871
1872 * (THIS ADDR MUST BE EVEN) 00 01872
1873 * 00 01873
1874 DA05 GEN /N(DA06),12(ASGPR),24(R3),11(BSGPR),23(R4), 00 01874
1875 * C14(EOR),15(LOG),TF0,SFO,7(SSALU) 00 01875
1876 * 00 01876
1877 * INIT SHFT TO FLAG ADD: 00 01877
1878 * IF OPERANDS HAVE SAME SIGN, RESULT SIGN = SIGN OP A. 00 01878
1879 * 00 01879
1880 DA06 GEN /T(DA18,DA07),5(TT),7(ALUZ),24(R0),VF1 00 01880
```

1881	*			00 01881
1882	*	SET SHFT TO FLAG SUBTRACT. (IE OPS HAVE DIF. SIGN).		00 01882
1883	*	(THIS ADDR MUST BE EVEN)		00 01883
1884	*			00 01884
1885	DA07	GEN	/N(DA08),14(ONES),17(GPROUT),24(R5)	00 01885
1886	DA08	GEN	/N(DA09),24(R5),VF1	00 01886
1887	*			00 01887
1888	*	COMPARE OPERANDS TO SEE WHICH HAS GREATER ABSOLUTE VALUE		00 01888
1889	*			00 01889
1890	DA09	GMSK	/P(DP00+P),10(STACK),15(DA10+P),LB3,TF0,SFO,	00 01890
1891			C12(ASGPR),16(RB),14(TRNA),13(OPROUT)	00 01891
1892	*			00 01892
1893	*	IF OP A HAS GREATER ABSOLUTE VALUE THAN OP B,		00 01893
1894	*	RESULT SIGN = SIGN OF OP A		00 01894
1895	*			00 01895
1896	DA10	GEN	/T(DA11,DA18),5(TT),7(ALUS)	00 01896
1897		ORG	*+1	00 01897
1898	*			00 01898
1899	*	SWAP OP A AND OP B SINCE OP B HAS GREATER ABS VALUE.		00 01899
1900	*			00 01900
1901	DA11	GMSK	/P(DA11S+P),10(STACK),16(PUSH),15(DA18+P),	00 01901
1902			CTFO,SFO,LB3	00 01902
1903		ORG	X'57	00 01903
1904	DA11S	GEN	/N(DA12),12(ASGPR),24(RA),14(TRNA),15(LOG),13(OPROUT)	00 01904
1905		ORG	DA11+1	00 01905
1906	DA12	GEN	/N(DA13),11(BSGPR),23(R6),14(TRNB),15(LOG),	00 01906
1907			C17(GPROUT),24(RA)	00 01907
1908	DA13	GEN	/N(DA14),11(BSSPEC),23(OPR),14(TRNB),15(LOG),	00 01908
1909			C17(GPROUT),24(R6)	00 01909
1910	DA14	GEN	/N(DA15),12(ASGPR),24(RB),14(TRNA),15(LOG),	00 01910
1911			C13(OPROUT)	00 01911
1912	DA15	GEN	/N(DA16),11(BSGPR),23(R7),14(TRNB),15(LOG),	00 01912
1913			C17(GPROUT),24(RB)	00 01913
1914	DA16	GEN	/N(DA17),11(BSSPEC),23(OPR),14(TRNB),15(LOG),	00 01914
1915			C17(GPROUT),24(R7)	00 01915
1916	DA17	GEN	/N(COMPOP),11(BSGPR),23(R4),14(TRNB),15(LOG),	00 01916
1917			C17(GPROUT),24(R3)	00 01917
1918	*			00 01918
1919	*	CLEAR ROUND CARRY		00 01919
1920	*			00 01920
1921		ORG	DA10+1	00 01921
1922	DA18	GEN	/N(DA19),14(ZERO),15(LOG),17(GPROUT),24(RC)	00 01922
1923		ORG	DA17+1	00 01923
1924	*			00 01924
1925	*	CLEAR CARRY;		00 01925
1926	*	CHECK IF RESULT SIGNED.		00 01926
1927	*			00 01927
1928	DA19	GEN	/T(DA20,DA21),5(TT),7(QUOS),14(ZERO),15(LOG),	00 01928
1929			C17(GPROUT),24(R5)	00 01929
1930	*			00 01930
1931	*	STORE SIGN OF RESULT		00 01931
1932	*			00 01932
1933		ORG	X'47	00 01933
1934	DA20	GMSK	/P(DN00+P),10(STACK),15(DA21+P),LB3,TF0,SFO,	00 01934
1935			C12(ASGPR),16(R3),14(TRNA),13(POUT)	00 01935
1936	*			00 01936
1937	*	FETCH BYTE OF OP B		00 01937

1938	*			00 01938
1939		ORG	X'48	00 01939
1940	DA21	GEN	/N(DA22),11(BSGPR),23(R6),14(TRNB),15(LOG),	00 01940
1941			C17(GPROUT),24(R8)	00 01941
1942		ORG	DA19+1	00 01942
1943	DA22	GEN	/N(DA22P),12(ASGPR),24(R7),14(TRNA),15(LOG),	00 01943
1944			C13(OPROUT)	00 01944
1945		ORG	X'49	00 01945
1946	DA22P	GMSK	/P(DF00+P),10(STACK),16(PUSH),15(DA23+P),	00 01946
1947			CLB3,SFO,TFO	00 01947
1948		ORG	DA22+1	00 01948
1949	*			00 01949
1950	*	UPDATE OP B LENGTH		00 01950
1951	*			00 01951
1952	DA23	GEN	/N(DA24),11(BSSPEC),23(OPR),14(TRNB),15(LOG),	00 01952
1953			C17(GPROUT),24(R7)	00 01953
1954	*			00 01954
1955	*	CHECK DIGIT		00 01955
1956	*			00 01956
1957	DA24	GMSK	/P(DH00+P),10(STACK),16(PUSH),15(DA25+P),LB3,TFO,SFO	00 01957
1958	*			00 01958
1959	*	SAVE DIGIT		00 01959
1960	*			00 01960
1961	DA25	GEN	/N(DA26),12(ASP),14(TRNA),15(LOG),	00 01961
1962			C17(GPROUT),24(R3)	00 01962
1963	*			00 01963
1964	*	FETCH BYTE OF OP A		00 01964
1965	*			00 01965
1966	DA26	GEN	/N(DA27),11(BSGPR),23(RA),14(TRNB),15(LOG),	00 01966
1967			C17(GPROUT),24(R8)	00 01967
1968	DA27	GMSK	/P(DF00+P),10(STACK),15(DA28+P),LB3,TFO,SFO,	00 01968
1969			C12(ASGPR),16(RB),14(TRNA),13(OPROUT)	00 01969
1970	*			00 01970
1971	*	UPDATE OP A LENGTH		00 01971
1972	*			00 01972
1973	DA28	GEN	/N(DA29),11(BSSPEC),23(OPR),14(TRNB),15(LOG),	00 01973
1974			C17(GPROUT),24(RB)	00 01974
1975	*			00 01975
1976	*	CHECK DIGIT		00 01976
1977	*			00 01977
1978	DA29	GMSK	/P(DH00+P),10(STACK),16(PUSH),15(DA30+P),LB3,TFO,SFO	00 01978
1979	*			00 01979
1980	*	SETUP RESULT BYTE PTR FOR STORE;		00 01980
1981	*	CHOOSE BETWEEN ADD OR SUBTRACT.		00 01981
1982	*			00 01982
1983	DA30	GEN	/T(DA31S,DA31A),5(TT),7(SHFT),11(BSGPR),23(RD),	00 01983
1984			C14(TRNB),15(LOG),17(GPROUT),24(R8)	00 01984
1985	*			00 01985
1986	*	ADD OPA BYTE TO OPB BYTE		00 01986
1987	*			00 01987
1988	DA31A	GEN	/N(DA32),12(ASP),11(BSGPR),23(R3),14(ADD),	00 01988
1989			C17(GPROUT),24(R3)	00 01989
1990	*			00 01990
1991	*	SUBTRACT OPB BYTE FROM OPA BYTE		00 01991
1992	*			00 01992
1993	DA31S	GEN	/N(DA32),12(ASP),11(BSGPR),23(R3),14(SUB),16(CRY1),	00 01993
1994			C17(GPROUT),24(R3)	00 01994

```
1995 * 00 01995
1996 * ADD CARRY INTO RESULT 00 01996
1997 * 00 01997
1998 DA32 GEN /N(DA33),12(ASGPR),24(R5),11(BSGPR),23(R3), 00 01998
1999 C14(ADD),17(GPROUT) 00 01999
2000 * 00 02000
2001 * GET RESULT MODULO TEN AND NEW CARRY 00 02001
2002 * 00 02002
2003 DA33 GMSK /P(DV00+P),10(STACK),16(PUSH),15(DA34+P), 00 02003
2004 CLB3,SFO,TFO 00 02004
2005 * 00 02005
2006 * STORE RESULT BYTE 00 02006
2007 * 00 02007
2008 DA34 GEN /N(DA34P),12(ASGPR),24(RE),14(TRNA),15(LOG), 00 02008
2009 C13(OPROUT) 00 02009
2010 ORG X'15C 00 02010
2011 DA34P GMSK /P(DR00+P),10(STACK),16(PUSH),15(DA35+P), 00 02011
2012 CLB3,SFO,TFO 00 02012
2013 ORG DA34+1 00 02013
2014 * 00 02014
2015 * UPDATE RESULT LENGTH 00 02015
2016 * 00 02016
2017 DA35 GEN /N(DA36),11(BSSPEC),23(OPR),14(TRNB),15(LOG), 00 02017
2018 C17(GPROUT),24(RE) 00 02018
2019 * 00 02019
2020 * CHECK IF BOTH OPA AND OPB LENGTHS ZERO 00 02020
2021 * 00 02021
2022 DA36 GEN /N(DA37),12(ASGPR),24(RB),11(BSGPR),23(R7), 00 02022
2023 C14(OR),TFO,SFO,7(SSALU) 00 02023
2024 * 00 02024
2025 * WHEN BOTH LENGTHS ZERO, EXIT TO MOVE ROUTINE TO STORE CARRY 00 02025
2026 * AND THEN ZERO FILL REST OF RESULT. 00 02026
2027 * 00 02027
2028 DA37 GEN /T(DA21,DY11),5(FT),7(ALUZ),12(ASGPR),24(R5), 00 02028
2029 C14(TRNA),15(LOG),13(POUT) 00 02029
2030 EJEC 00 02030
2031 ***** 00 02031
2032 * 00 02032
2033 * TITLE; RESULT OPERAND SETUP ROUTINE 00 02033
2034 * 00 02034
2035 * PURPOSE; PERFORMS ALL SETUP AND PARAMETER BLOCK 00 02035
2036 * EVALUATION RELATED TO THE THIRD OPERAND. 00 02036
2037 * 00 02037
2038 * DESCRIPTION; REFER TO DE00 DESCRIPTION. 00 02038
2039 * 00 02039
2040 * 00 02040
2041 * ENTRY POINTS; DX01 - FULL RESULT OPERAND SETUP; * 00 02041
2042 * DX50 - SHIFTING SETUP ONLY; 00 02042
2043 * 00 02043
2044 * CALLING SEQUENCE: DX01 CALLED BY NORMAL BRANCH; * 00 02044
2045 * DX50 CALLED BY SUBR STACK; * 00 02045
2046 * 00 02046
2047 ***** 00 02047
2048 ORG *+1 00 02048
2049 * 00 02049
2050 * SETUP FIFTH PARAM BLOCK WORD 00 02050
2051 * 00 02051
```

