

KE44A

CIS INSTR EXER
CZKEEA0

AH-F227A-MC

COPYRIGHT 1980

FICHE 1 OF 2

JAN 1980

digital

MADE IN USA

KE44A

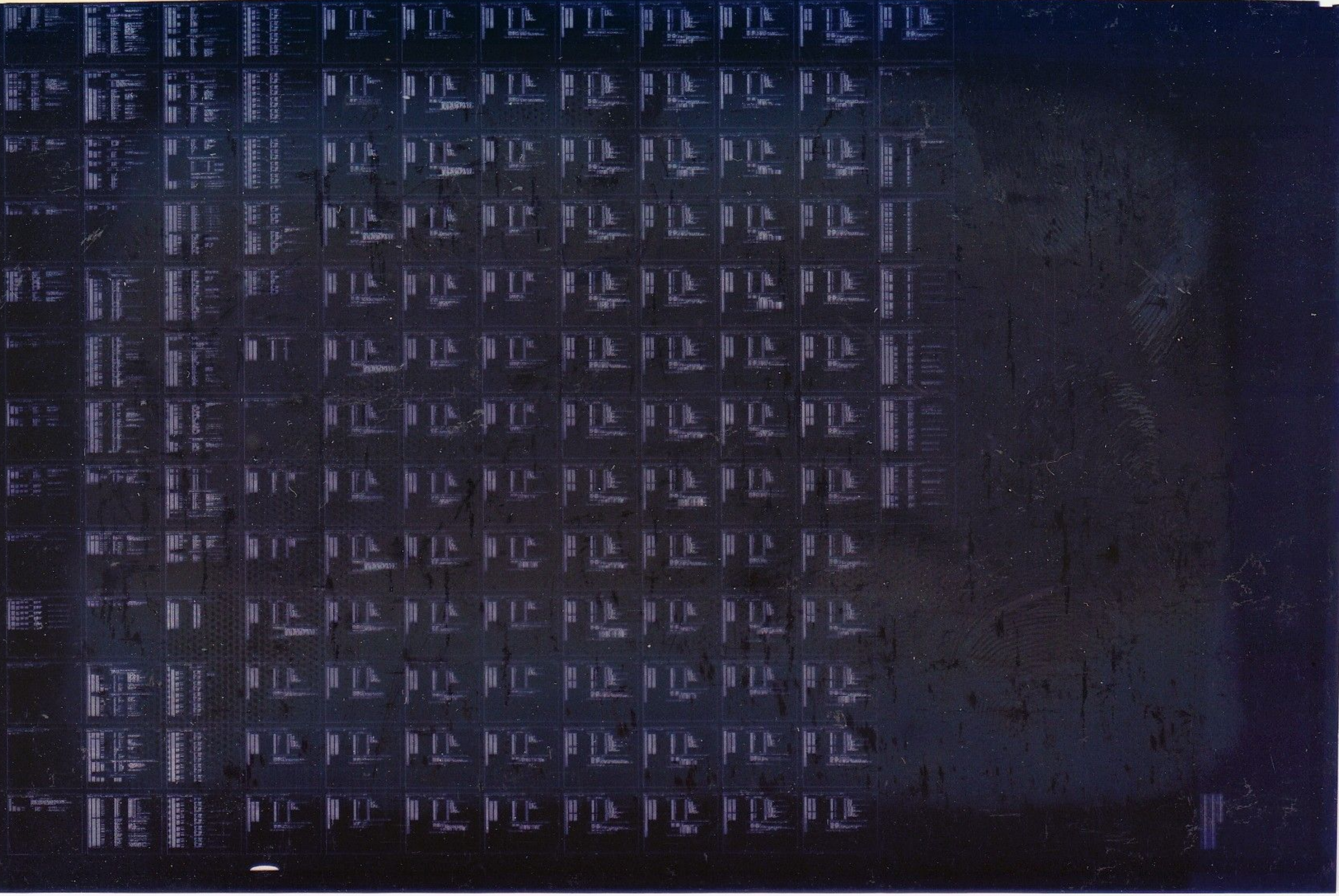
CIS INSTR EXER
CZKEEA0

AH-F227A-MC

COPYRIGHT 1980
FICHE 2 OF 2

JAN 1980

digital
MADE IN USA



5131
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177

.REM 8

IDENTIFICATION

PRODUCT CODE: AC-F226A-MC
PRODUCT NAME: CZKEEA0 PDP-11 CIS INST EXERCISER
MAINTAINER: BASE SYSTEMS DIAGNOSTIC ENGINEERING
AUTHOR: BARRY S POLAND

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

| | | | |
|---------|-------|---------|---------|
| DIGITAL | PDP | UNIBUS | MASSBUS |
| DEC | DECUS | DECTAPE | |

5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208

TABLE OF CONTENTS

| | |
|-----|------------------------------------|
| 1.0 | GENERAL INFORMATION |
| 1.1 | PROGRAM ABSTRACT |
| 1.2 | SYSTEM REQUIREMENTS |
| 1.3 | RELATED DOCUMENTS AND STANDARDS |
| 1.4 | DIAGNOSTIC HIERARCHY PREREQUISITES |
| 1.5 | ASSUMPTIONS |
| 2.0 | OPERATING INSTRUCTIONS |
| 2.1 | LOADING AND STARTING PROCEDURES |
| 2.2 | SPECIAL ENVIRONMENTS |
| 2.3 | PROGRAM OPTIONS |
| 2.4 | EXECUTION TIMES |
| 3.0 | ERROR INFORMATION |
| 4.0 | PERFORMANCE AND PROGRESS REPORTS |
| 4.1 | PERFORMANCE REPORTS |
| 4.2 | PROGRESS REPORTS |
| 5.0 | REVISION HISTORY |
| 6.0 | PROGRAM TABLE OF CONTENTS |

5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE CIS INSTRUCTION EXERCISER TESTS ALL CIS INSTRUCTIONS IN BOTH REGISTER AND IN-LINE MODES. EACH INSTRUCTION IS TESTED USING ALL COMBINATIONS OF OPERAND DATA TYPES, IN EACH OF THE THREE POSSIBLE PROCESSOR MODES (USER, SUPERVISOR, KERNEL), WITH MEMORY MANAGEMENT ENABLED/DISABLED, WITH D-SPACE ENABLED/DISABLED, IN AN INTERRUPT ENVIRONMENT, FOR MANY CASES OF STRING LENGTH, STRING ADDRESS AND STRING DATA.

THIS PROGRAM IS NOT DIRECTED AT ANY ONE CIS HARDWARE IMPLEMENTATION BUT RATHER IS INTENDED TO PROVIDE THOROUGH INSTRUCTION EXERCISING FOR ALL PDP-11 CIS PROCESSORS.

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 28K WORDS OF MEMORY AND IS COMPATIBLE WITH XXDP, ACT AND APT. IT CAN BE RUN STANDALONE UNDER XXDP, AND CAN BE CHAINED UNDER XXDP, ACT AND APT (REFERENCE XXDP USERS MANUAL FOR DETAILS OF CHAINING PROCEDURE).

THIS PROGRAM SETS UP FOR AND EXECUTES ONE CIS INSTRUCTION AT A TIME AND THEN COMPARES RESULTS WITH EXPECTED RESULTS. ERROR MESSAGES IDENTIFY ALL OPERANDS AND STRING DATA ASSOCIATED WITH THE FAILING INSTRUCTION TEST CASE. THE PROGRAM IS STRUCTURED AS A SINGLE COMPLEX LOOP WHICH GETS REEXECUTED ONCE FOR EACH INSTRUCTION TEST CASE. INSTRUCTION OPERANDS FOR EACH TEST CASE ARE EITHER EXTRACTED FROM INPUT TABLES OR GENERATED USING A RANDOM NUMBER GENERATOR. EXPECTED RESULTS ARE COMPUTED IN THE LOOP BY EMULATING CIS INSTRUCTIONS USING THE BASIC PDP-11 INSTRUCTIONS.

1.1.2 DIAGNOSTIC INFORMATION

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11 PROCESSOR (WITH CIS CAPABILITY) WITH 28K OR MORE OF MEMORY
CONSOLE DEVICE (LA30, LA36, VT50, ETC.)
PROGRAM LOAD DEVICE (PAPER TAPE, APT, ACT, DISK, MAGTAPE, ETC)
OPTIONAL HARDWARE:
1 OR 2 KW11-P PROGRAMMABLE REAL TIME CLOCKS
1 MHZ OSCILLATOR

5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317

1.2.2 SOFTWARE REQUIREMENTS

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP USERS MANUAL
DEC STANDARD 168 (PDP11 EXTENDED INSTRUCTIONS)

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

ALL BASE PROCESSOR DIAGNOSTICS AND THE CIS DIAGNOSTIC
SHOULD BE RUN ERROR FREE BEFORE ATTEMPTING TO EXECUTE THIS
CIS INSTRUCTION EXERCISER.

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK
PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC.,
DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER.
IT MAY ALSO BE LOADED FROM ANY XXDP LOAD MEDIA. THE PROGRAM IS
BOTH APT AND ACT COMPATIBLE AND CAN BE DOWN LINE LOADED
INTO THE SYSTEM UNDER TEST FROM THE APT OR ACT HOST PROCESSOR.

2.1.2 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE WITHOUT READING THE REMAINDER OF THIS
DOCUMENT, AS FOLLOWS:

- A) LOAD THE DIAGNOSTIC
- B) START AT ADDRESS 200
- G) GET END OF PASS MESSAGES OR ERROR MESSAGES

2.1.3 STARTING PROCEDURE

5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371

THE NORMAL PROGRAM STARTING ADDRESS IS 200. AN OPTIONAL STARTING ADDRESS (204) PROVIDES FOR USER SELECTION OF INSTRUCTION(S) TO TEST AND USER CONTROL OVER TEST ENVIRONMENT. AN OPTIONAL STARTING ADDRESS (210) PROVIDES A QUICK VERIFY (ONLY) MODE TAILORED TO THE PROCESSOR TYPE UNDER TEST TO RUN IN LESS THAN 5 MINUTES PER PASS AND PROVIDE A FAIR LEVEL OF MICROCODE COVERAGE (>80%).

STARTING ADDRESS = 200

STARTING AT ADDRESS 200 RESULTS IN EXECUTION OF THE STANDARD TEST SEQUENCE. A QV PASS IS RUN FIRST (SUBSET OF ALL TABLED TEST CASES). THIS QV PASS IS FOLLOWED BY A END OF QV PASS INDICATION. THEN ALL TABLED TEST CASES FOR ALL INSTRUCTIONS ARE EXECUTED (APPROX 30 MINUTES ON AN 11/44) FOLLOWED BY AN END OF PASS INDICATION. TESTING THEN PROCEEDS IN A RANDOM MODE UNTIL THE OPERATOR TERMINATES EXECUTION.

CIS INSTRUCTION INTERRUPTABILITY IS EXERCISED PROVIDED THE SYSTEM UNDER TEST HAS EITHER A LINE TIME CLOCK (KW11-L TYPE) OR A PROGRAMMABLE REAL TIME CLOCK (KW11-P). THE PROGRAM USES THE KW11-P @ 100KHZ IF BOTH CLOCKS EXIST.

PROCESSOR MODE (KERNEL,SUPERVISOR,USER) IS SELECTED RANDOMLY PRIOR TO EXECUTION OF EACH CIS INSTRUCTION TEST CASE. MEMORY MANAGEMENT IS ENABLED WITH THE D-SPACE ENABLE/DISABLE STATE SELECTED RANDOMLY PRIOR TO EACH TEST CASE. MODE IS SWITCHED TO THE TEST MODE AND MEMORY MANAGEMENT IS TURNED ON JUST PRIOR TO EXECUTION OF THE CIS INSTRUCTION UNDER TEST. DURING INTERRUPT SERVICE AND IMMEDIATELY FOLLOWING THE COMPLETION OF THE CIS INSTRUCTION EXECUTION THE MODE IS SWITCHED BACK TO KERNEL AND MEMORY MANAGEMENT IS SHUT OFF.

TABLED TEST CASES ARE EXHAUSTED FOR A GIVEN INSTRUCTION BEFORE PROCEEDING TO TEST THE NEXT CIS INSTRUCTION. AT THE START OF EACH NEW INSTRUCTION (NON-RANDOM MODE) A MESSAGE IS DISPLAYED AS A PROGRESS INDICATOR IDENTIFYING THE CIS INSTRUCTION UNDER TEST. A 'CONTROL T' ENTERED AT ANY TIME WILL CAUSE THE PROGRAM TO DISPLAY THE INSTRUCTION UNDER TEST AND THE CURRENT INSTRUCTION COUNT. THE FOLLOWING LIST IDENTIFIES THE ORDER IN WHICH INSTRUCTIONS ARE TESTED (NON-RANDOM MODE) AND THE APPROXIMATE NUMBER OF TESTS EXECUTED FOR EACH INSTRUCTION (AFTER THE QV PASS).

| INSTRUCTION | # OF TESTS |
|-------------|------------|
| ----- | ----- |
| L2D | 8 |
| L3D | 8 |
| MOVC | 354 |
| LOCC | 36 |
| CMPC | 362 |
| MOVRC | 354 |

| | | |
|------|-------|------|
| 5372 | MOVTC | 354 |
| 5373 | SKPC | 30 |
| 5374 | MATC | 904 |
| 5375 | SCANC | 126 |
| 5376 | SPANC | 126 |
| 5377 | CVTPN | 226 |
| 5378 | CVTNP | 568 |
| 5379 | CVTLP | 170 |
| 5380 | CVTLN | 323 |
| 5381 | CVTPL | 53 |
| 5382 | CVTNL | 99 |
| 5383 | ADDP | 1970 |
| 5384 | ADDN | 3872 |
| 5385 | SUBP | 1970 |
| 5386 | SUBN | 3746 |
| 5387 | CMPP | 502 |
| 5388 | CMPN | 1089 |
| 5389 | ASHP | 1972 |
| 5390 | ASHN | 3872 |
| 5391 | MULP | 1993 |
| 5392 | DIVP | 1973 |
| 5393 | | |
| 5394 | | |
| 5395 | | |
| 5396 | | |
| 5397 | | |
| 5398 | | |
| 5399 | | |
| 5400 | | |
| 5401 | | |
| 5402 | | |
| 5403 | | |
| 5404 | | |
| 5405 | | |
| 5406 | | |
| 5407 | | |
| 5408 | | |
| 5409 | | |
| 5410 | | |
| 5411 | | |
| 5412 | | |
| 5413 | | |
| 5414 | | |
| 5415 | | |
| 5416 | | |
| 5417 | | |
| 5418 | | |
| 5419 | | |
| 5420 | | |
| 5421 | | |
| 5422 | | |
| 5423 | | |
| 5424 | | |
| 5425 | | |

AFTER BEING STARTED AT LOCATION 200 THE PROGRAM SHOULD
RESPOND AS FOLLOWS:

CZKEEA0 PDP-11 CIS INSTRUCTION EXERCISER
QUICK VERIFY PASS TIME: LESS THAN 5 MINUTES
L2D0 INST CT: XX XXXXX

.