```

2052 DX01 GEN /N(DX02),11(B$SPEC),23(MIR),14(TRNB),15(LOG), 00 02052
2053 C17(GPROUT),24(R8),TF0,SF0,7(IBR$1) 00 02053
2054 * 00 02054
2055 * FLAG RESULT IN THIRD OPERAND 00 02055
2056 * 00 02056
2057 DX02 GEN /N(DX03),14(ZERO),15(LOG),17(GPROUT),24(RC) 00 02057
2058 * 00 02058
2059 * ADVANCE PARAM BLOCK PTR 00 02059
2060 * 00 02060
2061 DX03 GEN /N(DX04),12(A$GPR),24(R2),14(INCA),16(CRY1), 00 02061
2062 C17(GPROUT) 00 02062
2063 * 00 02063
2064 * JUSTIFY AND SETUP RESULT LENGTH 00 02064
2065 * 00 02065
2066 DX04 GEN /N(DX05),12(A$GPRL),24(R8),14(SHFA),17(GPROUT) 00 02066
2067 DX05 GMSK /N(DX06),12(A$GPRL),16(R8),11(LIT),15(X'E0FF), 00 02067
2068 C14(AND),13(OPROUT) 00 02068
2069 DX06 GEN /N(DX07),11(B$SPEC),23(OLSE),14(TRNB),15(LOG), 00 02069
2070 C17(GPROUT),24(RE) 00 02070
2071 * 00 02071
2072 * INIT DISPLACEMENT TO RESULT SIGN; 00 02072
2073 * CHECK IF RESULT SIGNED; SET OR RESET QUOS ACCORDINGLY. 00 02073
2074 * 00 02074
2075 DX07 GEN /F(DX08),FSF,2(1),11(B$GPR),23(RE), 00 02075
2076 C14(TRNB),15(LOG),17(GPROUT),24(RF),WF1,SC0 00 02076
2077 DX08 GEN /N(DX09),14(ONES),WF1,SC0 00 02077
2078 * 00 02078
2079 * GET RESULT BYTE POINTER 00 02079
2080 * 00 02080
2081 DX09 GMSK /P(DB00+P),10(STACK),16(PUSH),15(DX10+P), 00 02081
2082 CLB3,SF0,TF0 00 02082
2083 * 00 02083
2084 * SAVE RESULT BYTE PTR; IF ERROR, EXIT. 00 02084
2085 * 00 02085
2086 DX10 GEN /T(DX11,DE40),5(FT),7(OVFL),12(A$P), 00 02086
2087 C14(TRNA),15(LOG),17(GPROUT),24(RD) 00 02087
2088 * 00 02088
2089 * HANDLE SHIFTING AND ROUNDING SPECIFICATION; THEN EXIT 00 02089
2090 * 00 02090
2091 DX50 EQU * SAVE LOCATION 00 02091
2092 ORG X'41 00 02092
2093 DX11 GMSK /P(DX50+P),10(STACK),16(PUSH),15(DE40+P), 00 02093
2094 CLB3,SF0,TF0 00 02094
2095 EJEC 00 02095
2096 * 00 02096
2097 * RESULT SHIFT/ROUND SETUP SUBR. 00 02097
2098 * 00 02098
2099 * GET SHIFT COUNT; CHECK IF RIGHT OR LEFT SHIFT. 00 02099
2100 * 00 02100
2101 ORG DX50 00 02101
2102 DX50 GMSK /F(DX51),FSE,2(1),11(MSK),15(X'FFEO), 00 02102
2103 C14(TRNB),13(OPROUT) 00 02103
2104 * 00 02104
2105 * RIGHT SHIFT, SO SAVE COUNT FOR LATER USE; 00 02105
2106 * CHECK IF ROUNDING; 00 02106
2107 * 00 02107
2108 DX51 GEN /F(COMPOP),FSE,2(4),12(A$GPR),24(RE), 00 02108

```

```
2109          C11(B$SPEC),23(ORLZ),14(OR),17(GPROUT)      00 02109
2110      ORG      *+1          SAVE SPACE FOR DX55 FIELD SEL.      00 02110
2111      *
2112      *      ROUNDING, SO INCREMENT RIGHT SHIFT COUNT      00 02111
2113      *
2114      ORG      X'2E          PLACE FOR FSEL      00 02112
2115      DX52      GMSK      /N(DX53),12(ASGPR),16(RE),11(LIT),15(X'FEFF),      00 02113
2116                  C14(ADD),13(OPROUT)      00 02114
2117      *
2118      *      PLACE INCREMENTED RIGHT SHIFT COUNT BACK IN REG AND EXIT.      00 02115
2119      *
2120      ORG      DX51+2
2121      DX53      GEN      /N(COMPOP),11(B$SPEC),23(OPR),14(TRNB),15(LOG),      00 02116
2122                  C17(GPROUT),24(RE)      00 02117
2123      *
2124      *      LEFT SHIFTING, SO PREPARE TO ZERO FILL;      00 02118
2125      *      PLACE COUNT IN SHIFT COUNTER;      00 02119
2126      *      CHECK IF ROUNDING;      00 02120
2127      *
2128      ORG      DX51+1          PLACE FOR FIELD SEL.      00 02121
2129      DX55      GEN      /F(DX56),FSF,2(2),12(AS$SPEC),22(AZERO),      00 02122
2130                  C11(B$SPEC),23(OPR),14(TCB),16(CRY1),13(SCOUT),      00 02123
2131                  C7(SSALU)      00 02124
2132      *
2133      *      NO ROUNDING, SO SETUP BYTE PTR FOR ZERO FILL; NO COUNT ADJUST.      00 02125
2134      *
2135      ORG      X'54
2136      DX56      GEN      /T(DX57,COMPOP),5(FT),7(ALUZ),      00 02126
2137                  C11(B$GPR),23(RD),14(TRNB),15(LOG),      00 02127
2138                  C17(GPROUT),24(R8)      00 02128
2139      *
2140      *      ROUNDING, SO DECR LEFT SHIFT COUNT;      00 02129
2141      *      IF THIS BRINGS IT TO ZERO, EXIT.      00 02130
2142      *
2143      ORG      X'56
2144      DX56R      GEN      /T(DX57,COMPOP),5(FT),7(SFTC),13(INCSC),      00 02131
2145                  C11(B$GPR),23(RD),14(TRNB),15(LOG),      00 02132
2146                  C17(GPROUT),24(R8)      00 02133
2147      *
2148      *      ZERO FILL BYTES SPECIFIED BY LEFT SHIFT COUNT      00 02134
2149      *
2150      ORG      X'22
2151      DX57      GEN      /N(DX57P),12(ASGPR),24(RE),14(TRNA),15(LOG),      00 02135
2152                  C13(OPROUT)      00 02136
2153      ORG      X'43
2154      DX57P      GMSK      /P(DZ00+P),10(STACK),16(PUSH),15(DX58+P),      00 02137
2155                  CLB3,SFO,TF0      00 02138
2156      *
2157      *      UPDATE RESULT LENGTH AND EXIT      00 02139
2158      *
2159      ORG      DX53+1
2160      DX58      GEN      /N(COMPOP),11(B$SPEC),23(OPR),14(TRNB),15(LOG),      00 02140
2161                  C17(GPROUT),24(RE)      00 02141
2162      EJECT      00 02142
2163      *****      00 02143
2164      *      00 02144
2165      *      TITLE:          MOVE DECIMAL STRING      00 02145
```

```
2166 * *00 02166
2167 * PURPOSE: THIS ROUTINE PERFORMS THE ACTUAL DECIMAL *00 02167
2168 * STRING MOVE OPERATION. *00 02168
2169 * *00 02169
2170 * ENTRY POINTS: DY00 *00 02170
2171 * *00 02171
2172 * CALLING SEQUENCE: CALLED FROM DE00 SETUP ROUTINE; *00 02172
2173 * EXITS TO 620 EMULATION IN ROM. *00 02173
2174 * ***** *00 02174
2175 * *00 02175
2176 * CHECK IF ROUNDING *00 02176
2177 * *00 02177
2178 * ORG X'39 *00 02178
2179 DY00 GEN /F(DY05),FSF,2(2) *00 02179
2180 * *00 02180
2181 * ROUNDING, SO SET OVERFLOW TO FLAG IT. *00 02181
2182 * *00 02182
2183 * ORG X'15A *00 02183
2184 DY01 GEN /N(DY05),TF0,SF1,7(SOVFL) *00 02184
2185 * *00 02185
2186 * CLEAR ROUND CARRY; *00 02186
2187 * CHECK IF RESULT SIGNED; *00 02187
2188 * *00 02188
2189 * ORG X'158 *00 02189
2190 DY05 GEN /T(DY06,DY07),5(TT),7(QU05),14(ZERO),15(LOG), *00 02190
2191 * C17(GPROUT),24(RC) *00 02191
2192 * *00 02192
2193 * STORE RESULT SIGN *00 02193
2194 * *00 02194
2195 * ORG X'5B *00 02195
2196 DY06 GMSK /P(DN00+P),10(STACK),15(DY07+P),LB3,TF0,SF0, *00 02196
2197 * C12(ASGPR),16(R3),14(TRNA),13(POUT) *00 02197
2198 * *00 02198
2199 * FETCH OP A BYTE *00 02199
2200 * *00 02200
2201 * ORG X'5A *00 02201
2202 DY07 GEN /N(DY08),11(BSGPR),23(RA),14(TRNB),15(LOG), *00 02202
2203 * C17(GPROUT),24(R8) *00 02203
2204 * ORG X'15D *00 02204
2205 DY08 GMSK /P(DF00+P),10(STACK),15(DY09+P),LB3,TF0,SF0, *00 02205
2206 * C12(ASGPR),16(RB),14(TRNA),13(OPROUT) *00 02206
2207 * *00 02207
2208 * UPDATE OP A LENGTH *00 02208
2209 * *00 02209
2210 DY09 GEN /N(DY10),11(BSSPEC),23(OPR),14(TRNB),15(LOG), *00 02210
2211 * C17(GPROUT),24(RB) *00 02211
2212 * *00 02212
2213 * CHECK DIGIT *00 02213
2214 * *00 02214
2215 DY10 GMSK /P(DH00+P),10(STACK),16(PUSH),15(DY11+P), *00 02215
2216 * CLB3,TF0,SF0 *00 02216
2217 * *00 02217
2218 * STORE BYTE *00 02218
2219 * *00 02219
2220 * ORG X'46 *00 02220
2221 DY11 GEN /N(DY12),11(BSGPR),23(RD),14(TRNB),15(LOG), *00 02221
2222 * C17(GPROUT),24(R8) *00 02222
```



```
2223      ORG      DY10+1                      00 02223
2224      DY12     GEN      /N(DY12P),12(ASGPR),24(RE),14(TRNA),15(LOG), 00 02224
2225                      C13(OPROUT)          00 02225
2226      ORG      X'4D                        00 02226
2227      DY12P     GMSK      /P(DR00+P),10(STACK),16(PUSH),15(DY13+P), 00 02227
2228                      CLB3,SFO,TFO         00 02228
2229      ORG      DY12+1                      00 02229
2230      *                               00 02230
2231      *      CHECK IF BOTH LENGTHS GONE TO ZERO 00 02231
2232      *                               00 02232
2233      DY13     GEN      /N(DY14),12(ASGPR),24(RB),11(B$SPEC),23(OPR), 00 02233
2234                      C14(OR),TFO,SFO,7(S$ALU) 00 02234
2235      *                               00 02235
2236      *      UPDATE RESULT LENGTH;          00 02236
2237      *      IF BOTH LENGTHS ZERO, EXIT    00 02237
2238      *                               00 02238
2239      DY14     GEN      /T(DY07,EX30),5(FT),7(ALUZ),11(B$SPEC),23(OPR), 00 02239
2240                      C14(TRNB),15(LOG),17(GPROUT),24(RE) 00 02240
2241      EJECT                               00 02241
2242      *****                          00 02242
2243      *                               *00 02243
2244      *      TITLE:          COMPARE DECIMAL STRING *00 02244
2245      *                               *00 02245
2246      *      PURPOSE:       COMPARE TWO DECIMAL STRINGS AND SET *00 02246
2247      *                      CONDIT TO INDICATE LT, EQ, OR GT. *00 02247
2248      *                               *00 02248
2249      *      ENTRY POINT:   DC00 *00 02249
2250      *                               *00 02250
2251      *      CALLING SEQUENCE: CALLED FROM DE00 SETUP ROUTINE; *00 02251
2252      *                      EXITS TO 620 EMULATION WHEN DONE *00 02252
2253      *                               *00 02253
2254      *****                          00 02254
2255      *                               00 02255
2256      *      IF LENGTHS OF OPERANDS BOTH ZERO, FLAG EQUAL AND EXIT. 00 02256
2257      *                               00 02257
2258      DC01     EQU      *      SAVE LOC.      00 02258
2259      ORG      X'28      PLACE FOR FSEL.      00 02259
2260      DC00     GEN      /N(DC01),12(ASGPR),24(RB),11(B$GPR),23(R7), 00 02260
2261                      C14(OR),TFO,SFO,7(S$ALU) 00 02261
2262      ORG      DC01      RESTORE LOC.          00 02262
2263      DC01     GEN      /T(DC01E,DC02),5(TT),7(ALUZ) 00 02263
2264      DC01E    GEN      /N(EX00),12(AS$PEC),22(AZERO),14(INCA),16(CRY1), 00 02264
2265                      C13(OPROUT)            00 02265
2266      ORG      *+1                      00 02266
2267      *                               00 02267
2268      *      CHECK IF SIGNS EQUAL          00 02268
2269      *                               00 02269
2270      DC02     GEN      /N(DC03),12(ASGPR),24(R3),11(B$GPR),23(R4), 00 02270
2271                      C14(EOR),15(LOG),TFO,SFO,7(S$ALU) 00 02271
2272      *                               00 02272
2273      *      IF SIGNS EQUAL, CALL COMPARE SUBR TO CHECK ABSOLUTE VALUES 00 02273
2274      *                               00 02274
2275      DC03     GEN      /T(DC04N,DC04),5(FT),7(ALUZ),12(ASGPR),24(RB), 00 02275
2276                      C14(TRNA),15(LOG),13(OPROUT) 00 02276
2277      *                               00 02277
2278      *      IF OP'S NEGATIVE, REVERSE BEFORE COMPARING ABS VALUES 00 02278
2279      *                               00 02279
```



```

2280 DC04 GEN /N(DC04A),12(ASGPR),24(R3),14(TRNA),15(LOG), 00 02280
2281 CTFO,SFO,7(SSALU) 00 02281
2282 ORG X'5D 00 02282
2283 DC04A GEN /T(DC04C,DC04B),5(TT),7(ALUZ) 00 02283
2284 ORG X'5E 00 02284
2285 DC04B GMSK /P(DA11S+P),10(STACK),16(PUSH),15(DC04C+P), 00 02285
2286 CLB3,TFO,SFO 00 02286
2287 * 00 02287
2288 * COMPARE ABS VALUES 00 02288
2289 * 00 02289
2290 ORG X'5F 00 02290
2291 DC04C GMSK /P(DP00+P),10(STACK),16(PUSH),15(DC05+P), 00 02291
2292 CLB3,TFO,SFO 00 02292
2293 * 00 02293
2294 * SIGNS NOT EQUAL, SO SIGN OF A DETERMINES IF GT OR LT. 00 02294
2295 * 00 02295
2296 ORG DC04+1 00 02296
2297 DC04N GEN /N(DC06),12(AS$SPEC),22(AZERO),11(B$GPR),23(R3), 00 02297
2298 C14(SUB),16(CRY1),TFO,SFO,7(SSALU) 00 02298
2299 * 00 02299
2300 * IF ABSOLUTE VALUES EQUAL, JUMP TO FLAG IT ; 00 02300
2301 * 00 02301
2302 DC05 GEN /T(DC06,DC01E),5(FT),7(ALUZ) 00 02302
2303 * 00 02303
2304 * MAKE GT/LT DECISION 00 02304
2305 * 00 02305
2306 ORG DC01E+1 00 02306
2307 DC06 GEN /T(DC07L,DC07G),5(TT),7(ALUS),12(AS$SPEC),22(AZERO), 00 02307
2308 C14(INCA),16(CRY1),13(OPROUT) 00 02308
2309 ORG DC05+1 00 02309
2310 * 00 02310
2311 * FLAG LESS THAN 00 02311
2312 * 00 02312
2313 DC07L GEN /N(EX00),11(B$SPEC),23(OPR),12(AS$SPEC),22(AONES), 00 02313
2314 C14(DECb),13(OPROUT) 00 02314
2315 * 00 02315
2316 * FLAG GREATER THAN 00 02316
2317 * 00 02317
2318 DC07G GEN /N(EX00),11(B$SPEC),23(OPR),12(AS$SPEC),22(AZERO), 00 02318
2319 C14(INCB),16(CRY1),13(OPROUT) 00 02319
2320 EJEC 00 02320
2321 ***** 00 02321
2322 * 00 02322
2323 * TITLE: COMPARE SUBROUTINE 00 02323
2324 * 00 02324
2325 * PURPOSE: COMPARE THE ABSOLUTE VALUES OF TWO DECIMAL 00 02325
2326 * DIGIT STRINGS. 00 02326
2327 * 00 02327
2328 * DESCRIPTION: THE MOST SIGNIFICANT DIGIT POSITIONS ARE 00 02328
2329 * FIRST COMPUTED BY ADDING THE LENGTHS AND 00 02329
2330 * LEFT SHIFT COUNTS. SINCE LEADING ZEROS 00 02330
2331 * HAVE BEEN SUPPRESSED, A LT/GT DECISION 00 02331
2332 * CAN BE MADE IF THE MOST SIGNIFICANT DIGIT 00 02332
2333 * POSITIONS ARE UNEQUAL. IF THEY ARE EQUAL, 00 02333
2334 * THEN CHARACTERS ARE FETCHED UNTIL AN 00 02334
2335 * UNEQUAL PAIR ARE FOUND OR THE END OF THE 00 02335
2336 * STRINGS IS REACHED. 00 02336

```

```
2337 *                                *00 02337
2338 * ENTRY POINT:          DP00      *00 02338
2339 *                                *00 02339
2340 * CALLING SEQUENCE:      CALLED USING MICRO-SUBROUTINE STACK. *00 02340
2341 *                        RETURNS RESULT THROUGH ALUZ AND ALUS. *00 02341
2342 *                        ALUZ = A=B      * 00 02342
2343 *                        ALUS = A<B      * 00 02343
2344 *                                *00 02344
2345 * REGISTER USAGE:       R5- DISPLACEMENT TO CURRENT BYTES *00 02345
2346 *                        BEING CHECKED; *00 02346
2347 *                        RC- TEMP        *00 02347
2348 *                                *00 02348
2349 *.....*00 02349
2350 *                                00 02350
2351 * GET MOST SIGNIFICANT DIGIT POSITION FOR OP A BY ADDING 00 02351
2352 * LENGTH AND LEFT SHIFT COUNT. 00 02352
2353 *                                00 02353
2354 DP00 GEN /N(DP01),11(B$SPEC),23(OLSE),14(TRNB),15(LOG), 00 02354
2355 C13(POUT) 00 02355
2356 DP01 GEN /N(DP02),12(ASP),11(B$SPEC),23(ORZF),14(ADD), 00 02356
2357 C13(POUT) 00 02357
2358 *                                00 02358
2359 * SAVE THIS NET LENGTH IN SHIFT COUNTER AS NUMBER OF 00 02359
2360 * BYTES TO COMPARE. 00 02360
2361 *                                00 02361
2362 DP02 GEN /N(DP03),12(ASP),14(TRNA),15(LOG),13(OPROUT) 00 02362
2363 DP03 GEN /N(DP04),12(AS$PEC),22(AZERO),11(B$SPEC),23(OPR), 00 02363
2364 C14(TCB),16(CRY1),13(SCOUT) 00 02364
2365 *                                00 02365
2366 * CHECK IF OP B MOST SIG DIGIT POSITION IS SAME AS FOR OP A. 00 02366
2367 *                                00 02367
2368 DP04 GEN /N(DP05),12(AS$GPR),24(R7),14(TRNA),15(LOG), 00 02368
2369 C13(OPROUT) 00 02369
2370 DP05 GEN /N(DP06),12(ASP),11(B$SPEC),23(OLSE), 00 02370
2371 C14(SUB),16(CRY1),13(POUT) 00 02371
2372 DP06 GEN /N(DP07),12(ASP),11(B$SPEC),23(ORZF), 00 02372
2373 C14(SUB),16(CRY1),13(POUT),TFO,SFO,7(S$ALU) 00 02373
2374 *                                00 02374
2375 * INIT DISPLACEMENT FOR COMPARE OF FIRST BYTES; 00 02375
2376 * IF MOST SIG DIGIT PLACES NOT EQUAL, SKIP COMPARE AND EXIT. 00 02376
2377 *                                00 02377
2378 DP07 GEN /T(DP08,COMPOP),5(TT),7(ALUZ),12(AS$PEC),22(AZERO), 00 02378
2379 C14(INCA),16(CRY1),17(GPROUT),24(R5) 00 02379
2380 *                                00 02380
2381 * CHECK IF AT END OF OP B STRING 00 02381
2382 *                                00 02382
2383 DP09 EQU * SAVE LOC 00 02383
2384 ORG X'23 PLACE FOR FIELD SEC. 00 02384
2385 DP08 GEN /N(DP09),12(AS$GPR),24(R7),11(B$GPR),23(R5), 00 02385
2386 C14(SUB),16(CRY1),13(OPROUT) 00 02386
2387 ORG DP09 RESTORE LOC. 00 02387
2388 DP09 GEN /N(DP10),11(B$SPEC),23(ORSE),14(TRNB),15(LOG), 00 02388
2389 CTFO,SFO,7(S$ALU),13(OPROUT) 00 02389
2390 *                                00 02390
2391 * SETUP OP B BYTE PTR FOR FETCH; 00 02391
2392 * IF AT END OF OP B STRING, SKIP FETCH AND USE ZERO AS BYTE. 00 02392
2393 *                                00 02393
```

```

2394 DP10 GEN /T(DP12,DP11),5(TT),7(ALUS),11(B$GPR),23(R6), 00 02394
2395 C14(TRNB),15(LOG),17(GPROUT),24(R8) 00 02395
2396 * 00 02396
2397 * FETCH OP B BYTE 00 02397
2398 * 00 02398
2399 ORG *+1 00 02399
2400 DP11 GEN /N(DP12),12(A$GPR),24(R5),14(TRNA),15(LOG), 00 02400
2401 C13(OPROUT) 00 02401
2402 ORG *-2 00 02402
2403 DP12 GMSK /P(DFOO+P),10(STACK),16(PUSH),15(DP12H+P), 00 02403
2404 CLB3,TFO,SFO 00 02404
2405 ORG *+1 00 02405
2406 DP12H GMSK /P(DH00+P),10(STACK),16(PUSH),15(DP13+P), 00 02406
2407 CLB3,SFO,TFO 00 02407
2408 * 00 02408
2409 * SAVE OP B DIGIT 00 02409
2410 * 00 02410
2411 DP13 GEN /N(DP14),12(A$P),14(TRNA),15(LOG), 00 02411
2412 C17(GPROUT),24(RC) 00 02412
2413 * 00 02413
2414 * CHECK IF AT END OF OP A STRING 00 02414
2415 * 00 02415
2416 DP14 GEN /N(DP15),12(A$GPR),24(RB),11(B$GPR),23(R5), 00 02416
2417 C14(SUB),16(CRY1),13(OPROUT) 00 02417
2418 DP15 GEN /N(DP16),11(B$SPEC),23(ORSE),14(TRNB),15(LOG), 00 02418
2419 CTFO,SFO,7(SSALU),13(OPROUT) 00 02419
2420 * 00 02420
2421 * SETUP BYTE PTR FOR FETCH OF OP A DIGIT ; 00 02421
2422 * IF END OF OP A, SKIP FETCH AND RETURN ZERO. 00 02422
2423 * 00 02423
2424 DP16 GEN /T(DP18,DP17),5(TT),7(ALUS),11(B$GPR),23(RA), 00 02424
2425 C14(TRNB),15(LOG),17(GPROUT),24(R8) 00 02425
2426 * 00 02426
2427 * FETCH OP A DIGIT 00 02427
2428 * 00 02428
2429 DP17 GEN /N(DP18),12(A$GPR),24(R5),14(TRNA),15(LOG), 00 02429
2430 C13(OPROUT) 00 02430
2431 DP18 GMSK /P(DFOO+P),10(STACK),16(PUSH),15(DP18H+P), 00 02431
2432 CLB3,TFO,SFO 00 02432
2433 DP18H GMSK /P(DH00+P),10(STACK),16(PUSH),15(DP19+P), 00 02433
2434 CLB3,SFO,TFO 00 02434
2435 * 00 02435
2436 * COMPARE OP A DIGIT TO OP B DIGIT 00 02436
2437 * 00 02437
2438 DP19 GEN /N(DP20),12(A$P),11(B$GPR),23(RC),14(SUB),16(CRY1), 00 02438
2439 CTFO,SFO,7(SSALU) 00 02439
2440 * 00 02440
2441 * ADVANCE DISPLACEMENT FOR COMPARE OF NEXT BYTE; 00 02441
2442 * IF BYTES UNEQUAL, EXIT ; 00 02442
2443 * 00 02443
2444 DP20 GEN /T(DP21,COMPOP),5(TT),7(ALUZ),12(A$GPR),24(R5), 00 02444
2445 C14(INCA),16(CRY1),17(GPROUT) 00 02445
2446 * 00 02446
2447 * IF COMPARED ALL BYTES, EXIT; 00 02447
2448 * ELSE, BACK TO COMPARE NEXT. 00 02448
2449 * 00 02449
2450 DPEND EQU * SAVE LOC. 00 02450

```