DIVP INST CT: XX XXXXX
END OF QUICK VERIFY PASS
INST UNDER TEST WILL BE DISPLAYED
PASS TIME: 11/XX APPROX. XX MIN
L2D0 INST CT: XX XXXXX

.

DIVP INST CT: XX XXXXX
END OF PASS (EXECUTION OF TABLED TEST CASES COMPLETE)
ENTERING RANDOM TEST MODE
NO FURTHER END OF PASS MESSAGES WILL BE ISSUED
RANDOM # GENERATOR SEED CONSTANTS WILL BE PRINTED
EVERY 1024 CIS INSTRUCTION TESTS
RANDOM # GENERATOR SEED XXXXXX XXXXXX XXXXXX

.

.

(UNTIL PROGRAM EXECUTION IS TERMINATED BY USER)

5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479

THE INSTRUCTION COUNT DISPLAYED AT THE START OF TESTING FOR EACH INSTRUCTION IS CUMULATIVE FROM THE FIRST L2D0 CIS INSTRUCTION TESTED. THE LOWER 5 DIGIT COUNT GETS INCREMENTED ONCE PER CIS INSTRUCTION TEST (I.E. ONCE PER CIS INSTRUCTION EXECUTED) AND COUNTS FROM 0 TO 65,535 (DECIMAL). THE UPPER 2 DIGIT COUNT GETS INCREMENTED ONCE PER 65,535 TESTS. THE INSTRUCTION COUNT IS ZEROED AT THE START OF RANDOM MODE TESTING. CONTROL T MUST BE USED TO DISPLAY THE INSTRUCTION COUNT IN RANDOM MODE.

IN XXDP CHAIN AND ACT CHAIN MODE TESTING TERMINATES AFTER THE END OF PASS INDICATION AND CONTROL IS RETURNED TO THE RESPECTIVE MONITOR. RANDOM MODE IS NOT ENTERED IN THESE CHAIN MODES. RANDOM TEST MODE IS ENTERED AUTOMATICALLY IN ALL OTHER ENVIRONMENTS (STANDALONE,XXDP MANUAL,ACT DUMP,APT).

THE RANDOM # GENERATOR SEED CONSTANTS ARE DISPLAYED TO PERMIT THE USER TO STOP AND LATER RESUME RANDOM TESTING FROM THE TERMINATION POINT. THIS IS EXPLAINED BELOW UNDER THE HEADING 'STARTING ADDRESS = 214'.

STARTING ADDRESS = 204

STARTING AT ADDRESS 204 REQUIRES THE OPERATOR TO RESPOND TO QUESTIONS TO SELECT INSTRUCTION(S) FOR TEST, TEST MODE, AND TEST ENVIRONMENT.

AFTER BEING STARTED AT LOCATION 204 THE PROGRAM SHOULD RESPOND AS FOLLOWS:

CZKEEAO PDP-11 CIS INSTRUCTION EXERCISER
TEST INTERRUPTABILITY OF CIS INSTRUCTIONS (Y OR N)?
RANDOM EXERCISE MODE (Y OR N)?
ENTER INSTRUCTION TO TEST <ALL>

IF THE USER ANSWERS YES (Y) TO THE INTERRUPTABILITY QUESTION THE PROGRAM WILL PROMPT FOR WHAT INTERRUPT SOURCE TO USE (LTC- LINE TIME CLOCK, KW11-P @ 100KHZ, KW11-P @10KHZ, KW11-P WITH EXTERNAL 1 MHZ OSCILLATOR). IF THE LTC IS SELECTED, THE PROGRAM CONTROLS INTERRUPT TIMING TO ASSURE THAT MOST CIS INSTRUCTIONS ARE INTERRUPTED ONCE. IF THE KW11-P WITH A 1 MHZ EXTERNAL OSCILLATOR IS SELECTED, EACH CIS INSTRUCTION WILL BE INTERRUPTED AND FORCED TO SUSPEND EXECUTION AT ALL POSSIBLE SERVICE EXIT POINTS. USE OF THE P-CLK WILL GREATLY INCREASE RUN TIME.

IF EITHER THE KW11-P @100KHZ OR THE KW11-P WITH EXTERNAL OSCILLATOR IS SELECTED, THE PROGRAM WILL ASK WHETHER OR NOT TO ALLOW AN INTERRUPT DURING THE CIS INST (DIVP - STATE DISTURBING INSTRUCTION) NORMALLY EXECUTED WITHIN THE KW11-P INTERRUPT SERVICE ROUTINE.

IF THE USER ANSWERS YES (Y) TO THE RANDOM EXERCISE MODE QUESTION, MEMORY MANAGEMENT TEST STATE, PROCESSOR TEST MODE,

5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533

TEST OPERANDS AND STRING DATA FOR EACH CIS INSTRUCTION TEST WILL BE DERIVED USING A RANDOM NUMBER GENERATOR. A NO (N) ANSWER WILL CAUSE EXECUTION OF CIS INSTRUCTION TESTS WITH ALL TEST OPERANDS AND STRING DATA PROVIDED FROM PROGRAM INPUT AND PARAMETER TABLES. FOLLOWING A (N) RESPONSE, THE PROGRAM WILL PROMPT FOR PROCESSOR TEST MODE (KERNEL,SUPERVISOR,USER) AND MEMORY MANAGEMENT TEST STATE (OFF,ON WITH D SPACE ENABLED, ON WITH D SPACE DISABLED).

THE LAST QUESTION ENABLES THE USER TO SELECT ONE OR ALL CIS INSTRUCTIONS FOR TEST. TO SELECT A SINGLE INSTRUCTION FOR TEST ENTER THE NMEUMONIC FOR THE DESIRED INSTRUCTION FROM THE INSTRUCTION LIST ABOVE. THE SAME QUESTION WILL BE REASKED IF THE INSTRUCTION IS INCORRECTLY ENTERED. TO SELECT ALL CIS INSTRUCTIONS FOR TEST (THE DEFAULT CASE) SIMPLY RESPOND WITH A CARRIAGE RETURN.

IF THE RANDOM MODE QUESTION IS ANSWERED YES (Y) AND THE INSTRUCTION(S) FOR TEST IS ANSWERED 'ALL', THE ACTUAL INSTRUCTION UNDER TEST ON ANY GIVEN TEST WILL BE SELECTED AT RANDOM.

STARTING ADDRESS = 210

STARTING AT ADDRESS 210 PROVIDES REPETITIVE QUICK VERIFY PASSES. NOTE THAT THE QV PASS IS DESIGNED TO GIVE A FAIR LEVEL OF MICROCODE COVERAGE (>80%) IN LESS THAN 5 MINUTES PER PASS.

THIS QV MODE RESULTS IN EXECUTION OF A SUBSET OF THE TABLED TEST CASES. THE SUBSET HAS BEEN VERIFIED TO PROVIDE AT LEAST THE DESIRED 80% LEVEL OF COVERAGE. NOTE, THE SUBSET OF TABLED TEST CASES THAT GETS RUN IN QV MODE VARIES WITH PROCESSOR TYPE. ALSO NOTE THAT SOME CIS INSTRUCTIONS MAY NOT BE EXECUTED AT ALL IN QV MODE BECAUSE IT HAS BEEN DETERMINED THAT DUE TO COMMON ROUTINES WITHIN THE MICROCODE IMPLEMENTATION IT IS POSSIBLE TO GET THE 80% COVERAGE WITHOUT EXERCISING ALL INSTRUCTIONS.

THE INSTRUCTION COUNTS LISTED UNDER THE NORMAL RUN MODE (STARTING ADDRESS 200) ABOVE DO NOT APPLY IN QV MODE.

CIS INSTRUCTION INTERRUPTABILITY IS EXERCISED PROVIDED THAT THE SYSTEM UNDER TEST HAS EITHER A LINE TIME CLOCK OR A PROGRAMMABLE REAL TIME CLOCK (KW11-P).

PROCESSOR TEST MODE(KERNEL,SUPERVISOR, USER) AND MEMORY MANAGEMENT TEST STATE ARE SELECTED RANDOMLY AS IN THE 'STARTING ADDRESS = 200' SECTION ABOVE.

AFTER BEING STARTED AT LOCATION 210, THE PROGRAM SHOULD RESPOND AS FOLLOWS:

CZKEAAO PDP-11 CIS INSTRUCTION EXERCISER
QUICK VERIFY PASS TIME: APPROX. 3 MINUTES

5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587

L2D0 INST CT: XX XXXXX

DIVP INST CT: XX XXXXX
END OF QUICK VERIFY PASS

RANDOM MODE EXERCISING IS NOT INVOKED DURING A QUICK VERIFY PASS.

STARTING ADDRESS = 214

STARTING AT ADDRESS 214 ALLOWS THE USER TO MODIFY RANDOM NUMBER GENERATOR SEED CONSTANTS. THIS IS DESIRABLE IN 2 SITUATIONS.

IF THE USER DESIRES TO RUN IN RANDOM TEST MODE FOR VERY LONG PERIODS OF TIME (DAYS, WEEKS, ETC), THE RANDOM NUMBER GEN. SEED CONSTANTS PRINTED EVERY 1024 TESTS PROVIDE FOR STOPPING AND LATER CONTINUING WITHOUT REPEATING PRIOR TESTS RUN. (REMEMBER THAT THE RANDOM # GENERATOR USED IS PSEUDO RANDOM - I.E. THE SAME SEQUENCE OF RANDOM TESTS IS EXECUTED EVERY TIME THE PROGRAM IS RESTARTED FROM THE BEGINNING).

THE SEED CONSTANTS ARE ALSO DISPLAYED WITH THE STANDARD ERROR REPORT. THIS PERMITS THE USER TO START WITH THE FAILING TEST AT SOME FUTURE TIME.

AFTER STARTING AT 214 THE PROGRAM QUERIES FOR RANDOM NUMBER SEED CONSTANTS:

ENTER THE 3 RANDOM NUMBER GEN. SEED CONSTANTS:

AFTER THE THIRD SEED IS ENTERED THE PROGRAM WILL CONTINUE AS IF STARTED AT 204. ANSWER YES TO THE RANDOM EXERCISE MODE QUESTION AND <CR> TO THE ENTER INST TO TEST QUESTION. THE FIRST TEST EXECUTED WILL BE GENERATED USING THE NEW SEEDS.

2.2 SPECIAL ENVIRONMENTS

APT - THE CIS INSTRUCTION EXERCISER IS FULLY APT COMPATIBLE, HOWEVER ITS OPERATION UNDER APT IS SOMEWHAT DIFFERENT THAN THAT OF OTHER DIAGNOSTICS. THE FIRST 2 PASSES UNDER APT ARE IDENTICAL TO THE TESTS RUN IN STANDALONE - 1 QV PASS AND 1 FULL TABLED TEST CASE PASS. SUBSEQUENT PASSES ARE NOT IDENTICAL TO THE 2ND PASS BUT RATHER BLOCKS OF 20,000 (OCTAL) RANDOM MODE TEST CASES. THAT IS, EACH PASS (BEYOND THE FIRST) IS A UNIQUE SET OF RANDOM CIS

5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641

INSTRUCTION TEST CASES.

THE INFORMATION RECORDED (AND SUBSEQUENTLY DISPLAYED) BY APT ON ERROR INCLUDES TEST NUMBER AND FATAL ERROR NUMBER. THE FATAL ERROR NUMBER SHOULD BE INTERPRETED AS FOLLOWS:

BITS 5-0 FAILING CIS INST (REF OCTAL CODING TABLE)
BIT 6 INST TYPE (0=REG; 1=IN LINE)
BIT 9 ACTIVE REGISTER SET
BITS 13-12 PROCESSOR MODE (11=USER,01=SUP,00=KERNEL)
BIT 14 INTERRUPT (1=INST WAS INTERRUPTED)

2.3 PROGRAM OPTIONS

THE FOLLOWING CONTROL CHARACTERS ARE RECOGNIZED BY THE EXERCISER DURING TEST EXECUTION:

- CNTL T - DISPLAY INST UNDER TEST AND TEST #
- CNTL C - (RECOGNIZED ONLY IF PROGRAM WAS STARTED AT 204). RESTART EXERCISER.
- CNTL D - DISPLAY ALL TEST CASE OPERANDS AND RESULTS SUBSEQUENT TO EACH CIS INST TEST. CONTINUE (WITHOUT QUERY) TO NEXT TEST.
- CNTL E - DISPLAY ALL TEST CASE OPERANDS AND RESULTS SUBSEQUENT TO EACH CIS INST TEST. QUERY FOR CONTINUE.
- CNTL N - CANCEL PRIOR CNTL D OR CNTL E REQUEST
- CNTL O - CONTROL OVER PROGRESS INDICATION PRINTOUT (I.E. INST AND INST CNT; RANDOM NUMBER GENERATOR SEED). ON - OFF TOGGLE.

2.4 EXECUTION TIMES

THE FIRST PASS RUN TIME (TABLED TEST CASES ONLY) IS APPROXIMATELY:

11/44 - 30 MINUTES

AFTER THE FIRST PASS THE PROGRAM ENTERS RANDOM TEST MODE AND EXECUTES RANDOMLY GENERATED TEST CASES INDEFINITELY.

IN QV MODE THE PASS TIME IS LESS THAN 5 MINUTES. REFER TO DOCUMENTATION ABOVE FOR DEFINITION OF QV MODE.

3.0 ERROR INFORMATION

IF THE COMPUTER HALTS WITHOUT ERROR DISPLAY THE FOLLOWING LOCATIONS SHOULD BE EXAMINED TO DETERMINE INFORMATION ABOUT THE FAILING TEST.

5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695

TINST --- CIS INSTRUCTION UNDER TEST

TR0 - TR6 --- CIS INSTRUCTION OPERANDS (LENGTHS,ADDRESSES,ETC)

INFORMATION DISPLAYED UPON DETECTION OF AN ERROR DESCRIBES THE COMPLETE ENVIRONMENT OF THE FAILURE. ALL INSTRUCTION ERRORS ARE DISPLAYED WITH ONE FORMAT. THE FORMAT CONTAINS SLIGHT VARIATIONS TO ACCOUNT FOR DIFFERENCES BETWEEN CHARACTER AND DECIMAL STRING INSTRUCTIONS.

CHARACTER STRING INSTRUCTION ERROR DISPLAY

```

-----
ERROR #000001 -----
MOV  INST CNT: 00 00004  INTR CNT:0010  REG SET:0  MODE:K  D EN:N
      SL      SA      DL      DA      F
INPUT  R0-R6,CC/ 003760 111241 000054 111046 000344 155555 053444 1111  NZVC
EXP OUT R0-R6,CC/ 003704 000000 000000 000000 000344 155555 053444 0000
ACT OUT R0-R6,CC/          000001
EXP BUFFER 111246/ 057
ACT BUFFER 100246/ 344
C=CONT.;R=REPEAT TEST;S=RESTART;D=DISPLAY MEMORY;H=REPEAT AND HALT AT CIS?

```

DECIMAL STRING INSTRUCTION ERROR DISPLAY

```

-----
ERROR #000002-----
ASHP INST CNT: 00 00250  INTR CNT: 0000  REG SET:1  MODE:S  D EN:Y
      SL      SA      DL      DA      R,S
INPUT  R0-R6,CC/ 070000 110200 070000 110206 000005 155555 053444 0100  NZVC
EXP OUT R0-R6,CC/ 000000 000000 070000 000206 000000 155555 053444 1011
ACT OUT R0-R6,CC/
SRC          0+  SIGN BYTE=0F
EXP RESULT  0+  SIGN BYTE=0F
ACT RESULT  0+  SIGN BYTE=0F
C=CONT.;R=REPEAT TEST;S=RESTART;D=DISPLAY MEMORY;H=REPEAT AND HALT AT CIS?

```

THE WORD 'ERROR' FOLLOWED BY A COUNT OF THE NUMBER OF ERRORS WHICH HAVE OCCURRED UP TO AND INCLUDING THIS TEST FAILURE AND A LONG STRING OF DASHES IS USED TO SEPARATE ONE TEST FAILURE DISPLAY FROM THE NEXT.

THE SECOND LINE OF THE ERROR REPORT IDENTIFIES THE CIS INSTRUCTION THAT FAILED, A COUNT OF THE NUMBER OF CIS INSTRUCTIONS WHICH HAVE EXECUTED, A COUNT OF THE NUMBER OF TIMES THE FAILING INSTRUCTION TEST CASE WAS SUSPENDED DUE TO INTERRUPT,THE ACTIVE REGISTER SET (0 OR 1),

5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749

THE TEST MODE (KERNEL,SUPERVISOR,USER), AND WHETHER D SPACE (MEMORY MGMT) WAS ENABLED DURING CIS INSTRUCTION EXECUTION.

THE THIRD LINE GIVES HEADER LABELS TO IDENTIFY REGISTER OR IN-LINE OPERANDS FOR THE PARTICULAR CIS INSTRUCTION UNDER TEST. 'SL' IDENTIFIES THE SOURCE LENGTH OPERAND; 'DA' IDENTIFIES THE DESTINATION ADDRESS, ETC. CONDITION CODE LABELS ARE INCLUDED AT THE END OF THIS LINE.

THE FOURTH LINE DISPLAYS ACTUAL OPERAND VALUES AND CONDITION CODES USED AS CIS INSTRUCTION INPUTS.

THE FIFTH LINE DISPLAYS EXPECTED REGISTER AND CONDITION CODE CONTENTS AT THE COMPLETION OF CIS INSTRUCTION EXECUTION. THESE EXPECTED VALUES ARE DERIVED BY EMULATION AS NOTED ABOVE.

THE SIXTH LINE, ACTUAL CIS INSTRUCTION OUTPUT, IS DISPLAYED ONLY IF ANY OF THE ACTUAL REGISTER OR CONDITION CODE OUTPUTS DO NOT AGREE WITH THE EXPECTED VALUES. ONLY THOSE SPECIFIC RESULTS WHICH ARE NOT IN AGREEMENT ARE DISPLAYED.

THE REMAINING PORTION OF THE ERROR PRINTOUT VARIES WITH THE CIS INSTRUCTION UNDER TEST. SOURCE OPERANDS,EXPECTED AND ACTUAL OPERATION RESULTS ARE DISPLAYED WITH EACH NIBBLE REPRESENTED BY A HEXADECIMAL DIGIT. SIGNS ARE DISPLAYED IN SYMBOLIC FORMAT (+,-) AND THE SIGN BYTE IS GIVEN AS TWO HEXADECIMAL NIBBLES.

'BUFFER XXXXXX' IS DISPLAYED ONLY WHEN THE BUFFER ASSOCIATED WITH ACTUAL CIS INSTRUCTION EXECUTION DIFFERS FROM THAT ASSOCIATED WITH EMULATION. ONLY THE FIRST BYTE (STARTING FROM THE LOW ADDRESS END OF THE BUFFERS UNDER CONSIDERATION) IN DISAGREEMENT IS PRINTED.

THE LAST LINE DISPLAYED AS PART OF EACH ERROR REPORT PERMITS THE USER SEVERAL OPTIONS RELATING TO HOW TO PROCEED FOLLOWING AN ERROR. CONTINUE (C) PROCEEDS TO THE NEXT INSTRUCTION TEST CASE AS IF THE ERROR HAD NEVER OCCURRED. DISPLAY MEMORY (D) ALLOWS THE USER TO DISPLAY ANY BYTES(S) IN MEMORY. RESTART (S) RETURNS CONTROL TO THE BEGINNING OF THE PROGRAM. REPEAT TEST (R) REPEATS THE FAILING TEST CASE. THE SAME ERROR MESSAGE WILL BE DISPLAYED AGAIN PROVIDED THE TEST FAILS DURING THE REPEAT TEST. THE REPEAT AND HALT OPTION REPEATS THE FAILING TEST BUT HALTS JUST PRIOR TO EXECUTING THE CIS INSTRUCTION UNDER TEST. THIS MODE ALLOWS THE USER TO SINGLE STEP THROUGH THE FAILING CIS MICROCODE USING CONSOLE COMMANDS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776

NONE

4.2 PROGRESS REPORTS

THE CIS INSTRUCTION AND THE TEST COUNT IS DISPLAYED AT THE START OF TESTING FOR EACH CIS INSTRUCTION TYPE (EXCEPT IN RANDOM TEST MODE). NOTE, AS STATED IN SECTION 2.1.3 ABOVE THAT MANY TEST CASES ARE EXECUTED FOR EACH CIS INSTRUCTION.

IN RANDOM TEST MODE THE RANDOM NUMBER GENERATOR SEED CONSTANTS ARE DISPLAYED EVERY 1024 TESTS. IF THE DIAGNOSTIC IS TO BE RUN FOR A PROLONGED PERIOD IN THIS MODE, THESE CONSTANTS PROVIDE A MECHANISM FOR STOPPING AND LATER CONTINUING THE DIAG AT ANY FUTURE TIME. REFER TO SECTION 2.1.3 UNDER THE HEADING "STARTING ADDRESS = 214" FOR INSTRUCTIONS ON HOW TO USE THE RANDOM # GENERATOR SEED CONSTANTS.

5.0 REVISION HISTORY

6.0 PROGRAM TABLE OF CONTENTS

```

6211      .ENABL ABS,AMA
6326      .TITLE PDP-11 CIS INST EXERCISER
(1)      .*COPYRIGHT (C) 1979
(1)      .*DIGITAL EQUIPMENT CORP.
(1)      .*MAYNARD, MASS. 01754
(1)      .*
(1)      .*PROGRAM BY BARRY POLAND
(1)      .*
(1)      .*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
(1)      .*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
(1)      .*
(1)      000001 $TN=1
(1)      160000 $SWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

6331      .SBTTL BASIC DEFINITIONS
(1)
(1)      .*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
(1)      001100 STACK= 1100      ;;FIRST ADDRESS OF THE STACK
(1)      001100 KERSTK= STACK      ;;KERNEL STACK
(1)      000700 SUPSTK= STACK-200      ;;SUPERVISOR STACK
(1)      000600 USESTK= STACK-300      ;;USER STACK
(1)      .EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
(1)      .EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
(1)      177776 PS= 177776      ;;PROCESSOR STATUS WORD
(1)      .EQUIV PS,PSW
(1)      177774 STKLMT= 177774      ;;STACK LIMIT REGISTER
(1)      177772 PIRQ= 177772      ;;PROGRAM INTERRUPT REQUEST REGISTER
(1)      177570 DSWR= 177570      ;;SWITCH REGISTER
(1)      177570 DDISP=177570
(1)
(1)      .*MISCELLANEOUS DEFINITIONS
(1)      000011 HT= 11      ;;CODE FOR HORIZONTAL TAB
(1)      000012 LF= 12      ;;CODE LINE FEED
(1)      000015 CR= 15      ;;CODE CARRIAGE RETURN
(1)      000200 CRLF= 200      ;;CODE FOR CARRIAGE RETURN-LINE FEED
(1)
(1)      .*GENERAL PURPOSE REGISTER DEFINITIONS
(1)      000000 R0= %0      ;;GENERAL REGISTER
(1)      000001 R1= %1      ;;GENERAL REGISTER
(1)      000002 R2= %2      ;;GENERAL REGISTER
(1)      000003 R3= %3      ;;GENERAL REGISTER
(1)      000004 R4= %4      ;;GENERAL REGISTER
(1)      000005 R5= %5      ;;GENERAL REGISTER
(1)      000006 R6= %6      ;;GENERAL REGISTER
(1)      000007 R7= %7      ;;GENERAL REGISTER
(1)      .EQUIV R0,R10      ;;GENERAL REGISTER
(1)      .EQUIV R1,R11      ;;GENERAL REGISTER
(1)      .EQUIV R2,R12      ;;GENERAL REGISTER
(1)      .EQUIV R3,R13      ;;GENERAL REGISTER
(1)      .EQUIV R4,R14      ;;GENERAL REGISTER
(1)      .EQUIV R5,R15      ;;GENERAL REGISTER
(1)      .EQUIV R6,SP      ;;STACK POINTER
(1)      .EQUIV SP,KSP      ;;KERNEL STACK POINTER
(1)      .EQUIV SP,SSP      ;;SUPERVISOR STACK POINTER