```
2451      ORG      X'25          PLACE FOR FIELD SEL.          00 02451
2452      DP21    GEN      /T(DPO8,COMPOP),5(FT),7(SFTC),13(INCSC) 00 02452
2453      ORG      DPEND        RESTORE LOC.                   00 02453
2454      EJEC                                           00 02454
2455      *                                           00 02455
2456      *      STANDARD EXIT INSTRUCTIONS                   00 02456
2457      *                                           00 02457
2458      *                                           00 02458
2459      *                                           00 02459
2460      *      ERROR #3 EXIT                               00 02460
2461      *                                           00 02461
2462      ER03    GMSK      /N(EX00),11(LIT),15(X'FFFC),14(TRNB),13(OPROUT) 00 02462
2463      *                                           00 02463
2464      *      SET CONDITION WORD                           00 02464
2465      *                                           00 02465
2466      EX00    GEN      /N(EX10),12(ASGPR),24(R1),14(TRNA),15(LOG), 00 02466
2467      C6(MEMC),10(OS$ALU)                                00 02467
2468      EX10    GEN      /N(EX30),6(SPEC),10(WAITMD),11(B$SPEC),23(OPR), 00 02468
2469      C14(TRNB),15(LOG)                                   00 02469
2470      *                                           00 02470
2471      *      RETURN TO 620 EMULATION                       00 02471
2472      *      RESTORE P, R3, AND R5.                       00 02472
2473      *                                           00 02473
2474      ORG      X'52                                           00 02474
2475      EX30    GEN      /N(EX35),6(MEMC),10(OF$ALU),12(ASGPR),24(R0), 00 02475
2476      C14(TRNA),15(LOG)                                   00 02476
2477      ORG      X'DE                                           00 02477
2478      EX35    GEN      /N(EX40),14(ZERO),15(LOG),17(GPROUT),24(R3), 00 02478
2479      C6(SPEC),10(WAITMD)                                00 02479
2480      ORG      EX10+1                                         00 02480
2481      EX40    GEN      /N(DEC06),14(ONES),17(GPROUT),24(R5)   00 02481
2482      *                                           00 02482
2483      *      INTERRUPT EXIT                                00 02483
2484      *      RESET RB AND RC TO V75 VALUES               00 02484
2485      *      NOTE: RE MUST CONTAIN BCS ADDRESS ON ENTRY TO THIS ROUTINE 00 02485
2486      *                                           00 02486
2487      ORG      X'50                                           00 02487
2488      *                                           00 02488
2489      *      START I/O: LOOP TILL REQUEST ACCEPTED.       00 02489
2490      *                                           00 02490
2491      IEXIT    GEN      2(IEXIT),TS7,GF4,TF0,SFO,IME,MT1,MR1, 00 02491
2492      C14(ZERO),15(LOG),17(GPROUT),24(RB)                 00 02492
2493      *                                           00 02493
2494      *      ONCE I/O STARTED, BRANCH BACK TO ROM          00 02494
2495      *                                           00 02495
2496      GEN      /P(INT1),10(PJMP),SFO,TF0,14(ONES),17(GPROUT),24(RC) 00 02496
2497      ORG      EX40+1                                         00 02497
2498      EJEC                                           00 02498
2499      *****                                           00 02499
2500      *                                           00 02500
2501      *                                           00 02501
2502      *      TITLE'      MULTIPLY                          00 02502
2503      *                                           00 02503
2504      *      PURPOSE:    MULTIPLY TWO DECIMAL DIGIT STRINGS. DIGITS ARE 00 02504
2505      *                   FETCHED ONE AT A TIME FROM OP2. OP1 IS ADDED 00 02505
2506      *                   SPECIFIED NUMBER OF TIMES TO TEMP RESULT.    00 02506
2507      *                   WHEN MULTIPLY IS COMPLETE, PRODUCT IS IN TEMP. 00 02507
```

```

2508 *
2509 * CALLED BY THIS ROUTINE:
2510 *      DT50      CALCULATE LENGTH OF REQUIRED TEMP
2511 *                  (OP1=RA ADDR, R8 LENGTH)
2512 *                  (OP2=R7 LENGTH)
2513 *      DN00      STORE SIGN (+ IF P=0, - IF P=-1)
2514 *                  RA CONTAINS ADDR
2515 *                  RF DISPLACEMENT
2516 *      DZ00      ZERO FILL
2517 *                  SC=-ZERO COUNT
2518 *                  OPR=INITIAL DISPLACEMENT
2519 *                  R8=BASE BYTE POINTER
2520 *      DF00      FETCH BYTE      (ALSO DF10)
2521 *                  R8=BASE BYTE POINTER
2522 *                  OPR=DISPLACEMENT
2523 *                  BYTE RETURNED IN P
2524 *      DH00      VALIDATE DIGIT IN P
2525 *      DV00      DIVIDE BY 10
2526 *                  R5 1 INIT VALUE
2527 *                  QUOTIENT RETURNED IN R5
2528 *                  REMAINDER RETURNED IN P
2529 *      DR00      STORE BYTE
2530 *                  R8=BASE BYTE POINTER
2531 *                  OPR=DISPLACEMENT
2532 *                  P=BYTE
2533 *
2534 * REGISTER USAGE:
2535 *      ENTRY
2536 *      R3  OP 1 SIGN
2537 *      R4  OP 2 SIGN
2538 *      R0  TEMP WORD ADDR
2539 *      RA  OP1 BASE ADDR
2540 *      RB  OP1 DISPL
2541 *      R6  OP2 BASE ADDR
2542 *      R7  OP2 DISPL
2543 *
2544 *      INTERNAL
2545 *      R8  PASS BASE ADDR FOR FETCH, STORE
2546 *      OPR PASS DISPL FOR FETCH/STORE
2547 *      R 5  CARRY PROPAGATION
2548 *
2549 *****
2550 DTCONT EQU *
2551 *****
2552 *****
2553 *****
2554 *****
2555 *****
2556 *****
2557 *****
2558 *****
2559 *****
2560 *****
2561 *****
2562 *****
2563 *****
2564 *****

```

2565	*****			00 02565
2566	*****		ITS = SIGN FOR RESULT	00 02566
2567	*****			00 02567
2568	DT02	GEN	/N(DT03),14(ONES),13(POUT)	00 02568
2569	*****			00 02569
2570	*****		INIT POSITIVE SIGN AND SELECT + OR =	00 02570
2571	*****			00 02571
2572	DT01	GEN	/T(DT03,DT02),5(TT),7(ALUZ),14(ZERO),15(LOG),	00 02572
2573			C13(POUT)	00 02573
2574	*****			00 02574
2575	*****		SELECT MULTIPLY/DIVIDE	00 02575
2576	*****			00 02576
2577	DT04	GEN	/F(DT05),2(X'4),FSF,11(B\$GPR),23(RB),14(TRNB),	00 02577
2578			C15(LOG),17(GPROUT),24(R4)	00 02578
2579	*****			00 02579
2580	*****		STORE RESULT SIGN	00 02580
2581	*****			00 02581
2582	DT03	GMSK	/P(DN00+P),IMD,LB3,15(DT04+P),AK2	00 02582
2583	DTCON2	EQU	*	00 02583
2584	*****			00 02584
2585	*****		COMPUTE RESULT DISPLACEMENT	00 02585
2586	*****			00 02586
2587		ORG	X'58	00 02587
2588	DT05	GMSK	/P(DT50+P),IMD,LB3,15(DT07+P),AK2	00 02588
2589	*****			00 02589
2590	*****		INIT SHIFT COUNT WITH -DISPLACEMENT	00 02590
2591	*****		(FOR ZERO FILL)	00 02591
2592	*****			00 02592
2593		ORG	DTCON2	00 02593
2594	DT07	GEN	/N(DT08),11(B\$GPR),23(R3),12(A\$SPEC),22(AZERO),	00 02594
2595			C14(TCB),16(CRY1),13(SCOUT)	00 02595
2596	*****			00 02596
2597	*****		GET RTR INTO TEMP AREA FOR RESULT LENGTH STORE	00 02597
2598	*****			00 02598
2599	DT08	GEN	/N(DT09),11(B\$GPR),23(R0),12(A\$SPEC),22(AZERO),	00 02599
2600			C14(INCB),16(CRY1),17(GPROUT),24(RF),	00 02600
2601			C6(MEMC),10(OS\$ALU)	00 02601
2602	*****			00 02602
2603	*****		STORE TEMP LENGTH	00 02603
2604	*****			00 02604
2605	DT09	GEN	/N(DT09H),11(B\$SPEC),23(ORLZ),14(TRNB),15(LOG),	00 02605
2606			C6(SPEC),10(WAITMD)	00 02606
2607	*****			00 02607
2608	*****		CONVERT TEMP PTR INTO BYTE PTR	00 02608
2609	*****			00 02609
2610		ORG	X'59	00 02610
2611	DT09H	GEN	/N(DT10),12(A\$GPR),24(RF),14(TRNA),15(LOG),SH0,	00 02611
2612			C17(GPROUT)	00 02612
2613		ORG	DT09+1	00 02613
2614	*****			00 02614
2615	*****		SETUP TEMP BYTE PTR FOR ZERO FILL	00 02615
2616	*****			00 02616
2617	DT10	GEN	/N(DT10A),11(B\$GPR),23(RF),14(TRNB),15(LOG),	00 02617
2618			C17(GPROUT),24(R8)	00 02618
2619	*****			00 02619
2620	*****		ZERO FILL EXPECTED RESULT	00 02620
2621	*****			00 02621

2622	DT10A	GMSK	/P(DZ00+P),IMD,LB3,15(DT11+P),AK2	00 02622
2623	*****			00 02623
2624	*****		INIT CARRY (R5) AND TEST FOR INTERRUPTS	00 02624
2625	*****			00 02625
2626	DT11	GEN	/N(DT12),14(ZERO),15(LOG),17(GPROUT),24(R5),	00 02626
2627			CTS7,GF4,TFO,SFO,IMC	00 02627
2628	*****			00 02628
2629	*****		COMPUTE REMAINING RESULT DISPLACEMENT	00 02629
2630	*****			00 02630
2631	DT12	GMSK	/P(DT50+P),IMD,LB3,15(DT13+P),AK2	00 02631
2632	*****			00 02632
2633	*****		PLACE OP2 BASE ADDR IN R8 FOR DIGIT FETCH	00 02633
2634	*****			00 02634
2635	DT13	GEN	/N(DT14),11(B\$GPR),23(R6),14(TRNB),15(LOG),	00 02635
2636			C17(GPROUT),24(R8)	00 02636
2637	*****			00 02637
2638	*****		SET DISPLACEMENT FOR BYTE FETCH	00 02638
2639	*****			00 02639
2640	DT14	GEN	/N(DT15),12(A\$GPR),24(R7),14(TRNA),15(LOG),13(OPROUT)	00 02640
2641	*****			00 02641
2642	*****		FETCH OP2 DIGIT	00 02642
2643	*****			00 02643
2644	DT15	GMSK	/P(DF00+P),IMD,LB3,15(DT16+P),AK2	00 02644
2645	*****			00 02645
2646	*****		SET NEW DISPLACEMENT IN R7	00 02646
2647	*****			00 02647
2648	DT16	GEN	/N(DT17),11(B\$SPEC),23(OPR),14(TRNB),15(LOG),	00 02648
2649			C17(GPROUT),24(R7)	00 02649
2650	*****			00 02650
2651	*****		VALIDATE FETCHED DIGIT	00 02651
2652	*****			00 02652
2653		ORG	X'3D LEAVE 77 ADDRESS FOR INT ROUTINE	00 02653
2654	DT17	GMSK	/P(DH00+P),10(STACK),16(PUSH),15(DT18+P),	00 02654
2655			CLB3,TFO,SFO	00 02655
2656		ORG	DT16+2 SKIP OVER DT100	00 02656
2657	*			00 02657
2658	*	SAVE DIGIT IN RD		00 02658
2659	*			00 02659
2660	DT18	GEN	/N(DT19),11(B\$SPEC),23(OPR),14(TRNB),15(LOG),	00 02660
2661			C17(GPROUT),24(RD)	00 02661
2662	*****			00 02662
2663	*****		TEST FOR MULTIPLY COMPLETE	00 02663
2664	*****			00 02664
2665	DT37	GEN	/N(DT38),12(A\$GPR),24(R7),14(TRNA),15(LOG),7(S\$ALU)	00 02665
2666	*****			00 02666
2667	*****		PLACE OP 1 ADDR IN R8 FOR FETCH	00 02667
2668	*****			00 02668
2669	DT20	GEN	/N(DT21),11(B\$GPR),23(RA),14(TRNB),15(LOG),	00 02669
2670			C17(GPROUT),24(R8)	00 02670
2671	*****			00 02671
2672	*****		FETCH OP1 DIGIT / MOVE DISPLACEMENT TO OPR	00 02672
2673	*****			00 02673
2674	DT21	GMSK	/P(DF00+P),IMD,LB3,15(DT22+P),12(A\$GPR),16(RB),	00 02674
2675			C14(TRNA),13(OPROUT)	00 02675
2676	*****			00 02676
2677	*****		RESTORE NEW DISPLACEMENT IN RB	00 02677
2678	*****			00 02678

2679	DT22	GEN	/N(DT23),11(B\$SPEC),23(OPR),14(TRNB),15(LOG),	00 02679
2680			C17(GPROUT),24(RB)	00 02680
2681	*****			00 02681
2682	*****		VALIDATE OP1 DIGIT	00 02682
2683	*****			00 02683
2684	DT23	GMSK	/P(DH00+P),IMD,LB3,15(DT24+P),AK2	00 02684
2685	*****			00 02685
2686	*****		SET NEW OP 1 ADDR IN RA	00 02686
2687	*****			00 02687
2688	DT39	GEN	/N(DT40),11(B\$GPR),23(RF),14(TRNB),15(LOG),	00 02688
2689			C17(GPROUT),24(RA)	00 02689
2690	*****			00 02690
2691	*****		PLACE TEMP ADDR IN R8 FOR FETCH OF TEMP RESULT	00 02691
2692	*****		IGNORE OP1 IF ZERO	00 02692
2693	*****			00 02693
2694	DT24	GEN	/T(DT25,DT26),5(FT),7(ALUZ),11(B\$GPR),23(RF),	00 02694
2695			C14(TRNB),15(LOG),17(GPROUT),24(R8)	00 02695
2696	*****			00 02696
2697	*****		FETCH TEMP RESULT BYTE	00 02697
2698	*****		(MOVE DISPLACEMENT TO OPR)	00 02698
2699	*****			00 02699
2700	DT26	GMSK	/P(DF00+P),IMD,LB3,15(DT27+P),12(A\$GPR),16(R3),	00 02700
2701			C14(TRNA),13(OPROUT)	00 02701
2702	*****			00 02702
2703	*****		ADD OP1 + OP1 UNTIL OP2 DIGIT EXHAUSTED	00 02703
2704	*****		(NOTE CARRY IN R5)	00 02704
2705	DT25	GEN	/T(DT25,DT26),5(FT),7(SFTC),12(ASP),11(B\$GPR),	00 02705
2706			C23(R5),14(ADD),17(GPROUT),24(R5),13(INCSC)	00 02706
2707	*****			00 02707
2708	*****		ISOLATE DIGIT FROM ASCII	00 02708
2709	*****			00 02709
2710	DT27	GMSK	/N(DT28),11(LIT),15(X'FF4F),12(ASP),14(SUB),	00 02710
2711			C16(CRY1),13(POUT)	00 02711
2712	*****			00 02712
2713	*****		BUILD NEW TEMP RESULT	00 02713
2714	*****			00 02714
2715	DT28	GEN	/N(DT29),12(ASP),11(B\$GPR),23(R5),14(ADD),	00 02715
2716			C17(GPROUT),24(R5)	00 02716
2717	*****			00 02717
2718	*****		GET VALUE OF CARRY (DIVIDE BY 10)	00 02718
2719	*****			00 02719
2720	DT29	GMSK	/P(DV00+P),IMD,LB3,15(DT30+P),AK2	00 02720
2721	*****			00 02721
2722	*****		SET BYTE STORE ADDR IN R8	00 02722
2723	*****			00 02723
2724	DT30	GEN	/N(DT31),11(B\$GPR),23(RF),14(TRNB),15(LOG),	00 02724
2725			C17(GPROUT),24(R8)	00 02725
2726	*****			00 02726
2727	*****		STORE DIGIT IN TEMP RESULT	00 02727
2728	*****		(MOVE DISPLACEMENT TO OPR)	00 02728
2729	*****			00 02729
2730	DT31	GMSK	/P(DR00+P),IMD,LB3,15(DT32+P),12(A\$GPR),16(R3),	00 02730
2731			C14(TRNA),13(OPROUT)	00 02731
2732	*****			00 02732
2733	*****		RESTORE NEW DISPLACEMENT IN R3	00 02733
2734	*****			00 02734
2735	DT32	GEN	/N(DT33),11(B\$SPEC),23(OPR),14(TRNB),15(LOG),	00 02735

2736			C17(GPROUT),24(R3)	00 02736
2737	*****			00 02737
2738	*****		TEST FOR OP 1 EXHAUSTED	00 02738
2739	*****			00 02739
2740	DT33	GEN	/N(DT34),12(ASGPR),24(RB),14(TRNA),15(LOG),	00 02740
2741			C7(SSALU)	00 02741
2742	*****			00 02742
2743	*****		PREPARE CARRY STORE IN CASE COMPLETE	00 02743
2744	*****		EITHER STORE CARRY OR CONTINUE MULTIPLY	00 02744
2745	*****			00 02745
2746	DT34	GEN	/T(DT19,DT35),5(FT),7(ALUZ),12(ASGPR),24(R5),	00 02746
2747			C14(TRNA),15(LOG),13(POUT)	00 02747
2748	*****			00 02748
2749	*****		PLACE REMAINDER STORE ADDR IN R8	00 02749
2750	*****			00 02750
2751	DT35	GEN	/N(DT36),11(BSGPR),23(RF),14(TRNB),15(LOG),	00 02751
2752			C17(GPROUT),24(R8)	00 02752
2753	*****			00 02753
2754	*****		INIT MULTIPLY REPEAT COUNT: (MINUS DIGIT)	00 02754
2755	*****		DONT MULTIPLY IF ZERO DIGIT	00 02755
2756	*****			00 02756
2757	DT19	GEN	/T(DT37,DT20),5(TT),7(ALUZ),12(ASPEC),22(AZERO),	00 02757
2758			C11(BSGPR),23(RD),14(TCB),16(CRY1),13(SCOUT)	00 02758
2759	*****			00 02759
2760	*****		STORE CARRY IN TEMP RESULT	00 02760
2761	*****			00 02761
2762	DT36	GMSK	/P(DR00+P),IMD,LB3,15(DT37+P),12(ASGPR),16(R3),	00 02762
2763			C14(TRNA),13(OPROUT)	00 02763
2764	*****			00 02764
2765	*****		RESTORE OP1 LENGTH AND CONTINUE MULTIPLY	00 02765
2766	*****		UNTIL COMPLETE	00 02766
2767	*****			00 02767
2768	DT38	GEN	/T(DT11,DT39),5(FT),7(ALUZ),11(BSGPR),23(R4),	00 02768
2769			C14(TRNB),15(LOG),17(GPROUT),24(RB)	00 02769
2770	*****			00 02770
2771	*****		PREPARE LENGTH FETCH (TEMP BASE TO R8)	00 02771
2772	*****			00 02772
2773	DT40	GEN	/N(DT41),11(BSGPR),23(RA),14(TRNB),15(LOG),	00 02773
2774			C17(GPROUT),24(R8)	00 02774
2775	*****			00 02775
2776	*****		FETCH TEMP LENGTH (NEED NON ZERO OPR)	00 02776
2777	*****			00 02777
2778	DT41	GMSK	/P(DF10+P),IMD,LB3,15(DT42+P),12(ASGPR),16(R2),	00 02778
2779			C14(TRNA),13(OPROUT)	00 02779
2780	*****			00 02780
2781	*****		SAVE TEMP LENGTH	00 02781
2782	*****		RESTORE RD BY FETCHING SAVED VALUE	00 02782
2783	*****		EXIT TO MOVE RESULT TO DESTINATION	00 02783
2784	*****			00 02784
2785	DT42	GMSK	/N(DT43),11(LIT),15(X'FFAF),12(ASGPR),16(R0),	00 02785
2786			C14(ADD),6(MEMC),10(OF\$ALU)	00 02786
2787		ORG	X'CD	00 02787
2788	DT43	GEN	/N(DT44),12(ASP),14(TRNA),15(LOG),17(GPROUT),	00 02788
2789			C24(RB),6(SPEC),10(WAITMD)	00 02789
2790		ORG	X'148	00 02790
2791	DT44	GEN	/N(DY07),11(BSPEC),23(M1R),14(TRNB),15(LOG),	00 02791
2792			C17(GPROUT),24(RD)	00 02792