```



```
(1) .EQUIV SP,USP          ;;USER STACK POINTER
(1) .EQUIV R7,PC          ;;PROGRAM COUNTER
(1)
(1) ;*PRIORITY LEVEL DEFINITIONS
(1) 000000 PR0= 0          ;;PRIORITY LEVEL 0
(1) 000040 PR1= 40        ;;PRIORITY LEVEL 1
(1) 000100 PR2= 100       ;;PRIORITY LEVEL 2
(1) 000140 PR3= 140       ;;PRIORITY LEVEL 3
(1) 000200 PR4= 200       ;;PRIORITY LEVEL 4
(1) 000240 PR5= 240       ;;PRIORITY LEVEL 5
(1) 000300 PR6= 300       ;;PRIORITY LEVEL 6
(1) 000340 PR7= 340       ;;PRIORITY LEVEL 7
(1)
(1) ;*'SWITCH REGISTER' SWITCH DEFINITIONS
(1) 100000 SW15= 100000
(1) 040000 SW14= 40000
(1) 020000 SW13= 20000
(1) 010000 SW12= 10000
(1) 004000 SW11= 4000
(1) 002000 SW10= 2000
(1) 001000 SW09= 1000
(1) 000400 SW08= 400
(1) 000200 SW07= 200
(1) 000100 SW06= 100
(1) 000040 SW05= 40
(1) 000020 SW04= 20
(1) 000010 SW03= 10
(1) 000004 SW02= 4
(1) 000002 SW01= 2
(1) 000001 SW00= 1
(1) .EQUIV SW09,SW9
(1) .EQUIV SW08,SW8
(1) .EQUIV SW07,SW7
(1) .EQUIV SW06,SW6
(1) .EQUIV SW05,SW5
(1) .EQUIV SW04,SW4
(1) .EQUIV SW03,SW3
(1) .EQUIV SW02,SW2
(1) .EQUIV SW01,SW1
(1) .EQUIV SW00,SW0
(1)
(1) ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1) 100000 BIT15= 100000
(1) 040000 BIT14= 40000
(1) 020000 BIT13= 20000
(1) 010000 BIT12= 10000
(1) 004000 BIT11= 4000
(1) 002000 BIT10= 2000
(1) 001000 BIT09= 1000
(1) 000400 BIT08= 400
(1) 000200 BIT07= 200
(1) 000100 BIT06= 100
(1) 000040 BIT05= 40
(1) 000020 BIT04= 20
```

```
(1) 000010 BIT03= 10
(1) 000004 BIT02= 4
(1) 000002 BIT01= 2
(1) 000001 BIT00= 1
(1) .EQUIV BIT09,BIT9
(1) .EQUIV BIT08,BIT8
(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0
```

```
(1) ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1) 000004 ERRVEC= 4 ;;TIME OUT AND OTHER ERRORS
(1) 000010 RESVEC= 10 ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1) 000014 TBITVEC=14 ;;'T' BIT
(1) 000014 TRTVEC= 14 ;;TRACE TRAP
(1) 000014 BPTVEC= 14 ;;BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 ;;POWER FAIL
(1) 000030 EMTVEC= 30 ;;EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC=34 ;;'TRAP' TRAP
(1) 000060 TKVEC= 60 ;;TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 ;;TTY PRINTER VECTOR
(1) 000114 CACHVEC=114 ;;CACHE ERROR INTERRUPT VECTOR
(1) 000240 PIRQVEC=240 ;;PROGRAM INTERRUPT REQUEST VECTOR
(1) 000250 MMVEC= 250 ;;MEMORY MANAGEMENT VECTOR
```

(1) .SBTTL CACHE REGISTER DEFINITIONS

```
(1) 177740 LOADRS = 177740 ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
(1) 177742 HIADRS = 177742 ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
(1) 177744 MEMERR = 177744 ;;CACHE ERROR REGISTER
(1) 177746 CONTRL = 177746 ;;MEMORY CONTROL REGISTER
(1) 177750 MAINT = 177750 ;;MEMORY MAINTENANCE REGISTER
(1) 177752 HITMIS = 177752 ;;HIT MISS REGISTER '1' IMPLIES HIT IN CACHE
```

(1) .SBTTL CPU REGISTER DEFINITIONS

```
(1) 177760 SIZELO = 177760 ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
(1) 177762 SIZEHI = 177762 ;;TO GET TO THE LAST 32 WORDS OF MEMORY
(1) 177764 SYSTID = 177764 ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
(1) 177766 CPUERR = 177766 ;;CURRENTLY ALL ZERO
(1) ;;SYSTEM ID REGISTER
(1) ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
(1) ;;THE TRAP TO ERRVEC (000004)
```

```
(1)
(1)
(1)          .SBTTL MEMORY MANAGEMENT DEFINITIONS
(1)
(1)          ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
(1)
(1)          177572 MMR0= 177572
(1)          177574 MMR1= 177574
(1)          177576 MMR2= 177576
(1)          172516 MMR3= 172516
(1)          .EQUIV MMR0,SR0
(1)          .EQUIV MMR1,SR1
(1)          .EQUIV MMR2,SR2
(1)          .EQUIV MMR3,SR3
(1)
(1)          ;*USER 'I' PAGE DESCRIPTOR REGISTERS
(1)
(1)          177600 UIPDR0= 177600
(1)          177602 UIPDR1= 177602
(1)          177604 UIPDR2= 177604
(1)          177606 UIPDR3= 177606
(1)          177610 UIPDR4= 177610
(1)          177612 UIPDR5= 177612
(1)          177614 UIPDR6= 177614
(1)          177616 UIPDR7= 177616
(1)
(1)          ;*USER 'D' PAGE DESCRIPTOR REGISTORS
(1)
(1)          177620 UDPDR0= 177620
(1)          177622 UDPDR1= 177622
(1)          177624 UDPDR2= 177624
(1)          177626 UDPDR3= 177626
(1)          177630 UDPDR4= 177630
(1)          177632 UDPDR5= 177632
(1)          177634 UDPDR6= 177634
(1)          177636 UDPDR7= 177636
(1)
(1)          ;*USER 'I' PAGE ADDRESS REGISTERS
(1)
(1)          177640 UIPAR0= 177640
(1)          177642 UIPAR1= 177642
(1)          177644 UIPAR2= 177644
(1)          177646 UIPAR3= 177646
(1)          177650 UIPAR4= 177650
(1)          177652 UIPAR5= 177652
(1)          177654 UIPAR6= 177654
(1)          177656 UIPAR7= 177656
(1)
(1)          ;*USER 'D' PAGE ADDRESS REGISTERS
(1)
(1)          177660 UDPAR0= 177660
(1)          177662 UDPAR1= 177662
(1)          177664 UDPAR2= 177664
```

MEMORY MANAGEMENT DEFINITIONS

| | | |
|-----|--------|--|
| (1) | 177666 | UDPAR3= 177666 |
| (1) | 177670 | UDPAR4= 177670 |
| (1) | 177672 | UDPAR5= 177672 |
| (1) | 177674 | UDPAR6= 177674 |
| (1) | 177676 | UDPAR7= 177676 |
| (1) | | |
| (1) | | ;*SUPERVISOR 'I' PAGE DESCRIPTOR REGISTERS |
| (1) | | |
| (1) | 172200 | SIPDR0= 172200 |
| (1) | 172202 | SIPDR1= 172202 |
| (1) | 172204 | SIPDR2= 172204 |
| (1) | 172206 | SIPDR3= 172206 |
| (1) | 172210 | SIPDR4= 172210 |
| (1) | 172212 | SIPDR5= 172212 |
| (1) | 172214 | SIPDR6= 172214 |
| (1) | 172216 | SIPDR7= 172216 |
| (1) | | |
| (1) | | ;*SUPERVISOR 'D' PAGE DESCRIPTOR REGISTERS |
| (1) | | |
| (1) | 172220 | SDPDR0= 172220 |
| (1) | 172222 | SDPDR1= 172222 |
| (1) | 172224 | SDPDR2= 172224 |
| (1) | 172226 | SDPDR3= 172226 |
| (1) | 172230 | SDPDR4= 172230 |
| (1) | 172232 | SDPDR5= 172232 |
| (1) | 172234 | SDPDR6= 172234 |
| (1) | 172236 | SDPDR7= 172236 |
| (1) | | |
| (1) | | ;*SUPERVISOR 'I' PAGE ADDRESS REGISTERS |
| (1) | | |
| (1) | 172240 | SIPAR0= 172240 |
| (1) | 172242 | SIPAR1= 172242 |
| (1) | 172244 | SIPAR2= 172244 |
| (1) | 172246 | SIPAR3= 172246 |
| (1) | 172250 | SIPAR4= 172250 |
| (1) | 172252 | SIPAR5= 172252 |
| (1) | 172254 | SIPAR6= 172254 |
| (1) | 172256 | SIPAR7= 172256 |
| (1) | | |
| (1) | | ;*SUPERVISOR 'D' PAGE ADDRESS REGISTERS |
| (1) | | |
| (1) | 172260 | SDPAR0= 172260 |
| (1) | 172262 | SDPAR1= 172262 |
| (1) | 172264 | SDPAR2= 172264 |
| (1) | 172266 | SDPAR3= 172266 |
| (1) | 172270 | SDPAR4= 172270 |
| (1) | 172272 | SDPAR5= 172272 |
| (1) | 172274 | SDPAR6= 172274 |
| (1) | 172276 | SDPAR7= 172276 |
| (1) | | |
| (1) | | ;*KERNEL 'I' PAGE DESCRIPTOR REGISTERS |
| (1) | | |
| (1) | 172300 | KIPDR0= 172300 |
| (1) | 172302 | KIPDR1= 172302 |

```
(1) 172304 KIPDR2= 172304
(1) 172306 KIPDR3= 172306
(1) 172310 KIPDR4= 172310
(1) 172312 KIPDR5= 172312
(1) 172314 KIPDR6= 172314
(1) 172316 KIPDR7= 172316
(1)
(1) ;*KERNEL 'D' PAGE DESCRIPTOR REGISTERS
(1)
(1) 172320 KDPDR0= 172320
(1) 172322 KDPDR1= 172322
(1) 172324 KDPDR2= 172324
(1) 172326 KDPDR3= 172326
(1) 172330 KDPDR4= 172330
(1) 172332 KDPDR5= 172332
(1) 172334 KDPDR6= 172334
(1) 172336 KDPDR7= 172336
(1)
(1) ;*KERNEL 'I' PAGE ADDRESS REGISTERS
(1)
(1) 172340 KIPAR0= 172340
(1) 172342 KIPAR1= 172342
(1) 172344 KIPAR2= 172344
(1) 172346 KIPAR3= 172346
(1) 172350 KIPAR4= 172350
(1) 172352 KIPAR5= 172352
(1) 172354 KIPAR6= 172354
(1) 172356 KIPAR7= 172356
(1)
(1) ;*KERNEL 'D' PAGE ADDRESS REGISTERS
(1)
(1) 172360 KDPAR0= 172360
(1) 172362 KDPAR1= 172362
(1) 172364 KDPAR2= 172364
(1) 172366 KDPAR3= 172366
(1) 172370 KDPAR4= 172370
(1) 172372 KDPAR5= 172372
(1) 172374 KDPAR6= 172374
(1) 172376 KDPAR7= 172376
(1)
(1)
(1) .SBTTL UNIBUS MAP REGISTER DEFINITIONS
(1)
(1) ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
(1) ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'
(1)
(1) 170200 MAPL00 = 170200
(1) 170202 MAPH00 = 170202
(1) 170204 MAPL01 = 170204
(1) 170206 MAPH01 = 170206
```

| | | |
|-----|--------|-----------------|
| (1) | 170210 | MAPL02 = 170210 |
| (1) | 170212 | MAPH02 = 170212 |
| (1) | 170214 | MAPL03 = 170214 |
| (1) | 170216 | MAPH03 = 170216 |
| (1) | 170220 | MAPL04 = 170220 |
| (1) | 170222 | MAPH04 = 170222 |
| (1) | 170224 | MAPL05 = 170224 |
| (1) | 170226 | MAPH05 = 170226 |
| (1) | 170230 | MAPL06 = 170230 |
| (1) | 170232 | MAPH06 = 170232 |
| (1) | 170234 | MAPL07 = 170234 |
| (1) | 170236 | MAPH07 = 170236 |
| (1) | 170240 | MAPL10 = 170240 |
| (1) | 170242 | MAPH10 = 170242 |
| (1) | 170244 | MAPL11 = 170244 |
| (1) | 170246 | MAPH11 = 170246 |
| (1) | 170250 | MAPL12 = 170250 |
| (1) | 170252 | MAPH12 = 170252 |
| (1) | 170254 | MAPL13 = 170254 |
| (1) | 170256 | MAPH13 = 170256 |
| (1) | 170260 | MAPL14 = 170260 |
| (1) | 170262 | MAPH14 = 170262 |
| (1) | 170264 | MAPL15 = 170264 |
| (1) | 170266 | MAPH15 = 170266 |
| (1) | 170270 | MAPL16 = 170270 |
| (1) | 170272 | MAPH16 = 170272 |
| (1) | 170274 | MAPL17 = 170274 |
| (1) | 170276 | MAPH17 = 170276 |
| (1) | 170300 | MAPL20 = 170300 |
| (1) | 170302 | MAPH20 = 170302 |
| (1) | 170304 | MAPL21 = 170304 |
| (1) | 170306 | MAPH21 = 170306 |
| (1) | 170310 | MAPL22 = 170310 |
| (1) | 170312 | MAPH22 = 170312 |
| (1) | 170314 | MAPL23 = 170314 |
| (1) | 170316 | MAPH23 = 170316 |
| (1) | 170320 | MAPL24 = 170320 |
| (1) | 170320 | MAPH24 = 170320 |
| (1) | 170324 | MAPL25 = 170324 |
| (1) | 170326 | MAPH25 = 170326 |
| (1) | 170330 | MAPL26 = 170330 |
| (1) | 170332 | MAPH26 = 170332 |
| (1) | 170334 | MAPL27 = 170334 |
| (1) | 170336 | MAPH27 = 170336 |
| (1) | 170340 | MAPL30 = 170340 |
| (1) | 170342 | MAPH30 = 170342 |
| (1) | 170344 | MAPL31 = 170344 |
| (1) | 170346 | MAPH31 = 170346 |
| (1) | 170350 | MAPL32 = 170350 |
| (1) | 170352 | MAPH32 = 170352 |
| (1) | 170354 | MAPL33 = 170354 |
| (1) | 170356 | MAPH33 = 170356 |
| (1) | 170360 | MAPL34 = 170360 |
| (1) | 170362 | MAPH34 = 170362 |

| | | |
|------|--------|---------------------|
| (1) | 170364 | MAPL35 = 170364 |
| (1) | 170366 | MAPH35 = 170366 |
| (1) | 170370 | MAPL36 = 170370 |
| (1) | 170372 | MAPH36 = 170372 |
| (1) | 170374 | MAPL37 = 170374 |
| (1) | 170376 | MAPH37 = 170376 |
| (1) | | .EQUIV MAPL00,MAPL0 |
| (1) | | .EQUIV MAPH00,MAPH0 |
| (1) | | .EQUIV MAPL01,MAPL1 |
| (1) | | .EQUIV MAPH01,MAPH1 |
| (1) | | .EQUIV MAPL02,MAPL2 |
| (1) | | .EQUIV MAPH02,MAPH2 |
| (1) | | .EQUIV MAPL03,MAPL3 |
| (1) | | .EQUIV MAPH03,MAPH3 |
| (1) | | .EQUIV MAPL04,MAPL4 |
| (1) | | .EQUIV MAPH04,MAPH4 |
| (1) | | .EQUIV MAPL05,MAPL5 |
| (1) | | .EQUIV MAPH05,MAPH5 |
| (1) | | .EQUIV MAPL06,MAPL6 |
| (1) | | .EQUIV MAPH06,MAPH6 |
| (1) | | .EQUIV MAPL07,MAPL7 |
| (1) | | .EQUIV MAPH07,MAPH7 |
| (1) | | |
| (1) | | |
| (1) | | |
| 6332 | | : |
| 6333 | 076175 | DIVPI==076175 |
| 6334 | 170000 | BS0==170000 |
| 6335 | 100000 | BS00==100000 |
| 6336 | 174000 | BS2==174000 |
| 6337 | 176000 | BS4==176000 |
| 6338 | 177000 | BS8==177000 |
| 6339 | 177400 | BS16==177400 |
| 6340 | 177600 | BS32==177600 |
| 6341 | 177700 | BS64==177700 |
| 6342 | 177740 | BS128==177740 |
| 6343 | 177760 | BS256==177760 |
| 6344 | 177770 | BS512==177770 |
| 6345 | 177777 | BSNULL==177777 |
| 6346 | 177400 | BY==177400 |
| 6347 | 006000 | MP06000==006000 |
| 6348 | 004000 | MP04000==004000 |
| 6349 | 000000 | WD==0 |
| 6350 | 125252 | EOT==125252 |
| 6351 | 000020 | MP020==000020 |
| 6352 | 000070 | MP070==000070 |
| 6353 | 000100 | MP0100==000100 |
| 6354 | 000140 | MP0140==000140 |
| 6355 | 002000 | MP02000==002000 |
| 6356 | 004040 | MP04040==004040 |
| 6357 | 152525 | DSCPTR==152525 |
| 6358 | 022000 | MP02200==22000 |
| 6359 | | |

:MASK (176000) THEN OFFSET (6000)

```
(1) .SBTTL TRAP CATCHER
(1)
(1) 000000 .=0
(1) ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1) ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1) ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1) 000174 000174 .=174
(1) 000174 000000 DISPREG: .WORD 0 ;:SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SWREG: .WORD 0 ;:SOFTWARE SWITCH REGISTER
(1)
(1) .SBTTL STARTING ADDRESS(ES)
(1) 000200 000137 036514 JMP @#START ;:JUMP TO STARTING ADDRESS OF PROGRAM
6361 000204 000137 036710 JMP DVTST
6362 000210 000137 036522 JMP QVST
6363 000214 000137 036504 JMP SEEDST
6370
6372 ;*****
(1)
(1) .SBTTL ACT11 HOOKS
(1) ;HOOKS REQUIRED BY ACT11
(1) 000220 000220 $SVPC=. ;SAVE PC
(1) 000046 000046 .=46
(1) 053076 ENDAD ;:1)SET LOC.46 TO ADDRESS OF ENDAD
(1) 000052 000052 .=52
(1) 000000 .WORD 0 ;:2)SET LOC.52 TO ZERO
(1) 000220 .=$SVPC ;: RESTORE PC
6373 001100 .=1100
6375 ;*****
(1)
(1) .SBTTL APT PARAMETER BLOCK
(1) ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2) ;*****
(1) 001100 .SX=. ;:SAVE CURRENT LOCATION
(1) 000024 .=24 ;:SET POWER FAIL TO POINT TO START OF PROGRAM
(1) 000024 000200 200 ;:FOR APT START UP
(1) 000044 .=44 ;:POINT TO APT INDIRECT ADDRESS PNTR.
(1) 000044 001100 $APTHDR ;:POINT TO APT HEADER BLOCK
(1) 001100 .=$X ;:RESET LOCATION COUNTER
(2) ;*****
(1) ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1) ;INTERFACE SPEC.
(1)
(1) 001100 $APTHD:
(1) 001100 000000 $HIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 001102 001114 $MBADR: .WORD $MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 001104 000170 $TSTM: .WORD 120. ;:RUN TIM OF LONGEST TEST
(1) 001106 000264 $PASTM: .WORD 180. ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 001110 000000 $UNITM: .WORD 0 ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 001112 000016 .WORD $ETEND-$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)
6380 ;*****
(1)
(1) .SBTTL APT MAILBOX-ETABLE
(1)
(1)
```



```
(1)
(1)
(1) 001114 .EVEN
(1) 001114 000000 $MAIL: ::APT MAILBOX
(1) 001116 000000 $MSGTY: .WORD AMSGTY ::MESSAGE TYPE CODE
(1) 001120 000000 $FATAL: .WORD AFATAL ::FATAL ERROR NUMBER
(1) 001122 000000 $TESTN: .WORD ATESTN ::TEST NUMBER
(1) 001124 000000 $PASS: .WORD APASS ::PASS COUNT
(1) 001126 000000 $DEVCT: .WORD ADEVCT ::DEVICE COUNT
(1) 001130 000000 $UNIT: .WORD AUNIT ::I/O UNIT NUMBER
(1) 001132 000000 $MSGAD: .WORD AMSGAD ::MESSAGE ADDRESS
(1) 001134 000000 $MSGLG: .WORD AMSGLG ::MESSAGE LENGTH
(1) 001134 000 $ETABLE: ::APT ENVIRONMENT TABLE
(1) 001135 000 $ENV: .BYTE AENV ::ENVIRONMENT BYTE
(1) 001136 000000 $ENVM: .BYTE AENVM ::ENVIRONMENT MODE BITS
(1) 001140 000000 $SWREG: .WORD ASWREG ::APT SWITCH REGISTER
(1) 001142 000000 $USWR: .WORD AUSWR ::USER SWITCHES
(1) $CPUOP: .WORD ACPUOP ::CPU TYPE,OPTIONS
(1) * BITS 15-11=CPU TYPE
(1) * 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(1) * 11/70=06,PDQ=07,Q=10
(1) * BIT 10=REAL TIME CLOCK
(1) * BIT 9=FLOATING POINT PROCESSOR
(1) * BIT 8=MEMORY MANAGEMENT
(1) 001144 000 $MAMS1: .BYTE AMAMS1 ::HIGH ADDRESS,M.S. BYTE
(1) 001145 000 $MTYP1: .BYTE AMTYP1 ::MEM. TYPE,BLK#1
(1) * MEM.TYPE BYTE -- (HIGH BYTE)
(1) * 900 NSEC CORE=001
(1) * 300 NSEC BIPOLAR=002
(1) * 500 NSEC MOS=003
(1) 001146 000000 $MADR1: .WORD AMADR1 ::HIGH ADDRESS,BLK#1
(1) * MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(1) 001150 $ETEND:
(1) .MEXIT
6381 *
(1) .SBTTL APT COMMUNICATIONS ROUTINE
(1) 001150 112737 000001 001414 $ATY1: MOV #1,$FFLG ;TO REPORT FATAL ERROR
(1) 001156 112737 000001 001412 $ATY3: MOV #1,$MFLG ;TO TYPE A MESSAGE
(1) 001164 000403 BR $ATYC
(1) 001166 112737 000001 001414 $ATY4: MOV #1,$FFLG ;TO ONLY REPORT FATAL ERROR
(2) 001174 $ATYC:
(3) 001174 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
(3) 001176 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
(1) 001200 105737 001412 TSTB $MFLG ;SHOULD TYPE A MESSAGE?
(1) 001204 001450 BEQ 5$ ;IF NOT: BR
(1) 001206 122737 000001 001134 CMPB #APTENV,$ENV ;OPERATING UNDER APT?
(1) 001214 001031 BNE 3$ ;IF NOT: BR
(1) 001216 132737 000100 001135 BITB #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGES?
(1) 001224 001425 BEQ 3$ ;IF NOT: BR
(1) 001226 017600 000004 MOV @4(SP),R0 ;GET MESSAGE ADDR.
(1) 001232 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDR.
(1) 001240 005737 001114 1$: TST $MSGTYPE ;SEE IF DONE W/ LAST XMISSION?
(1) 001244 001375 BNE 1$ ;IF NOT: WAIT
(1) 001246 010037 001130 MOV R0,$MSGAD ;PUT ADDR IN MAILBOX
```

```

(1) 001252 105720          2$:  TSTB  (R0)+      ;FIND END OF MESSAGE
(1) 001254 001376          BNE   2$
(1) 001256 163700 001130  SUB   $MSGAD,R0    ;SUB START OF MESSAGE
(1) 001262 006200          ASR   R0           ;GET MESSAGE LNTH IN WORDS
(1) 001264 010037 001132  MOV   R0,$MSGLGT  ;PUT LENGTH IN MAILBOX
(1) 001270 012737 000004 001114  MOV   #4,$MSGTYPE ;TELL APT TO TAKE MSG.
(1) 001276 000413          BR    5$
(1) 001300 017637 000004 001324 3$:  MOV   @4(SP),4$    ;PUT MSG ADDR IN JSR LINKAGE
(1) 001306 062766 000002 000004  ADD   #2,4(SP)     ;BUMP RETURN ADDRESS
(3) 001314 013746 177776  MOV   177776,-(SP) ;:PUSH 177776 ON STACK
(1) 001320 004737 110156  JSR   PC,$TYPE    ;CALL TYPE MACRO
(1) 001324 000000          4$:  .WORD 0
(1) 001326          5$:
(1) 001326 105737 001414 10$:  TSTB  $FFLG      ;SHOULD REPORT FATAL ERROR?
(1) 001332 001416          BEQ   12$        ;IF NOT: BR
(1) 001334 005737 001134  TST   $ENV       ;RUNNING UNDER APT?
(1) 001340 001413          BEQ   12$        ;IF NOT: BR
(1) 001342 005737 001114 11$:  TST   $MSGTYPE   ;FINISHED LAST MESSAGE?
(1) 001346 001375          BNE   11$       ;IF NOT: WAIT
(1) 001350 017637 000004 001116  MOV   @4(SP),$FATAL ;GET ERROR #
(1) 001356 062766 000002 000004  ADD   #2,4(SP)     ;BUMP RETURN ADDR.
(1) 001364 005237 001114  INC   $MSGTYPE   ;TELL APT TO TAKE ERROR
(1) 001370 105037 001414 12$:  CLRB  $FFLG      ;CLEAR FATAL FLAG
(1) 001374 105037 001413  CLRB  $LFLG      ;CLEAR LOG FLAG
(1) 001400 105037 001412  CLRB  $MFLG      ;CLEAR MESSAGE FLAG
(3) 001404 012601          MOV   (SP)+,R1    ;:POP STACK INTO R1
(3) 001406 012600          MOV   (SP)+,R0    ;:POP STACK INTO R0
(1) 001410 000207          RTS   PC         ;RETURN
(1) 001412          000  $MFLG: .BYTE 0      ;MESSG. FLAG
(1) 001413          000  $LFLG: .BYTE 0      ;LOG FLAG
(1) 001414          000  $FFLG: .BYTE 0      ;FATAL FLAG
(1)          001416  .EVEN
(1)          000200  APTSIZE=200
(1)          000001  APTENV=001
(1)          000100  APTSPOOL=100
(1)          000040  APTCSUP=040
6382 ;*****
  
```

6384 .SBTTL GLOBAL DATA SECTION

6385
6386
6387
6388
6389
6390
6391
6392
6393
6394

001416 000000
001420 000000
001422 000000
001424 000000

.SBTTL TEST COUNTS
TOTTC: .WORD 0
TOTTC: .WORD 0
INVTC: .WORD 0
REDTC: .WORD 0

;TEST COUNT MULTIPLIER
;TOTAL TEST COUNT
;INVALID TEST COUNT
;REDUNDANT TEST COUNT

| | | | | |
|------|--------|--------|--------|---------------------------------------|
| 6396 | | | .SBTTL | INPUT TABLE ENTRY TYPE DISPATCH TABLE |
| 6397 | | | :INPUT | TABLE ENTRY TYPE DISPATCH TABLE. |
| 6398 | | | : | |
| 6399 | 001426 | | :TYPE: | |
| 6400 | 001426 | 040622 | .WORD | TYPE0 |
| 6401 | 001430 | 040772 | .WORD | TYPE1 |
| 6402 | 001432 | 000000 | .WORD | 0 |
| 6403 | 001434 | 000000 | .WORD | 0 |
| 6404 | | | | |

| | | | | |
|------|--------|--------|---|----------------------------------|
| 6406 | | | .SBTTL | POINTERS TO CIS INST FLOW TABLES |
| 6407 | | | :POINTERS TO CIS INSTRUCTION FLOW TABLES. | |
| 6408 | | | : | |
| 6409 | 001436 | | INO: | |
| 6410 | 001436 | 000000 | .WORD | 0 |
| 6411 | 001440 | 006330 | .WORD | XMOV C |
| 6412 | 001442 | 006330 | .WORD | XMOV RC |
| 6413 | 001444 | 006410 | .WORD | XMOV TC |
| 6414 | 001446 | 006772 | .WORD | XLOCC |
| 6415 | 001450 | 006772 | .WORD | XSKPC |
| 6416 | 001452 | 007130 | .WORD | XSCANC |
| 6417 | 001454 | 007130 | .WORD | XSPANC |
| 6418 | 001456 | 006472 | .WORD | XCMPC |
| 6419 | 001460 | 006554 | .WORD | XMATCHC |
| 6420 | 001462 | 007040 | .WORD | XADDN |
| 6421 | 001464 | 007040 | .WORD | XSUBN |
| 6422 | 001466 | 006636 | .WORD | XCMPN |
| 6423 | 001470 | 007340 | .WORD | XCVTNL |
| 6424 | 001472 | 007212 | .WORD | XCVTPN |
| 6425 | 001474 | 007212 | .WORD | XCVTNP |
| 6426 | 001476 | 006712 | .WORD | XASHN |
| 6427 | 001500 | 007270 | .WORD | XCVTLN |
| 6428 | 001502 | 007040 | .WORD | XADDP |
| 6429 | 001504 | 007040 | .WORD | XSUBP |
| 6430 | 001506 | 006636 | .WORD | XCMPP |
| 6431 | 001510 | 007340 | .WORD | XCVTPL |
| 6432 | 001512 | 007040 | .WORD | XMULP |
| 6433 | 001514 | 007040 | .WORD | XDIVP |
| 6434 | 001516 | 006712 | .WORD | XASHP |
| 6435 | 001520 | 007270 | .WORD | XCVTLP |
| 6436 | 001522 | 007402 | .WORD | XL2D |
| 6437 | 001524 | 007402 | .WORD | XL3D |
| 6438 | | | | |