```
2793          ORG      DT42+1          00 02793
2794          EJECT          00 02794
2795          *****          00 02795
2796          *          00 02796
2797          * TITLE:      COMPUTE RESULT DISPLACEMENT          00 02797
2798          *          00 02798
2799          * PURPOSE:    CALCULATE LENGTH OF RESULT BY ADDING LEFT SHIFT          00 02799
2800          *          COUNTS AND LENGTHS OF BOTH OPERANDS.          00 02800
2801          *          00 02801
2802          * REGISTERUSAGE:      ENTRY          00 02802
2803          *          RB = OP1  LSC/DISPLACEMENT          00 02803
2804          *          R7 = OP2  LSC/DISPLACEMENT          00 02804
2805          *          00 02805
2806          *          EXIT          00 02806
2807          *          R3 = ESTIMATED RESULT DISPLACEMENT          00 02807
2808          *          00 02808
2809          *          INTERNAL          00 02809
2810          *          OPER= USED FOR SPLITTING LSC/DISPL          00 02810
2811          *          00 02811
2812          *****          00 02812
2813          *****          00 02813
2814          *****          BEGIN ADD OR OP 1 LSC + LENGTH          00 02814
2815          *****          00 02815
2816          DT50      GEN      /N(DT51),12(ASGPR),24(RB),14(TRNA),15(LOG),13(OPROUT)          00 02816
2817          *****          00 02817
2818          *****          PLACE LSC IN R3          00 02818
2819          *****          00 02819
2820          DT51      GEN      /N(DT52),11(B$SPEC),23(OLSE),14(TRNB),15(LOG),          00 02820
2821          *****          C17(GPROUT),24(R3)          00 02821
2822          *****          00 02822
2823          *****          ADD LENGTH TO LSC          00 02823
2824          *****          00 02824
2825          DT52      GEN      /N(DT53),12(ASGPR),24(R3),17(GPROUT),11(B$SPEC),          00 02825
2826          *****          C23(ORZF),14(ADD)          00 02826
2827          *****          00 02827
2828          *****          NOW BEGIN ADD OF OP 2 LSC + LENGTH          00 02828
2829          *****          00 02829
2830          DT53      GEN      /N(DT54),12(ASGPR),24(R7),14(TRNA),15(LOG),13(OPROUT)          00 02830
2831          *****          00 02831
2832          *****          ADD SHIFT COUNT          00 02832
2833          *****          00 02833
2834          DT54      GEN      /N(DT55),11(B$SPEC),23(OLSE),12(ASGPR),24(R3),          00 02834
2835          *****          C17(GPROUT),14(ADD)          00 02835
2836          *****          00 02836
2837          *****          ADD LENGTH AND RETURN RESULT IN R3          00 02837
2838          *****          00 02838
2839          DT55      GEN      /N(COMPOP),11(B$SPEC),23(ORZF),12(ASGPR),24(R3),          00 02839
2840          *****          C17(GPROUT),14(ADD),13(OPROUT)          00 02840
2841          *****          00 02841
2842          *          EJECT          00 02842
2843          *          MULTIPLY INTERRUPT ROUTINE          00 02843
2844          *          00 02844
2845          *          SAVE CONTEXT (R4 THRU RF) IN TEMP AREA          00 02845
2846          *          FLAG INTERRUPTION BY SETTING R0 BIT 15          00 02846
2847          *          EXIT TO INTERRUPT SERVICE          00 02847
2848          *          00 02848
2849          DTIORG EQU      *          00 02849
```

2850	*				00 02850
2851	*	SPECIFY # OF REGISTERS TO STORE THRU SHIFT COUNTER			00 02851
2852	*				00 02852
2853		ORG DT16+1	PLACE ON '7' ADDR FOR INT TEST		00 02853
2854	DT100	GMSK	/N(DT102),11(LIT),15(X'000C),14(TRNB),13(SCOUT)		00 02854
2855		ORG DT1ORG			00 02855
2856	*				00 02856
2857	*	COMPUTE ADDR IN TEMP BLOCK TO STORE REGISTERS			00 02857
2858	*				00 02858
2859	DT102	GMSK	/N(DT103),12(ASGPR),16(RO),11(LIT),15(X'FFBF),		00 02859
2860			C14(ADD),13(POUT)		00 02860
2861	*				00 02861
2862	*	STORE REGISTERS IN TEMP AREA			00 02862
2863	*				00 02863
2864	DT103	GMSK	/P(STR4+P),10(STACK),16(PUSH),15(DT104+P),		00 02864
2865			CLB3,TFO,SFO		00 02865
2866	*				00 02866
2867	*	FETCH P REGISTER FROM TEMP AREA			00 02867
2868	*				00 02868
2869	DT104	GEN	/N(DT105),6(MEMC),10(OF\$ALU),12(ASGPR),24(RO),		00 02869
2870			C14(TRNA),15(LOG)		00 02870
2871	*				00 02871
2872	*	INIT R3 AND R5 TO THEIR V73 VALUES; WAIT FOR P FROM MEM			00 02872
2873	DT105	GEN	/N(DT106),14(ZERO),15(LOG),17(GPROUT),24(R3)		00 02873
2874	DT106	GEN	/N(DT107),14(ONES),17(GPROUT),24(R5),		00 02874
2875			C6(SPEC),10(WAITMD)		00 02875
2876	*				00 02876
2877	*	SETUP P IN RE FOR INTERRUPT SERVICE			00 02877
2878	DT107	GEN	/N(DT108),11(B\$SPEC),23(MIR),14(DECB),		00 02878
2879			C12(AS\$SPEC),22(AONES),17(GPROUT),24(RE)		00 02879
2880	*				00 02880
2881	*	SET RO BIT 15 TO FLAG INTERRUPT			00 02881
2882	*	EXIT			00 02882
2883	*				00 02883
2884	DT108	GMSK	/N(DT109),11(LIT),15(X'7FFF),14(TRNB),13(OPROUT)		00 02884
2885	DT109	GEN	/N(1EXIT),12(ASGPR),24(RO),11(B\$SPEC),23(OPR),		00 02885
2886			C14(OR),17(GPROUT)		00 02886
2887		EJEC			00 02887
2888	*				00 02888
2889	*	MULTIPLY INTERRUPT REENTRY			00 02889
2890	*				00 02890
2891	*	RESTORE CONTEXT (R4 THRU RF) AND CONTINUE MULTIPLY			00 02891
2892	*				00 02892
2893	DTRORG	EQU	*		00 02893
2894	*				00 02894
2895	*	SET # OF REGISTERS TO LOAD			00 02895
2896	*				00 02896
2897		ORG X'1F			00 02897
2898	DTR00	GMSK	/N(DTR01),11(LIT),15(X'000C),14(TRNB),13(SCOUT)		00 02898
2899		ORG DTRORG			00 02899
2900	*				00 02900
2901	*	RESTORE RO			00 02901
2902	*				00 02902
2903	DTR01	GEN	/N(DTR02),11(B\$SPEC),23(OPR),14(TRNB),15(LOG),		00 02903
2904			C17(GPROUT),24(RO)		00 02904
2905	*				00 02905
2906	*	SET ADDRESS IN TEMP AREA TO RECOVER REGISTERS			00 02906

2907	*			00 02907
2908	DTR02	GMSK	/N(DTR03),12(ASGPR),16(RO),11(LIT),15(X'FFBF),	00 02908
2909			C14(ADD),13(POUT)	00 02909
2910	*			00 02910
2911	*	LOAD R4 THRU RE FROM TEMP AREA		00 02911
2912	*	EXIT BACK INTO MULTIPLY		00 02912
2913	*			00 02913
2914	DTR03	GMSK	/P(LDR4+P),10(STACK),16(PUSH),15(DT12+P),	00 02914
2915			CLB3,SFO,TFO	00 02915
2916		EJEC		00 02916
2917	*			00 02917
2918	*	DIVIDE		00 02918
2919	*			00 02919
2920	*	STORE REGISTERS 0 THRU E AND RETURN TO SOFTWARE		00 02920
2921	*			00 02921
2922	DDORG	EQU	*	00 02922
2923	*			00 02923
2924	*	SET # OF REGISTERS TO SAVE		00 02924
2925	*			00 02925
2926		ORG	X'5C	00 02926
2927	DD00	GMSK	/N(DD01),11(LIT),15(X'F),14(TRNB),13(SCOUT)	00 02927
2928	*			00 02928
2929	*	SET ADDR IN TEMP FOR REG STORAGE		00 02929
2930	*			00 02930
2931		ORG	X'78	00 02931
2932	DD01	GEN	/N(DD02),12(ASGPR),24(RO),14(INCA),16(CRY1),13(POUT)	00 02932
2933	*			00 02933
2934	*	CALL STORE REGISTERS SUBROUTINE		00 02934
2935	*			00 02935
2936		ORG	X'123	00 02936
2937	DD02	GMSK	/P(STRO+P),10(STACK),16(PUSH),15(DD03+P),	00 02937
2938			CSFO,TFO,LB3	00 02938
2939	*			00 02939
2940	*	CLEAR R2 TO FLAG DIVIDE; EXIT		00 02940
2941	*			00 02941
2942		ORG	X'2C	00 02942
2943	DD03	GEN	/N(EX30),14(ZERO),15(LOG),17(GPROUT),24(R2)	00 02943
2944		ORG	DDORG	00 02944
2945		EJEC		00 02945
2946	*			00 02946
2947	*	COMPARE-SUBTRACT		00 02947
2948	*			00 02948
2949	*	COMPARE OPA TO OPB AND SUBTRACT IF B NOT > A		00 02949
2950	*	IF B>A, CLEAR R2 TO FLAG IT		00 02950
2951	*			00 02951
2952	DDORG2	EQU	*	00 02952
2953	*			00 02953
2954	*	COMPARE OPERANDS		00 02954
2955	*			00 02955
2956		ORG	X'38	00 02956
2957	DD10	GMSK	/P(DPO0+P),10(STACK),15(DD11+P),LB3,TFO,SFO,	00 02957
2958			C12(ASGPR),16(RB),14(TRNA),13(OPROUT)	00 02958
2959	*			00 02959
2960	*	IF A > OR = B, JUMP TO SUBTRACT; ELSE EXIT		00 02960
2961	*			00 02961
2962		ORG	X'EA	00 02962
2963	DD11	GEN	/T(X'24,DD03),5(FT),7(ALUS)	00 02963

```
2964      ORG      DDORG2      00 02964
2965      EJECT      00 02965
2966      *          00 02966
2967      *          32 BIT INTEGER MATH ROUTINES      00 02967
2968      *          *****      00 02968
2969      *          00 02969
2970      *          THESE ROUTINES IMPLEMENT ADD, SUBTRACT, MULTIPLY, AND DIVIDE      00 02970
2971      *          FOR 32 BIT INTEGER OPERANDS.      00 02971
2972      *          00 02972
2973      *          CALLING SEQUENCE; A REGISTER (R0) POINTS TO FOUR WORD PARAMETER      00 02973
2974      *          BLOCK WHICH CONTAINS THE TWO OPERANDS.      00 02974
2975      *          00 02975
2976      *          BCS VALUES;      00 02976
2977      *          0105023 : OP1+OP2 REPLACES OP2;      00 02977
2978      *          0105063 : OP1-OP2 REPLACES OP2;      00 02978
2979      *          0105123 : OP1*OP2 REPLACES OP1 AND OP2;      00 02979
2980      *          0105163 : OP1/OP2 REPLACES OP1; REMAINDER REPLACES OP2.      00 02980
2981      *          00 02981
2982      *          MULTIPLY ALGORITHM; QUADRUPLE LENGTH SHIFT AND ADD.      00 02982
2983      *          00 02983
2984      *          DIVIDE ALGORITHM; IF FIRST DIVISOR WORD ZERO, 16 BIT SUBTRACT      00 02984
2985      *          AND TRIPLE LENGTH SHIFT THROUGH 32 ITERATIONS;      00 02985
2986      *          IF FIRST DIVISOR WORD NON-ZERO, 32 BIT SUBTRACT AND      00 02986
2987      *          TRIPLE LENGTH SHIFT THROUGH 16 ITERATIONS.      00 02987
2988      *          00 02988
2989      *          00 02989
2990      *          SPAC      00 02990
2991      *          TA01 EQU      *      00 02991
2992      *          00 02992
2993      *          SAVE P REG      00 02993
2994      *          00 02994
2995      *          ORG      X'D8      00 02995
2996      *          T100 GEN      /N(T101),12(ASP),14(TRNA),15(LOG),      00 02996
2997      *          C17(GPROUT),24(R7)      00 02997
2998      *          00 02998
2999      *          START FETCH OF MS(OP1)      00 02999
3000      *          SET P TO PARAM BLOCK ADDR      00 03000
3001      *          00 03001
3002      *          ORG      1      00 03002
3003      *          T101 GEN      /N(T102),6(MEMC),10(OF$ALU),12(ASGPR),24(R0),      00 03003
3004      *          C14(TRNA),15(LOG),7(ROVFL),13(POUT)      00 03004
3005      *          00 03005
3006      *          START FETCH OF LS(OP1)      00 03006
3007      *          00 03007
3008      *          T102 GEN      /N(T103),6(MEMC),10(OF$ALU),12(AS$SPEC),22(AZERO),      00 03008
3009      *          C11(B$GPR),23(R0),14(INCB),16(CRY1),17(GPROUT),24(RF)      00 03009
3010      *          00 03010
3011      *          SAVE MS(OP1) IN R8      00 03011
3012      *          00 03012
3013      *          T103 GEN      /N(T104),11(B$SPEC),23(MIR),14(TRNB),15(LOG),      00 03013
3014      *          C17(GPROUT),24(R8)      00 03014
3015      *          00 03015
3016      *          START FETCH MS(OP2)      00 03016
3017      *          00 03017
3018      *          T104 GEN      /N(T105),6(MEMC),10(OF$ALU),12(ASGPR),24(RF),      00 03018
3019      *          C14(INCA),16(CRY1),17(GPROUT)      00 03019
3020      *          00 03020
```

```
3021 *      SAVE LS(OP1) IN R9 AND OPR      00 03021
3022 *                                     00 03022
3023 T105  GEN      /N(T106),11(B$SPEC),23(MIR),14(TRNB),15(LOG), 00 03023
3024          C17(GPROUT),24(R9),13(OPROUT) 00 03024
3025 *                                     00 03025
3026 *      START FETCH LS(OP2)             00 03026
3027 *                                     00 03027
3028 T106  GEN      /N(T107),6(MEMC),10(OF$ALU),12(A$GPR),24(RF), 00 03028
3029          C14(INCA),16(CRY1),17(GPROUT) 00 03029
3030 *                                     00 03030
3031 *      SAVE MS(OP2) IN RA              00 03031
3032 *                                     00 03032
3033 T107  GEN      /N(T108),11(B$SPEC),23(MIR),14(TRNB),15(LOG), 00 03033
3034          C17(GPROUT),24(RA)            00 03034
3035 *                                     00 03035
3036 *      WAIT FOR LS(OP2); CLEAR RD FOR 00 03036
3037 *      LATER USE.                     00 03037
3038 T108  GEN      /N(T109),6(SPEC),10(WAITMD), 00 03038
3039          C12(A$SPEC),22(AZERO),14(TRNA),15(LOG), 00 03039
3040          C17(GPROUT),24(RD)            00 03040
3041 *                                     00 03041
3042 *      SAVE LS(OP2) IN RB; FIELD SELECT 00 03042
3043 *      ADD=SUBTRACT OR MULTIPLY-DIVIDE. 00 03043
3044 *                                     00 03044
3045 T109  GEN      /F(TAO1),2(1),FSA,11(B$SPEC),23(MIR), 00 03045
3046          C14(TRNB),15(LOG),17(GPROUT),24(RB) 00 03046
3047 *                                     00 03047
3048 *      LEAVE SPACE FOR FIRST ADD-SUBTRACT MICRO 00 03048
3049 **      (THIS MICRO ADDR MUST BE EVEN) 00 03049
3050 *                                     00 03050
3051      ORG      TAO1 00 03051
3052      ORG      *+1 00 03052
3053 *                                     00 03053
3054 *      CLEAR RE FOR LATER USE;          00 03054
3055 *      FIELD SELECT BRANCH TO CHOOSE MULTIPLY OR DIVIDE; 00 03055
3056 **      (THIS MICRO ADDR MUST BE AT LOCATION TAO1 +1); 00 03056
3057 *                                     00 03057
3058 T110  GEN      /F(TMO1),2(1),FS9,12(A$SPEC),22(AZERO), 00 03058
3059          C14(TRNA),15(LOG),17(GPROUT),24(RE) 00 03059
3060      EJEC      00 03060
3061 *                                     00 03061
3062 *      FIRST ADD-SUBTRACT MICRO         00 03062
3063 *                                     00 03063
3064      SPAC      00 03064
3065 *                                     00 03065
3066 *      START STORE OF LS WORD OF RESULT; 00 03066
3067 *      FIELD SELECT TO ADD OR SUBTRACT; 00 03067
3068 *                                     00 03068
3069 TAO2A  EQU      *      SAVE CURRENT LOCATION 00 03069
3070      ORG      TAO1  PLACE IN FIELD SELECT BLOCK 00 03070
3071 TAO1   GEN      /F(TAO2A),2(1),FS9,6(MEMC),10(OF$ALU), 00 03071
3072          C12(A$GPR),24(RF),14(TRNA),15(LOG) 00 03072
3073      ORG      TAO2A  RESTORE LOCATION 00 03073
3074 *                                     00 03074
3075 *      STORE LS(OP1)+LS(OP2), IE R9+RB; 00 03075
3076 **      (THIS MICRO ADDR MUST BE EVEN) 00 03076
3077 *                                     00 03077
```