6440
6441
6442
6443 001526
6444 001526 053236
6445 001530 041612
6446 001532 042022
6447 001534 042122
6448 001536 042172
6449 001540 042300
6450 001542 042406
6451 001544 042516
6452 001546 045502
6453 001550 045536
6454 001552 045734
6455 001554 047700
6456 001556 051000
6457 001560 053706
6458 001562 053236
6459 001564 053236
6460
6461

.SBTTL FLOW COMMAND DISPATCH TABLE
;FLOW COMMAND DISPATCH TABLE

;FLODIS:

.WORD FC00
.WORD FC01
.WORD FC02
.WORD FC03
.WORD FC04
.WORD FC05
.WORD FC06
.WORD FC07
.WORD FC10
.WORD FC11
.WORD FC12
.WORD FC13
.WORD FC14
.WORD FC15
.WORD FC00
.WORD FC00

;FLOW COMMANDS 16,& 17 ARE UNUSED.

| | | | | |
|------|--------|--------|---------------------------|--------------------------|
| 6463 | | | .SBTTL | PARAMETER TABLE POINTERS |
| 6464 | | | :PARAMETER TABLE POINTERS | |
| 6465 | | | : | |
| 6466 | 001566 | | PTP: | |
| 6467 | 001566 | 000000 | | .WORD 0 |
| 6468 | 001570 | 000000 | PTP01: | .WORD 0 |
| 6469 | 001572 | 000000 | PTP02: | .WORD 0 |
| 6470 | 001574 | 000000 | PTP03: | .WORD 0 |
| 6471 | 001576 | 000000 | PTP04: | .WORD 0 |
| 6472 | 001600 | 000000 | PTP05: | .WORD 0 |
| 6473 | 001602 | 000000 | PTP06: | .WORD 0 |
| 6474 | 001604 | 000000 | PTP07: | .WORD 0 |
| 6475 | 001606 | 000000 | PTP10: | .WORD 0 |
| 6476 | 001610 | 000000 | PTP11: | .WORD 0 |
| 6477 | 001612 | 000000 | PTP12: | .WORD 0 |
| 6478 | 001614 | 000000 | PTP13: | .WORD 0 |
| 6479 | 001616 | 000000 | PTP14: | .WORD 0 |
| 6480 | 001620 | 000000 | PTP15: | .WORD 0 |
| 6481 | 001622 | 000000 | PTP16: | .WORD 0 |
| 6482 | 001624 | 000000 | PTP17: | .WORD 0 |
| 6483 | 001626 | 000000 | PTP20: | .WORD 0 |
| 6484 | 001630 | 000000 | PTP21: | .WORD 0 |
| 6485 | 001632 | 000000 | PTP22: | .WORD 0 |
| 6486 | 001634 | 000000 | PTP23: | .WORD 0 |
| 6487 | 001636 | 000000 | PTP24: | .WORD 0 |
| 6488 | | | | |

```
6490          .SBTTL          MISCELLANEOUS CONSTANTS
6491          ;MISCELLANEOUS CONSTANTS
6492          ;
6494 001640 130000          TBADR: .WORD 130000          ;TEST BUFFER STARTING ADDRESS
6495 001642 002000          TBLN: .WORD 2000          ;TEST BUFFER LENGTH
6496 001644 010000          RTBLN: .WORD 10000          ;RANDOM MODE TEST BUFFER LENGTH
6497 001646 140000          EBADR: .WORD 140000          ;EMULATION BUFFER STARTING ADDRESS
6498 001650 002000          EBLN: .WORD 2000          ;EMULATION BUFFER LENGTH
6499 001652 020000          TPERP: .WORD 20000          ;# OF TESTS PER APT PASS IN RANDOM MODE
6500 001654 002000          PSEED: .WORD 2000          ;# OF TESTS TO EXECUTE IN RANDOM MODE
6501
6502
6503
6517 001656 000024          IPNU: .WORD 24          ;# OF INPUT PARAMETERS PER INPUT
6518                                     ; TABLE ENTRY.
6519 001660 000001          INCSQ1: .WORD 1          ;INCREMENT SEQUENCE WORD 1
6520 001662 000002          INCSQ2: .WORD 2          ;INCREMENT SEQUENCE WORD 2
6521 001664 177776          TPSW: .WORD 177776          ;PSW
6522 001666 177570          DISPR: .WORD 177570          ;CONSOLE SWITCH AND DISPLAY REGISTER
6523 001670 177560          TKS: 177560
6524 001672 000240          KNOP: NOP
6525 001674 000000          KHALT: HALT
6526 001676 000405          KBR5: .WORD 405
6527 001700 000403          KBR3: .WORD 403
6528 001702 000402          KBR2: .WORD 402
6529 001704 123321          NOTREG: .WORD 123321          ;PATTERN LOADED INTO REG SET NO SELECTED FOR TEST
6530 001706 000002          EL74: .WORD 2          ;11/74 PROCESSOR TYPE
6531 001710 000001          EL44: .WORD 1          ;11/44 PROCESSOR TYPE
6532                                     ; BETWEEN PRINTING OF RNG SEED.
6533                                     ;NOTE: ONLY 1 BIT IS ALLOWED TO BE SET IN PSEED.
6534 001712 177562          TKB: 177562
6535 001714 177564          TPS: 177564
6536 001716 177566          TPB: 177566
6537          000015          CR= 15
6538          000012          LF= 12
6539          000057          SL= 57
6540          000010          BS= 10
6541          000007          MFPT= 7
6542 001720 177777          PAT0: .WORD 177777
6543 001722 111111          PAT1: .WORD 111111
6544 001724 122222          PAT2: .WORD 122222
6545 001726 133333          PAT3: .WORD 133333
6546 001730 144444          PAT4: .WORD 144444
6547 001732 155555          PAT5: .WORD 155555
6548 001734 120606          IXLTB1: .WORD XLTLB1
6549 001736 177570          SWR: .WORD DSWR          ;ADDRESS OF SWITCH REGISTER
6550 001740 177570          DISPLAY: .WORD DDISP          ;ADDRESS OF DISPLAY REGISTER
```


| | | | .SBTTL | MISCELLANEOUS VARIABLES |
|------|--------|--------|--------------------------|-------------------------|
| 6552 | | | | |
| 6553 | | | .SBTTL | MISCELLANEOUS VARIABLES |
| 6554 | | | ;MISCELLANEOUS VARIABLES | |
| 6555 | | | . | |
| 6556 | 001742 | 000000 | FLOPTR: | .WORD 0 |
| 6557 | 001744 | 000000 | ESEED: | .WORD 0 |
| 6558 | 001746 | 000000 | DENS: | .WORD 0 |
| 6559 | 001750 | 000000 | N2OOM: | .WORD 0 |
| 6560 | 001752 | 000000 | TSTPSW: | .WORD 0 |
| 6561 | 001754 | 000000 | TBEND: | .WORD 0 |
| 6562 | 001756 | 000000 | SBR: | .WORD 0 |
| 6563 | 001760 | 000000 | RANDOM: | .WORD 0 |
| 6564 | 001762 | 000000 | CTACT: | .WORD 0 |
| 6565 | 001764 | 000000 | PMASK: | .WORD 0 |
| 6566 | 001766 | 177417 | ZMSK: | .WORD 177417 |
| 6567 | 001770 | 000000 | RANDTA: | .WORD 0 |
| 6568 | 001772 | 000000 | LIMSTG: | .WORD 0 |
| 6569 | 001774 | 000000 | RNIB: | .WORD 0 |
| 6570 | 001776 | 000000 | STRNC: | .WORD 0 |
| 6571 | 002000 | 000000 | STRP1: | .WORD 0 |
| 6572 | 002002 | 000000 | STRP2: | .WORD 0 |
| 6573 | 002004 | 000000 | SXRNC: | .WORD 0 |
| 6574 | 002006 | 000000 | SXRP1: | .WORD 0 |
| 6575 | 002010 | 000000 | SXRP2: | .WORD 0 |
| 6576 | 002012 | 000000 | SYRNC: | .WORD 0 |
| 6577 | 002014 | 000000 | SYRP1: | .WORD 0 |
| 6578 | 002016 | 000000 | SYRP2: | .WORD 0 |
| 6579 | 002020 | 000000 | SWRNC: | .WORD 0 |
| 6580 | 002022 | 000000 | SWRP1: | .WORD 0 |
| 6581 | 002024 | 000000 | SWRP2: | .WORD 0 |
| 6582 | 002026 | 000000 | SVRNC: | .WORD 0 |
| 6583 | 002030 | 000000 | SVRP1: | .WORD 0 |
| 6584 | 002032 | 000000 | SVRP2: | .WORD 0 |
| 6585 | 002034 | 000000 | ZCCR: | .WORD 0 |
| 6586 | 002036 | 000000 | SPCV: | .WORD 0 |
| 6587 | 002040 | 000000 | TPRECS: | .WORD 0 |
| 6588 | 002042 | 000000 | NOERDS: | .WORD 0 |
| 6589 | 002044 | 000000 | PROGD: | .WORD 0 |
| 6590 | 002046 | 000000 | SURLN: | .WORD 0 |
| 6591 | 002050 | 000000 | SURADR: | .WORD 0 |
| 6592 | 002052 | 000000 | REGSET: | .WORD 0 |
| 6593 | 002054 | 000000 | ERRCT: | .WORD 0 |
| 6594 | 002056 | 000000 | ERRS: | .WORD 0 |
| 6595 | 002060 | 000000 | TWOSETS: | .WORD 0 |
| 6596 | 002062 | 000000 | FSAVR5: | .WORD 0 |
| 6597 | 002064 | 000000 | FSAVR4: | .WORD 0 |
| 6598 | 002066 | 000000 | FSAVR3: | .WORD 0 |
| 6599 | 002070 | 000000 | FSAVR2: | .WORD 0 |
| 6600 | 002072 | 000000 | FSAVR1: | .WORD 0 |
| 6601 | 002074 | 000000 | FSRUN: | .WORD 0 |
| 6602 | 002076 | 000000 | MSEED: | .WORD 0 |
| 6603 | 002100 | 000000 | NBLKS: | .WORD 0 |
| 6604 | 002102 | 000000 | TRA: | .WORD 0 |
| 6605 | 002104 | 000000 | TRL: | .WORD 0 |

| | | | | | |
|------|--------|--------|---------|-------|---|
| 6606 | 002106 | 000000 | STGDS1: | .WORD | 0 |
| 6607 | 002110 | 000000 | STGDS2: | .WORD | 0 |
| 6608 | 002112 | 000000 | STGLN: | .WORD | 0 |
| 6609 | 002114 | 000000 | STGAD: | .WORD | 0 |
| 6610 | 002116 | 000000 | SAVSL: | .WORD | 0 |
| 6611 | 002120 | 000000 | SAVSA: | .WORD | 0 |
| 6612 | 002122 | 000000 | SAVSGL: | .WORD | 0 |
| 6613 | 002124 | 000000 | SIGN: | .WORD | 0 |
| 6614 | 002126 | 000000 | VIP: | .WORD | 0 |
| 6615 | 002130 | 000000 | PTW1: | .WORD | 0 |
| 6616 | 002132 | 000000 | FATAL: | .WORD | 0 |
| 6617 | 002134 | 000000 | INPTP: | .WORD | 0 |
| 6618 | 002136 | 000000 | PTPTR: | .WORD | 0 |
| 6619 | 002140 | 000000 | SPHAND: | .WORD | 0 |
| 6620 | 002142 | 000000 | EMPTR: | .WORD | 0 |
| 6621 | 002144 | 000000 | ERRCC: | .WORD | 0 |
| 6622 | 002146 | 000000 | ERRREG: | .WORD | 0 |
| 6623 | 002150 | 000000 | ERRBUF: | .WORD | 0 |
| 6624 | 002152 | 000000 | ERRSTK: | .WORD | 0 |
| 6625 | 002154 | 000000 | PT34: | .WORD | 0 |
| 6626 | 002156 | 000000 | MMFLG: | .WORD | 0 |
| 6627 | 002160 | 000000 | MODE: | .WORD | 0 |
| 6628 | | | | | |
| 6629 | 002162 | 000000 | DEN: | .WORD | 0 |
| 6630 | 002164 | 000000 | NMODES: | .WORD | 0 |
| 6631 | | | | | |
| 6632 | 002166 | 000000 | TRPLOC: | .WORD | 0 |
| 6633 | 002170 | 000000 | HLTLOC: | .WORD | 0 |
| 6634 | 002172 | 000000 | IRXLT: | .WORD | 0 |
| 6635 | 002174 | 000000 | AEADR: | .WORD | 0 |
| 6636 | 002176 | 000000 | AEDTA: | .WORD | 0 |
| 6637 | 002200 | 000000 | EMADR: | .WORD | 0 |
| 6638 | 002202 | 000000 | EMDTA: | .WORD | 0 |
| 6639 | 002204 | 000000 | QRYFLG: | .WORD | 0 |
| 6640 | 002206 | 000000 | QVMODE: | .WORD | 0 |
| 6641 | 002210 | 000000 | MMSTAT: | .WORD | 0 |
| 6642 | 002212 | 000000 | PTQV: | .WORD | 0 |
| 6643 | 002214 | 000000 | ICOMPC: | .WORD | 0 |
| 6644 | 002216 | 000000 | RLL: | .WORD | 0 |
| 6645 | 002220 | 000000 | RUL: | .WORD | 0 |
| 6646 | 002222 | 000000 | BAD: | .WORD | 0 |
| 6647 | 002224 | 000000 | TSP: | .WORD | 0 |
| 6648 | 002226 | 000000 | NXFLD: | .WORD | 0 |
| 6649 | 002230 | 000000 | TTR0: | .WORD | 0 |
| 6650 | 002232 | 000000 | TTR1: | .WORD | 0 |
| 6651 | 002234 | 000000 | TTR2: | .WORD | 0 |
| 6652 | 002236 | 000000 | TTR3: | .WORD | 0 |
| 6653 | 002240 | 000000 | TTR4: | .WORD | 0 |
| 6654 | 002242 | 000000 | TTR5: | .WORD | 0 |
| 6655 | 002244 | 000000 | TTR6: | .WORD | 0 |
| 6656 | 002246 | 000000 | TEROR: | .WORD | 0 |
| 6657 | 002250 | 000000 | TER1R: | .WORD | 0 |
| 6658 | 002252 | 000000 | TER2R: | .WORD | 0 |
| 6659 | 002254 | 000000 | TER3R: | .WORD | 0 |

:11/34 TYPE PROCESSOR FLAG
:WHEN NON ZERO, TESTING WITH MEMORY MGMT ON
:PROCESSOR MODE USED FOR CIS INST TEST
: (0=KERNEL,1=SUPERVISOR,3=USER)
:D-SPACE ENABLED(1)/DISABLED(0)
:# OF PROCESSOR MODES ON MACHINE UNDER TEST
: (I.E. KERNEL,SUP,USER)

| | | | | | | |
|------|--------|--------|---------|-------|-------|---------------------|
| 6660 | 002256 | 000000 | TER4R: | .WORD | 0 | |
| 6661 | 002260 | 000000 | TER5R: | .WORD | 0 | |
| 6662 | 002262 | 000000 | TER6R: | .WORD | 0 | |
| 6663 | 002264 | 000000 | TERR: | .WORD | 0 | |
| 6664 | 002266 | 000000 | RPTFLG: | .WORD | 0 | |
| 6665 | 002270 | 000000 | FILLS2: | .WORD | 0 | |
| 6666 | 002272 | 000000 | OCTIC: | .WORD | 0 | |
| 6667 | 002274 | 000000 | TW1: | .WORD | 0 | |
| 6668 | 002276 | 000000 | TW2: | .WORD | 0 | |
| 6669 | 002300 | 000000 | PRTSGN: | .WORD | 0 | |
| 6670 | 002302 | 000000 | STGDIG: | .WORD | 0 | |
| 6671 | 002304 | 000000 | STGTYP: | .WORD | 0 | |
| 6672 | 002306 | 043632 | TYPTAB: | .WORD | TYPSZ | :SIGNED ZONED |
| 6673 | 002310 | 043664 | | .WORD | TYPUZ | :UNSIGNED ZONED |
| 6674 | 002312 | 043666 | | .WORD | TYPTO | :TRAILING OVERPUNCH |
| 6675 | 002314 | 043774 | | .WORD | TYPLO | :LEADING OVERPUNCH |
| 6676 | 002316 | 044004 | | .WORD | TYPTS | :TRAILING SEPARATE |
| 6677 | 002320 | 044050 | | .WORD | TYPLS | :LEADING SEPARATE |
| 6678 | 002322 | 002326 | | .WORD | TYPSP | :RESERVED |
| 6679 | 002324 | 002326 | | .WORD | TYPSP | :RESERVED |
| 6680 | 002326 | 000000 | TYPSP: | HALT | | |
| 6681 | 002330 | 175 | NEGTAB: | .BYTE | 175 | :-0 |
| 6682 | 002331 | 112 | | .BYTE | 112 | :-1 |
| 6683 | 002332 | 113 | | .BYTE | 113 | :-2 |
| 6684 | 002333 | 114 | | .BYTE | 114 | :-3 |
| 6685 | 002334 | 115 | | .BYTE | 115 | :-4 |
| 6686 | 002335 | 116 | | .BYTE | 116 | :-5 |
| 6687 | 002336 | 117 | | .BYTE | 117 | :-6 |
| 6688 | 002337 | 120 | | .BYTE | 120 | :-7 |
| 6689 | 002340 | 121 | | .BYTE | 121 | :-8 |
| 6690 | 002341 | 122 | | .BYTE | 122 | :-9 |
| 6691 | 002342 | 135 | NEGTB1: | .BYTE | 135 | :-0 |
| 6692 | 002343 | 112 | | .BYTE | 112 | :-1 |
| 6693 | 002344 | 113 | | .BYTE | 113 | :-2 |
| 6694 | 002345 | 114 | | .BYTE | 114 | :-3 |
| 6695 | 002346 | 115 | | .BYTE | 115 | :-4 |
| 6696 | 002347 | 116 | | .BYTE | 116 | :-5 |
| 6697 | 002350 | 117 | | .BYTE | 117 | :-6 |
| 6698 | 002351 | 120 | | .BYTE | 120 | :-7 |
| 6699 | 002352 | 121 | | .BYTE | 121 | :-8 |
| 6700 | 002353 | 122 | | .BYTE | 122 | :-9 |
| 6701 | 002354 | 173 | POSTAB: | .BYTE | 173 | :+0 |
| 6702 | 002355 | 101 | | .BYTE | 101 | :+1 |
| 6703 | 002356 | 102 | | .BYTE | 102 | :+2 |
| 6704 | 002357 | 103 | | .BYTE | 103 | :+3 |
| 6705 | 002360 | 104 | | .BYTE | 104 | :+4 |
| 6706 | 002361 | 105 | | .BYTE | 105 | :+5 |
| 6707 | 002362 | 106 | | .BYTE | 106 | :+6 |
| 6708 | 002363 | 107 | | .BYTE | 107 | :+7 |
| 6709 | 002364 | 110 | | .BYTE | 110 | :+8 |
| 6710 | 002365 | 111 | | .BYTE | 111 | :+9 |
| 6711 | 002366 | 060 | POSTB1: | .BYTE | 060 | :+0 |
| 6712 | 002367 | 061 | | .BYTE | 061 | :+1 |
| 6713 | 002370 | 062 | | .BYTE | 062 | :+2 |

| | | | | | |
|------|--------|--------|---------------|--------|--|
| 6714 | 002371 | 063 | .BYTE | 063 | :+3 |
| 6715 | 002372 | 064 | .BYTE | 064 | :+4 |
| 6716 | 002373 | 065 | .BYTE | 065 | :+5 |
| 6717 | 002374 | 066 | .BYTE | 066 | :+6 |
| 6718 | 002375 | 067 | .BYTE | 067 | :+7 |
| 6719 | 002376 | 070 | .BYTE | 070 | :+8 |
| 6720 | 002377 | 071 | .BYTE | 071 | :+9 |
| 6721 | 002400 | 133 | POSTB2: .BYTE | 133 | :+0 |
| 6722 | 002401 | 061 | .BYTE | 061 | :+1 |
| 6723 | 002402 | 062 | .BYTE | 062 | :+2 |
| 6724 | 002403 | 063 | .BYTE | 063 | :+3 |
| 6725 | 002404 | 064 | .BYTE | 064 | :+4 |
| 6726 | 002405 | 065 | .BYTE | 065 | :+5 |
| 6727 | 002406 | 066 | .BYTE | 066 | :+6 |
| 6728 | 002407 | 067 | .BYTE | 067 | :+7 |
| 6729 | 002410 | 070 | .BYTE | 070 | :+8 |
| 6730 | 002411 | 071 | .BYTE | 071 | :+9 |
| 6731 | 002412 | 045004 | PTYPTA: .WORD | PTYPSZ | :SIGNED ZONED |
| 6732 | 002414 | 045500 | .WORD | EISTG | :UNSIGNED ZONED |
| 6733 | 002416 | 044646 | .WORD | PTYPTO | :TRAILING OVERPUNCH |
| 6734 | 002420 | 045014 | .WORD | PTYPLO | :LEADING OVERPUNCH |
| 6735 | 002422 | 045024 | .WORD | PTYPTS | :TRAILING SEPARATE |
| 6736 | 002424 | 045040 | .WORD | PTYPLS | :LEADING SEPARATE |
| 6737 | 002426 | 002326 | .WORD | TYPSP | :RESERVED |
| 6738 | 002430 | 002326 | .WORD | TYPSP | :RESERVED |
| 6739 | 002432 | 000000 | ONEBEY: .WORD | 0 | |
| 6740 | 002434 | 000000 | DECINS: .WORD | 0 | |
| 6741 | 002436 | 000006 | S1TYPE: .WORD | 6 | |
| 6742 | 002440 | 000006 | S2TYPE: .WORD | 6 | |
| 6743 | 002442 | 000006 | S3TYPE: .WORD | 6 | |
| 6744 | 002444 | 070000 | TYPFLD: .WORD | 070000 | |
| 6745 | 002446 | 000000 | MIXTYP: .WORD | 0 | :MANUALLY SET TO ANY NONZERO VALUE |
| 6746 | | | | | : TO CAUSE TESTING OF MIXED DATA TYPES |
| 6747 | | | | | : WITHIN INST. NOTE: THIS WILL |
| 6748 | | | | | : GREATLY INCREASE RUN TIME!!!! |
| 6749 | 002450 | 000000 | PKPTW: .WORD | 0 | |
| 6750 | 002452 | 000000 | NDESC: .WORD | 0 | |
| 6751 | 002454 | 000000 | ZPM: .WORD | 0 | |
| 6752 | 002456 | 000000 | ACINST: .WORD | 0 | |
| 6753 | 002460 | 000000 | .WORD | 0 | |
| 6754 | 002462 | 000000 | .WORD | 0 | |
| 6755 | 002464 | 000000 | ONEINS: .WORD | 0 | |
| 6756 | 002466 | 000000 | PZCODE: .WORD | 0 | |
| 6757 | 002470 | 000000 | SAVPTR: .WORD | 0 | |
| 6758 | 002472 | 000000 | SAVSRF: .WORD | 0 | |
| 6759 | 002474 | 000000 | INSRC1: .WORD | 0 | :SRC1 STRING SAVE BUFFER DESCRIPTOR |
| 6760 | 002476 | 003035 | .WORD | BUFSR1 | |
| 6761 | 002500 | 000000 | INSRC2: .WORD | 0 | :SRC2 STRING SAVE BUFFER DESCRIPTOR |
| 6762 | 002502 | 003076 | .WORD | BUFSR2 | |
| 6763 | | | | | |
| 6764 | 002504 | 120000 | RANDSC: .WORD | 120000 | |
| 6765 | 002506 | 000000 | .WORD | 0 | |

```
6767 .SBTTL PROGRAMMABLE CLOCK CONSTANTS
6768 ;PROGRAMMABLE CLOCK CONSTANTS
6769 002510 000104 PCLK1V: .WORD 104 ;P-CLK 1
6770 002512 000106 PCLK1P: .WORD 106
6771 002514 172540 PC1CSR: .WORD 172540
6772 002516 172542 PC1CSB: .WORD 172542
6773 002520 172544 PC1CTR: .WORD 172544
6774 002522 000000 PCLK2V: .WORD 000 ;P-CLK 2
6775 002524 172550 PC2CSR: .WORD 172550
6776 002526 172552 PC2CSB: .WORD 172552
6777 002530 172554 PC2CTR: .WORD 172554
6778 002532 000004 TIMEOUT: .WORD 4
6779
6780 002534 000000 PROGCT: .WORD 0
6781 002536 000000 LATCT: .WORD 0
6782 002540 000000 LATEN: .WORD 0
6783 002542 000000 INTCT: .WORD 0
6784 002544 000240 KNOP1: .WORD NOP
6785 002546 000240 KNOP2: .WORD NOP
6786 002550 000240 KNOP3: .WORD NOP
6787 002552 000000 INTRVL: .WORD 0
6788 002554 000000 STOPLA: .WORD 0 ;IF NONZERO & LATENCY EXCEEDS THIS VALUE PROGRAM HALTS
6789 002556 001000 MAXIVL: .WORD 1000 ;USER DEFINED MAXIMUM INTERVAL ALLOWED
6790
6791 002560 000000 STATPS: .WORD 0
6792 002562 000000 STATR0: .WORD 0
6793 002564 000000 STATR1: .WORD 0
6794 002566 000000 STATR2: .WORD 0
6795 002570 000000 STATR3: .WORD 0
6796 002572 000000 STATR4: .WORD 0
6797 002574 000000 STATR5: .WORD 0
6798 002576 000000 STATR6: .WORD 0
6799 002600 000100 .BLKW ^D64
6800 003000 SCSTK:
6801
6802 003000 000000 SGPR0: .WORD 0
6803 003002 000000 SGPR1: .WORD 0
6804 003004 000000 SGPR2: .WORD 0
6805 003006 000000 SGPR3: .WORD 0
6806 003010 000000 SGPR4: .WORD 0
6807 003012 000000 SGPR5: .WORD 0
6808 003014 000000 SGPR6: .WORD 0
6809
6810
```