```

3078 TA02A GEN /N(TA03),12(ASGPR),24(R9),11(BSGPR),23(RB),14(ADD), 00 03078
3079 CTFO,SFO,7(SSALU),10(WAITMD) 00 03079
3080 * 00 03080
3081 * STORE LS(OP1)-LS(OP2), IE R9*RB; 00 03081
3082 * 00 03082
3083 TA02S GEN /N(TA03),12(ASGPR),24(R9),11(BSGPR),23(RB), 00 03083
3084 C14(SUB),16(CRY1),TF0,SFO,7(SSALU),10(WAITMD) 00 03084
3085 * 00 03085
3086 * START STORE OF MS RESULT WORD; 00 03086
3087 * FIELD SELECT TO ADD OR SUBTRACT ; 00 03087
3088 * 00 03088
3089 ORG *+2 LEAVE ROOM FOR TA04'S TO BE EVEN 00 03089
3090 TA03 GEN /F(TA04A),2(1),FS9,6(MEMC),10(OS$ALU), 00 03090
3091 C12(ASGPR),24(RF),14(DECA) 00 03091
3092 * 00 03092
3093 * STORE MS(OP1)+ MS(OP2) + CARRY; 00 03093
3094 * (THIS MICRO ADDR MUST BE EVEN); 00 03094
3095 * 00 03095
3096 ORG *+3 PLACE ON EVEN TWO WORD BLOCK 00 03096
3097 TA04A GEN /N(TA05A),12(ASGPR),24(R8),11(BSGPR),23(RA), 00 03097
3098 C14(ADD),CF1,SFO,TF0,7(SSALU),10(WAITMD) 00 03098
3099 * 00 03099
3100 * STORE MS(OP1) - MS(OP2) + CARRY; 00 03100
3101 * 00 03101
3102 TA04S GEN /N(TA05S),12(ASGPR),24(R8),11(BSGPR),23(RA), 00 03102
3103 C14(SUB),CF1,SFO,TF0,7(SSALU),10(WAITMD) 00 03103
3104 ORG *+1 SKIP RELOCATED TA03 00 03104
3105 * 00 03105
3106 * IF ADD PRODUCED CARRY, JUMP TO FLAG OVERFLOW; ELSE EXIT 00 03106
3107 * 00 03107
3108 TA05A GEN /T(TM19,TA06),5(FT),7(ALUC) 00 03108
3109 * 00 03109
3110 * IF SUBTRACT DID NOT PRODUCE CARRY, JUMP TO FLAG OVERFLOW; 00 03110
3111 * 00 03111
3112 ORG X'A 00 03112
3113 TA05S GEN /T(TM19,TA06),5(TT),7(ALUC) 00 03113
3114 * 00 03114
3115 * SET OVERFLOW 00 03115
3116 * 00 03116
3117 ORG X'3E 00 03117
3118 TA06 GEN /N(TM19),TF0,SF1,GF2 00 03118
3119 ORG TA05A+1 00 03119
3120 EJE 00 03120
3121 * 00 03121
3122 * MULTIPLY ROUTINE 00 03122
3123 * 00 03123
3124 * RC,RD,RE, AND RF USED AS QUADRUPLE LENGTH ACCUMULATOR; 00 03124
3125 * 00 03125
3126 SPAC S00 03126
3127 * 00 03127
3128 * CLEAR RF 00 03128
3129 ** (THIS LOCATION MUST BE EVEN) 00 03129
3130 * 00 03130
3131 TM01 GEN /N(TM02),12(AS$PEC),22(AZERO),14(TRNA),15(LOG), 00 03131
3132 C17(GPROUT),24(RF) 00 03132
3133 ORG *+1 SKIP LOCATION FOR TD01 00 03133
3134 * 00 03134

```


3135	*	LOAD SHIFT COUNTER;	00 03135
3136	*		00 03136
3137	TM02	GMSK /N(TM03),11(LIT),15(X'001F),	00 03137
3138		C14(TRNB),13(SCOUT)	00 03138
3139	*		00 03139
3140	*	SHIFT FOURTH WORD OF ACCUMULATOR LEFT	00 03140
3141	*		00 03141
3142	TM03	GEN /N(TM04),12(ASGPRL),24(RF),VF1,SH0,	00 03142
3143		C14(TRNA),15(LOG),17(GPROUT)	00 03143
3144		ORG *+1 LEAVE ROOM FOR TM17 TEST ADDR	00 03144
3145	*		00 03145
3146	*	SHIFT THIRD WORD OF ACCUMULATOR LEFT;	00 03146
3147	*	TEST SHFT FLAG TO SEE IF NEW BIT 0 SHOULD BE A ONE.	00 03147
3148	*		00 03148
3149	TM04	GEN /T(TM06,TM05),5(FT),7(SHFT),	00 03149
3150		C12(ASGPRL),24(RE),VF1,SH0,	00 03150
3151		C14(TRNA),15(LOG),17(GPROUT)	00 03151
3152	*		00 03152
3153	*	SET BIT 0 OF THIRD ACCUMULATOR WORD TO A ONE.	00 03153
3154	**	(THIS MICRO ADDRESS MUST BE EVEN);	00 03154
3155	*		00 03155
3156	TM05	GEN /N(TM06),12(ASGPR),24(RE),14(INCA),16(CRY1),	00 03156
3157		C17(GPROUT)	00 03157
3158	*		00 03158
3159	*	SHIFT SECOND ACCUMULATOR WORD LEFT ;	00 03159
3160	*	CHECK SHFT FLAG TO SEE IF BIT 0 SHOULD BE A ONE;	00 03160
3161	*		00 03161
3162	TM06	GEN /T(TM08,TM07),5(FT),7(SHFT),	00 03162
3163		C12(ASGPRL),24(RD),VF1,SH0,	00 03163
3164		C14(TRNA),15(LOG),17(GPROUT)	00 03164
3165	*		00 03165
3166	*	SET BIT 0 OF SECOND ACCUMULATOR WORD TO A ONE;	00 03166
3167	**	(THIS ADDRESS MUST BE EVEN);	00 03167
3168	*		00 03168
3169	TM07	GEN /N(TM08),12(ASGPR),24(RD),	00 03169
3170		C14(INCA),16(CRY1),17(GPROUT)	00 03170
3171	*		00 03171
3172	*	SHIFT FIRST ACCUMULATOR WORD LEFT;	00 03172
3173	*	CHECK SHFT FLAG TO SEE IF BIT 0 SHOULD BE A ONE;	00 03173
3174	*		00 03174
3175	TM08	GEN /T(TM10,TM09),5(FT),7(SHFT),	00 03175
3176		C12(ASGPRL),24(RC),14(TRNA),15(LOG),17(GPROUT)	00 03176
3177	*		00 03177
3178	*	SET BIT 0 OF FIRST ACCUMULATOR WORD TO ONE;	00 03178
3179	**	(THIS MICRO ADDR MUST BE EVEN);	00 03179
3180	*		00 03180
3181	TM09	GEN /N(TM10),12(ASGPR),24(RC),14(INCA),16(CRY1),	00 03181
3182		C17(GPROUT)	00 03182
3183	*		00 03183
3184	*	DOUBLE SHIFT LEFT OP1, (SHFT,R8,OPR).	00 03184
3185	*		00 03185
3186	TM10	GEN /N(TM11),12(ASGPRL),24(R8),VF1,SH2,14(TRNA),15(LOG),	00 03186
3187		C17(GPROUT),18(SHFTOP),20(LFT),XF3	00 03187
3188	*		00 03188
3189	*	IF OLD MSB OF OP1 (IE SHFT) WAS A ZERO, SKIP ADD OF	00 03189
3190	*	OP2 INTO QUAD ACCUMULATOR;	00 03190
3191	*		00 03191


```

3192 TM11 GEN /T(TM12,TM16),5(TT),7(SHFT) 00 03192
3193 * 00 03193
3194 * ADD LS(OP2) INTO FOURTH ACCUMULATOR WORD 00 03194
3195 * 00 03195
3196 TM12 GEN /N(TM13),12(ASGPR),24(RF),11(BSGPR),23(RB), 00 03196
3197 C14(ADD),17(GPROUT),TF0,SF0,7(SSALU) 00 03197
3198 * 00 03198
3199 * ADD MS(OP2) INTO THIRD ACCUMULATOR WORD,(WITH PREV. CARRY). 00 03199
3200 * 00 03200
3201 TM13 GEN /N(TM14),12(ASGPR),24(RE),11(BSGPR),23(RA), 00 03201
3202 C14(ADD),CF1,17(GPROUT),TF0,SF0,7(SSALU) 00 03202
3203 * 00 03203
3204 * PROPAGATE CARRY INTO SECOND ACCUMULATOR WORD 00 03204
3205 * 00 03205
3206 TM14 GEN /N(TM15),12(ASPEC),22(AZERO),11(BSGPR),23(RD), 00 03206
3207 C14(ADD),CF1,17(GPROUT),TF0,SF0,7(SSALU),24(RD) 00 03207
3208 * 00 03208
3209 * PROPAGATE CARRY INTO FIRST ACCUMULATOR WORD 00 03209
3210 * 00 03210
3211 ORG *+1 LEAVE SPACE FOR TM16 TO BE EVEN 00 03211
3212 TM15 GEN /N(TM16),12(ASPEC),22(AZERO),11(BSGPR),23(RC), 00 03212
3213 C14(ADD),CF1,17(GPROUT),TF0,SF0,7(SSALU),24(RC) 00 03213
3214 * 00 03214
3215 * INCR SHIFT COUNTER AND CHECK IF ITERATED ENOUGH; 00 03215
3216 ** (THIS ADDRESS MUST BE EVEN) 00 03216
3217 * 00 03217
3218 ORG *-2 PLACE ON EVEN WORD 00 03218
3219 TM16 GEN /T(TMO3,TM17),5(FT),7(SFTC),13(INCSC) 00 03219
3220 * 00 03220
3221 * SAVE RC THRU RF BACK INTO PARAMETER BLOCK 00 03221
3222 * 00 03222
3223 ORG TMO3+1 PLACE FOR TEST ADDRESSING 00 03223
3224 TM17 GMSK /N(TM18),11(LIT),15(X'4),14(TRNB),13(SCOUT) 00 03224
3225 ORG TM15+1 RESTORE LOCATION COUNTER 00 03225
3226 TM18 GMSK /P(STRC+P),10(STACK),16(PUSH),15(TM19+P), 00 03226
3227 CLB3,TF0,SF0 00 03227
3228 * 00 03228
3229 * RESTORE P AND RETURN TO ROM 00 03229
3230 * 00 03230
3231 ORG X'3F 00 03231
3232 TM19 GEN /P(SSIM),10(PJMP),SF0,TF0,12(ASGPR),24(R7), 00 03232
3233 C14(TRNA),15(LOG),13(POUT) 00 03233
3234 TM25 EQU TM19 00 03234
3235 ORG TM18+1 00 03235
3236 EJEK 00 03236
3237 * 00 03237
3238 * DIVIDE ROUTINE 00 03238
3239 * 00 03239
3240 * RD,R8, AND R9 USED AS TRIPLE LENGTH ACCUMULATOR FOR SHIFTING; 00 03240
3241 * WHEN FIRST DIVISOR WORD ZERO, QUOTIENT BUILT IN OPR AND RE; 00 03241
3242 * OTHERWISE, QUOTIENT BUILT IN OPR ONLY; 00 03242
3243 * 00 03243
3244 SPAC 1 00 03244
3245 TD31 EQU * 00 03245
3246 TD02 EQU *+1 00 03246
3247 * 00 03247
3248 * SAMPLE MS WORD OF DIVISOR FOR ZERO; 00 03248

```

3249	*			00 03249	
3250		ORG	TM01+1	PLACE TD01 IN FIELD SELECT TABLE	00 03250
3251	TD01	GEN	/N(TD02),12(ASGPR),24(RA),14(TRNA),15(LOG),		00 03251
3252			CTFO,SFO,7(SSALU)		00 03252
3253		ORG	TD02	RESTORE LOCATION	00 03253
3254	*				00 03254
3255	*		CHOOSE ALGORITHM BASED ON MS DIVISOR WORD ZERO OR NOT;		00 03255
3256	*				00 03256
3257	TD02	GEN	/T(TD03,TD30),5(FT),7(ALUZ)		00 03257
3258	*				00 03258
3259	*		SET SHIFT COUNTER TO -17; MS DIVISOR NOT ZERO;		00 03259
3260	*				00 03260
3261		ORG	*+2	LEAVE SPACE FOR TD30 AND ALGN FOR TEST	00 03261
3262	TD03	GMSK	/N(TD04),11(LIT),15(X'0010),		00 03262
3263			C14(TRNB),13(SCOUT)		00 03263
3264	*				00 03264
3265	*		SHIFT WORD 3 OF ACCUMULATOR LEFT;		00 03265
3266	*		ALSO SHIFT QUOTIENT (OPR) LEFT;		00 03266
3267	*				00 03267
3268	TD04	GEN	/N(TD05),12(ASGPRL),24(R9),VF1,SHO,		00 03268
3269			C14(TRNA),15(LOG),17(GPROUT),		00 03269
3270			C18(SHFTOP),20(LFT),XF3		00 03270
3271	*				00 03271
3272	*		SHIFT WORD 2 OF ACCUMULATOR LEFT;		00 03272
3273	*		TEST SHFT FLAG TO SEE IF NEW BIT 0 SHOULD BE A ONE;		00 03273
3274	*				00 03274
3275	TD05	GEN	/T(TD07,TD06),5(FT),7(SHFT),12(ASGPRL),24(R8),		00 03275
3276			CVF1,SHO,14(TRNA),15(LOG),17(GPROUT)		00 03276
3277	*				00 03277
3278	*		SET BIT 0 OF ACCUMULATOR WORD 2 TO A ONE;		00 03278
3279	**		(THIS MICRO ADDRESS MUST BE EVEN)		00 03279
3280	*				00 03280
3281	TD06	GEN	/N(TD07),12(ASGPR),24(R8),		00 03281
3282			C14(INCA),16(CRY1),17(GPROUT)		00 03282
3283	*				00 03283
3284	*		SHIFT FIRST ACCUMULATOR WORD LEFT;		00 03284
3285	*		CHECK SHFT FLAG TO SEE IF BIT 0 SHOULD BE A ONE;		00 03285
3286	*				00 03286
3287	TD07	GEN	/T(TD09,TD08),5(FT),7(SHFT),		00 03287
3288			C12(ASGPRL),24(RD),14(TRNA),15(LOG),		00 03288
3289			C17(GPROUT)		00 03289
3290	*				00 03290
3291	*		SET BIT 0 OF ACCUMULATOR WORD 1 TO A ONE;		00 03291
3292	**		(THIS ADDRESS MUST BE EVEN)		00 03292
3293	*				00 03293
3294	TD08	GEN	/N(TD09),12(ASGPR),24(RD),		00 03294
3295			C14(INCA),16(CRY1),17(GPROUT)		00 03295
3296	*				00 03296
3297	*		CHECK IF DIVISOR GREATER THAN ACCUMULATOR WORDS 1 AND 2;		00 03297
3298	*		COMPARE LS WORDS;		00 03298
3299	*				00 03299
3300	TD09	GEN	/N(TD10),12(ASGPR),24(R8),11(BSGPR),23(RB),		00 03300
3301			C14(SUB),16(CRY1),17(GPROUT),		00 03301
3302			CSFO,TFO,7(SSALU)		00 03302
3303	*				00 03303
3304	*		COMPARE MS WORDS; (CONSIDER PREV. CARRY);		00 03304
3305	*				00 03305

3306	TD10	GEN	/N(TD11),12(ASGPR),24(RD),11(BSGPR),23(RA),	00	03306
3307			C14(SUB),CF1,17(GPROUT),	00	03307
3308			CSFO,TF0,7(SSALU)	00	03308
3309	*			00	03309
3310	*		IF NO CARRY, JUMP TO ADD DIVISOR BACK IN TO ACCUMULATOR;	00	03310
3311	*		ELSE, JUMP TO LOG DIVISION IN QUOTIENT;	00	03311
3312	*		ALSO INCREMENT SHIFT COUNTER	00	03312
3313	*			00	03313
3314	TD11	GEN	/T(TD12B,TD12A),5(FT),7(ALUC),13(INCSC)	00	03314
3315	*			00	03315
3316	*		SET BIT IN QUOTIENT TO FLAG DIVISION;	00	03316
3317	*		ALSO CHECK IF ITERATED ENOUGH	00	03317
3318	*			00	03318
3319	TD12A	GEN	/T(TD15,TD04),5(TT),7(SFTC),	00	03319
3320			C11(BSSPEC),23(OPR),12(ASPEC),22(AZERO),	00	03320
3321			C14(INCB),16(CRY1),13(OPROUT)	00	03321
3322	*			00	03322
3323	*		ADD DIVISOR BACK INTO FIRST TWO ACCUMULATOR WORDS;	00	03323
3324	*		ALSO CHECK IF ITERATED ENOUGH	00	03324
3325	*			00	03325
3326	TD12B	GEN	/N(TD13),12(ASGPR),24(RB),11(BSGPR),23(RB),	00	03326
3327			C14(ADD),17(GPROUT),TF0,SFO,7(SSALU)	00	03327
3328	TD13	GEN	/T(TD15,TD04),5(TT),7(SFTC),	00	03328
3329			C12(ASGPR),24(RD),11(BSGPR),23(RA),	00	03329
3330			C14(ADD),CF1,17(GPROUT)	00	03330
3331	*			00	03331
3332	*		SETUP LS(REMAINDER) FOR STORE;	00	03332
3333	*			00	03333
3334	TD15	GEN	/N(TD16),11(BSGPR),23(RB),14(TRNB),15(LOG),	00	03334
3335			C17(GPROUT),24(RF)	00	03335
3336	*			00	03336
3337	*		SETUP MS(REMAINDER) FOR STORE;	00	03337
3338	*			00	03338
3339	TD16	GEN	/N(TD17),11(BSGPR),23(RD),14(TRNB),15(LOG),	00	03339
3340			C17(GPROUT),24(RE)	00	03340
3341	*			00	03341
3342	*		SETUP LS(QUOTIENT) FOR STORE;	00	03342
3343	*			00	03343
3344	TD17	GEN	/N(TD18),11(BSSPEC),23(OPR),14(TRNB),15(LOG),	00	03344
3345			C17(GPROUT),24(RD)	00	03345
3346	*			00	03346
3347	*		SETUP MS(QUOTIENT) FOR STORE; JUMP TO STORE;	00	03347
3348	*			00	03348
3349	TD18	GEN	/N(TD17),12(ASPEC),22(AZERO),14(TRNA),15(LOG),	00	03349
3350			C17(GPROUT),24(RC)	00	03350
3351		EJEC		00	03351
3352	*			00	03352
3353	*		DIVIDE WITH MS(DIVISOR) = 0	00	03353
3354	*			00	03354
3355	*		SPAC	00	03355
3356	*			00	03356
3357	*		SAMPLE LS(DIVISOR) TO SEE IF DIVIDING BY ZERO	00	03357
3358	*			00	03358
3359	ORG	TD03-1	PLACE TD30 FOR TEST ADDRESSING	00	03359
3360	TD30	GEN	/N(TD31),12(ASGPR),24(RB),14(TRNA),15(LOG),	00	03360
3361			CTFO,SFO,7(SSALU)	00	03361
3362	*			00	03362

```
3363 * SET SHIFT COUNTER TO -32; 00 03363
3364 * IF ATTEMPTING TO DIVIDE BY ZERO, JUMP TO SET OVERFLOW; 00 03364
3365 * 00 03365
3366 ORG TD31 RESTORE LOCATION COUNTER 00 03366
3367 TD31 GMSK /T(TDOVF,TD32),5(TT),7(ALUZ), 00 03367
3368 C11(LIT),15(X'001F),14(TRNB),13(SCOUT) 00 03368
3369 ORG TD18+1 00 03369
3370 * 00 03370
3371 * FLAG OVERFLOW AND JUMP TO EXIT; 00 03371
3372 * 00 03372
3373 TDOVF GEN /N(TM25),TF0,SF1,GF2 00 03373
3374 * 00 03374
3375 * SHIFT WORD 3 OF ACCUMULATOR LEFT 00 03375
3376 * 00 03376
3377 TD32 GEN /N(TD33),12(ASGPRL),24(R9),VF1,SH0, 00 03377
3378 C14(TRNA),15(LOG),17(GPROUT) 00 03378
3379 * 00 03379
3380 * SHIFT QUOTIENT (IE OPR,RE) LEFT ONE; 00 03380
3381 * CHECK IF PREVIOUS SHIFT MSB WAS ZERO OR ONE; 00 03381
3382 * 00 03382
3383 TD33 GEN /T(TD34B,TD34A),5(TT),7(SHFT), 00 03383
3384 C12(ASGPRL),24(RE),14(TRNA),15(LOG), 00 03384
3385 C17(GPROUT),18(SHFTOP),20(LFT),XF1 00 03385
3386 * 00 03386
3387 * SHIFT SECOND ACCUMULATOR WORD LEFT; 00 03387
3388 * SHIFT ZERO INTO LSB; 00 03388
3389 ** (THIS MICRO ADDR MUST BE EVEN); 00 03389
3390 * 00 03390
3391 TD34A GEN /N(TD35),12(ASGPRL),24(R8),VF1,SH0, 00 03391
3392 C14(TRNA),15(LOG),17(GPROUT) 00 03392
3393 * 00 03393
3394 * SHIFT SECOND ACCUMULATOR WORD LEFT; 00 03394
3395 * SHIFT ONE INTO LSB; 00 03395
3396 * 00 03396
3397 TD34B GEN /N(TD35),12(ASGPRL),24(R8),VF1,SH0, 00 03397
3398 C14(INCA),16(CRY1),17(GPROUT) 00 03398
3399 * 00 03399
3400 * SHIFT FIRST ACCUMULATOR WORD LEFT; 00 03400
3401 * CHECK IF LSB SHOULD BE SET TO ONE; 00 03401
3402 * 00 03402
3403 TD35 GEN /T(TD36,TD37),5(TT),7(SHFT), 00 03403
3404 C12(ASGPRL),24(RD),VF1,SH0, 00 03404
3405 C14(TRNA),15(LOG),17(GPROUT) 00 03405
3406 * 00 03406
3407 * SET LSB OF FIRST ACCUMULATOR WORD TO ONE; 00 03407
3408 * 00 03408
3409 TD36 GEN /N(TD37),12(ASGPR),24(RD), 00 03409
3410 C14(INCA),16(CRY1),17(GPROUT) 00 03410
3411 * 00 03411
3412 * SUBTRACT DIVISOR FROM FIRST ACCUMULATOR WORD; 00 03412
3413 ** (THIS ADDR MUST BE EVEN) 00 03413
3414 * 00 03414
3415 TD37 GEN /N(TD38),12(ASGPR),24(RD),11(BSGPR),23(RB), 00 03415
3416 C14(SUB),16(CRY1),17(GPROUT),SF0,TF0,7(SSALU) 00 03416
3417 * 00 03417
3418 * IF CARRY FROM SUBTRACT, JUMP TO UPDATE QUOTIENT; 00 03418
3419 * 00 03419
```

3420	TD38	GEN	/T(TD40A,TD39),5(TT),7(ALUC)	00 03420
3421	*			00 03421
3422	*	NO CARRY, SO CHECK MSB PRESERVED IN SHFT;		00 03422
3423	*	IF BIT SET, JUMP TO UPDATE QUOTIENT;		00 03423
3424	*	ELSE, JUMP TO ADD DIVISOR BACK IN TO ACCUMULATOR;		00 03424
3425	**	(THIS ADDRESS MUST BE EVEN)		00 03425
3426	*			00 03426
3427	TD39	GEN	/T(TD40A,TD40B),5(TT),7(SHFT)	00 03427
3428	*			00 03428
3429	*	UPDATE QUOTIENT ; ALSO COUNT SHIFT COUNTER;		00 03429
3430	*	CHECK IF ITERATED ENOUGH;		00 03430
3431	*			00 03431
3432	TD40A	GEN	/T(TD41,TD32),5(TT),7(SFTC),13(INCSC),	00 03432
3433			C12(ASGPR),24(RE),14(INCA),16(CRY1),17(GPROUT)	00 03433
3434	*			00 03434
3435	*	ADD DIVISOR BACK INTO ACCUMULATOR;		00 03435
3436	*	ALSO COUNT SHIFT COUNTER; CHECK IF ITERATED ENOUGH.		00 03436
3437	*	(THIS ADDRESS MUST BE EVEN);		00 03437
3438	*			00 03438
3439	TD40B	GEN	/T(TD41,TD32),5(TT),7(SFTC),13(INCSC),	00 03439
3440			C12(ASGPR),24(RD),11(BSGPR),23(RB),	00 03440
3441			C14(ADD),17(GPROUT)	00 03441
3442	*			00 03442
3443	*	SETUP LS(REMAINDER) FOR STORE		00 03443
3444	*			00 03444
3445	TD41	GEN	/N(TD42),11(BSGPR),23(RD),14(TRNB),15(LOG),	00 03445
3446			C17(GPROUT),24(RF)	00 03446
3447	*			00 03447
3448	*	SETUP LS(QUOTIENT) FOR STORE		00 03448
3449	*			00 03449
3450	TD42	GEN	/N(TD43),11(BSGPR),23(RE),14(TRNB),15(LOG),	00 03450
3451			C17(GPROUT),24(RD)	00 03451
3452	*			00 03452
3453	*	SETUP MS(REMAINDER) FOR STORE		00 03453
3454	*			00 03454
3455	TD43	GEN	/N(TD44),12(ASPEC),22(AZERO),14(TRNA),15(LOG),	00 03455
3456			C17(GPROUT),24(RE)	00 03456
3457	*			00 03457
3458	*	SETUP MS(QUOTIENT) FOR STORE; JUMP TO STORE;		00 03458
3459	*			00 03459
3460		ORG	TD02+1	00 03460
3461	TD44	GEN	/N(TM17),11(BSPEC),23(OPR),14(TRNB),15(LOG),	00 03461
3462			C17(GPROUT),24(RC)	00 03462
3463		END		00 03463

CPU:0849 CTP:081 SUPS:15.480

@A*PS,PRESS,CO PFFFILE.WCSCOMEN,PFFFILE.WCSCOMEN

PRESS 09/15/77

CPU:0294 CTP:004 SUPS:2.958

@HDG,P

SOURCE DECK: 42;

WCSCOMEN

@PRT,S PFFFILE.WCSCOMEN

FURPUR 27-S 08/16/78 10:13:17