| | | | .SBTTL | LINE TIME CLOCK CONSTANTS |
|------|--------|--------|----------------------------|---------------------------|
| 6812 | | | | |
| 6813 | | | :LINE TIME CLOCK CONSTANTS | |
| 6814 | 003016 | 000100 | LTCIV: .WORD | 100 |
| 6815 | 003020 | 000102 | LTCIP: .WORD | 102 |
| 6816 | 003022 | 177546 | LKS: .WORD | 177546 |
| 6817 | 003024 | 000000 | LCNT: .WORD | 0 |
| 6818 | 003026 | 000000 | VLCNT: .WORD | 0 |
| 6819 | 003030 | 000240 | KNOP4: .WORD | 240 |
| 6820 | 003032 | 000000 | LTCPLY: .WORD | 0 |
| 6821 | | | | |
| 6822 | | | | |

```
6824
6825          .SBTTL          SOURCE STRING STORAGE BUFFER
6826          ;SOURCE STRING STORAGE BUFFER - USED BY ERROR PRINTOUT ROUTINES
6827          ;
6828 003034      000          .BYTE 0
6829 003035 000040          BUFSR1: .BLKB ^D32          ;S1 BUFFER
6830 003075      000          .BYTE 0
6831 003076 000040          BUFSR2: .BLKB ^D32          ;S2 BUFFER
6832
6833
6834
6835 003136      000          PB0: .BYTE 0          ;DISPLAY BUFFER
6836 003137      000          PB1: .BYTE 0
6837 003140      000          PB2: .BYTE 0
6838 003141      000          PB3: .BYTE 0
6839 003142      000          PB4: .BYTE 0
6840 003143      000          PB5: .BYTE 0
6841 003144      000          PB6: .BYTE 0
6842 003145      000          PB7: .BYTE 0
6843
```

```
6845 ;DESCRIPTORS AND DESTINATION BUFFER FOR INTERRUPTABILITY SERVICE
6846 ; ROUTINE DIVPI INST.
6847 ;
6848 003146 000012 DIVDS: .WORD 12 ;SOURCE 1 & 2 DESC
6849 003150 003244 ;.WORD SSTG2
6850 003152 000020 DIVDD: .WORD 20 ;DESTINATION DESC
6851 003154 003156 ;.WORD DESTBUF
6852 003156 000031 DESTBUF: .BLKW 31
6853
6854
```


| | | | .SBTTL | PRE-SPECIFIED STRINGS | |
|------|--------|--------|------------------------|-----------------------|----------------------------------|
| | | | :PRE-SPECIFIED STRINGS | | |
| 6856 | | | | | |
| 6857 | | | | | |
| 6858 | | | | | |
| 6859 | 003240 | 001 | SSTG1: | .BYTE 001 | |
| 6860 | 003241 | 001 | | .BYTE 001 | |
| 6861 | 003242 | 007 | | .BYTE 007 | |
| 6862 | 003243 | 000 | | .BYTE 000 | |
| 6863 | | | | | |
| 6864 | 003244 | 012002 | SSTG2: | .WORD 012002 | :PACKED 2,147,483,648 +MAX+1 |
| 6865 | 003246 | 101564 | | .WORD 101564 | |
| 6866 | 003250 | 106144 | | .WORD 106144 | |
| 6867 | | | | | |
| 6868 | 003252 | 024404 | SSTG2A: | .WORD 024404 | :PACKED 4,294,967,294 +MAX * 2 |
| 6869 | 003254 | 063511 | | .WORD 063511 | |
| 6870 | 003256 | 046051 | | .WORD 046051 | |
| 6871 | | | | | |
| 6872 | 003260 | 112102 | SSTG2B: | .WORD 112102 | :PACKED 42,949,672,940 +MAX * 20 |
| 6873 | 003262 | 071226 | | .WORD 071226 | |
| 6874 | 003264 | 006224 | | .WORD 006224 | |
| 6875 | | | | | |
| 6876 | 003266 | 000402 | SSTG3: | .WORD 000402 | :ZONED 2,147,483,648 +MAX+1 |
| 6877 | 003270 | 003404 | | .WORD 003404 | |
| 6878 | 003272 | 034064 | | .WORD 034064 | |
| 6879 | 003274 | 003003 | | .WORD 003003 | |
| 6880 | 003276 | 034164 | | .WORD 034164 | |
| 6881 | | | | | |
| 6882 | 003300 | 012002 | SSTG4: | .WORD 012002 | :PACKED 2,147,483,647 +MAX |
| 6883 | 003302 | 101564 | | .WORD 101564 | |
| 6884 | 003304 | 076144 | | .WORD 076144 | |
| 6885 | | | | | |
| 6886 | 003306 | 000402 | SSTG5: | .WORD 000402 | :ZONED 2,147,483,647 +MAX |
| 6887 | 003310 | 003404 | | .WORD 003404 | |
| 6888 | 003312 | 034064 | | .WORD 034064 | |
| 6889 | 003314 | 003003 | | .WORD 003003 | |
| 6890 | 003316 | 033564 | | .WORD 033564 | |
| 6891 | | | | | |
| 6892 | 003320 | 012002 | SSTG6: | .WORD 012002 | :PACKED -2,147,483,648 -MAX |
| 6893 | 003322 | 101564 | | .WORD 101564 | |
| 6894 | 003324 | 106544 | | .WORD 106544 | |
| 6895 | | | | | |
| 6896 | 003326 | 000402 | SSTG7: | .WORD 000402 | :ZONED -2,147,483,648 -MAX |
| 6897 | 003330 | 003404 | | .WORD 003404 | |
| 6898 | 003332 | 034064 | | .WORD 034064 | |
| 6899 | 003334 | 003003 | | .WORD 003003 | |
| 6900 | 003336 | 074164 | | .WORD 074164 | |
| 6901 | | | | | |
| 6902 | 003340 | 012002 | SSTG10: | .WORD 012002 | :PACKED -2,147,483,649 -MAX-1 |
| 6903 | 003342 | 101564 | | .WORD 101564 | |
| 6904 | 003344 | 116544 | | .WORD 116544 | |
| 6905 | | | | | |
| 6906 | 003346 | 000402 | SSTG11: | .WORD 000402 | :ZONED -2,147,483,649 -MAX-1 |
| 6907 | 003350 | 003404 | | .WORD 003404 | |
| 6908 | 003352 | 034064 | | .WORD 034064 | |
| 6909 | 003354 | 003003 | | .WORD 003003 | |

| | | | | | |
|------|--------|--------|---------|--------------|-------------------------------------|
| 6910 | 003356 | 074564 | .WORD | 074564 | |
| 6911 | | | | | |
| 6912 | 003360 | 032022 | SSTG12: | .WORD 032022 | :PACKED STRING |
| 6913 | 003362 | 074126 | | .WORD 074126 | : 1234567891234567891234000891233 |
| 6914 | 003364 | 021621 | | .WORD 021621 | |
| 6915 | 003366 | 063505 | | .WORD 063505 | |
| 6916 | 003370 | 011211 | | .WORD 011211 | |
| 6917 | 003372 | 000064 | | .WORD 000064 | |
| 6918 | 003374 | 110410 | | .WORD 110410 | |
| 6919 | 003376 | 036043 | | .WORD 036043 | |
| 6920 | | | | | |
| 6921 | 003400 | 000000 | STG12B: | .WORD 000000 | :PACKED STRING |
| 6922 | 003402 | 000000 | | .WORD 000000 | |
| 6923 | 003404 | 000000 | | .WORD 000000 | |
| 6924 | 003406 | 000000 | | .WORD 000000 | |
| 6925 | 003410 | 000000 | | .WORD 000000 | |
| 6926 | 003412 | 000000 | | .WORD 000000 | |
| 6927 | 003414 | 000000 | | .WORD 000000 | |
| 6928 | 003416 | 106610 | | .WORD 106610 | : 000888- |
| 6929 | | | | | |
| 6930 | 003420 | 000100 | STG12C: | .WORD 000100 | :PACKED STRING |
| 6931 | 003422 | 000000 | | .WORD 000000 | : 40000000000000000000000000000000- |
| 6932 | 003424 | 000000 | | .WORD 000000 | |
| 6933 | 003426 | 000000 | | .WORD 000000 | |
| 6934 | 003430 | 000000 | | .WORD 000000 | |
| 6935 | 003432 | 000000 | | .WORD 000000 | |
| 6936 | 003434 | 000000 | | .WORD 000000 | |
| 6937 | 003436 | 006400 | | .WORD 006400 | |
| 6938 | | | | | |
| 6939 | 003440 | 001001 | SSTG13: | .WORD 001001 | :ZONED STRING |
| 6940 | 003442 | 002003 | | .WORD 002003 | : 1234567891234567891234000891233 |
| 6941 | 003444 | 003005 | | .WORD 003005 | |
| 6942 | 003446 | 004007 | | .WORD 004007 | |
| 6943 | 003450 | 000411 | | .WORD 000411 | |
| 6944 | 003452 | 101602 | | .WORD 101602 | |
| 6945 | 003454 | 072564 | | .WORD 072564 | |
| 6946 | 003456 | 073566 | | .WORD 073566 | |
| 6947 | 003460 | 034470 | | .WORD 034470 | |
| 6948 | 003462 | 031061 | | .WORD 031061 | |
| 6949 | 003464 | 002003 | | .WORD 002003 | |
| 6950 | 003466 | 000000 | | .WORD 000000 | |
| 6951 | 003470 | 004000 | | .WORD 004000 | |
| 6952 | 003472 | 000411 | | .WORD 000411 | |
| 6953 | 003474 | 001402 | | .WORD 001402 | |
| 6954 | 003476 | 000063 | | .WORD 000063 | |
| 6955 | | | | | |
| 6956 | 003500 | 000000 | STG13B: | .WORD 000000 | :ZONED STRING |
| 6957 | 003502 | 000000 | | .WORD 000000 | |
| 6958 | 003504 | 000000 | | .WORD 000000 | |
| 6959 | 003506 | 000000 | | .WORD 000000 | |
| 6960 | 003510 | 000000 | | .WORD 000000 | |
| 6961 | 003512 | 000000 | | .WORD 000000 | |
| 6962 | 003514 | 000000 | | .WORD 000000 | |
| 6963 | 003516 | 000000 | | .WORD 000000 | |

| | | | | | |
|------|--------|--------|---------------|--------|-------------------------------------|
| 6964 | 003520 | 000000 | .WORD | 000000 | |
| 6965 | 003522 | 000000 | .WORD | 000000 | |
| 6966 | 003524 | 000000 | .WORD | 000000 | |
| 6967 | 003526 | 000000 | .WORD | 000000 | |
| 6968 | 003530 | 000000 | .WORD | 000000 | |
| 6969 | 003532 | 000000 | .WORD | 000000 | |
| 6970 | 003534 | 004000 | .WORD | 004000 | : 000888- |
| 6971 | 003536 | 074010 | .WORD | 074010 | |
| 6972 | | | | | |
| 6973 | 003540 | 000004 | STG13C: .WORD | 000004 | : ZONED STRING |
| 6974 | 003542 | 000000 | .WORD | 000000 | : 40000000000000000000000000000000- |
| 6975 | 003544 | 000000 | .WORD | 000000 | |
| 6976 | 003546 | 000000 | .WORD | 000000 | |
| 6977 | 003550 | 000000 | .WORD | 000000 | |
| 6978 | 003552 | 000000 | .WORD | 000000 | |
| 6979 | 003554 | 000000 | .WORD | 000000 | |
| 6980 | 003556 | 000000 | .WORD | 000000 | |
| 6981 | 003560 | 000000 | .WORD | 000000 | |
| 6982 | 003562 | 000000 | .WORD | 000000 | |
| 6983 | 003564 | 000000 | .WORD | 000000 | |
| 6984 | 003566 | 000000 | .WORD | 000000 | |
| 6985 | 003570 | 000000 | .WORD | 000000 | |
| 6986 | 003572 | 000000 | .WORD | 000000 | |
| 6987 | 003574 | 000000 | .WORD | 000000 | |
| 6988 | 003576 | 000160 | .WORD | 000160 | |
| 6989 | | | | | |
| 6990 | 003600 | 042006 | SSTG14: .WORD | 042006 | : PACKED STRING 3X2**31 |
| 6991 | 003602 | 050044 | .WORD | 050044 | : 6442450944 + |
| 6992 | 003604 | 046224 | .WORD | 046224 | |
| 6993 | | | | | |
| 6994 | 003606 | 042006 | SSTG15: .WORD | 42006 | : PACKED STRING 3X2**31 - |
| 6995 | 003610 | 050044 | .WORD | 50044 | : 6442450944 - |
| 6996 | 003612 | 046624 | .WORD | 46624 | |
| 6997 | | | | | |
| 6998 | 003614 | 000000 | SSTG16: .WORD | 000000 | : PACKED STRING |
| 6999 | 003616 | 000000 | .WORD | 000000 | : 0000000000000333+ |
| 7000 | 003620 | 000000 | .WORD | 000000 | |
| 7001 | 003622 | 036063 | .WORD | 036063 | |
| 7002 | | | | | |

```
7004 .SBTTL TEST AND EMULATION OPERANDS
7005 ;TEST OPERANDS
7006 :
7007 003624 TRN:
7008 003624 000000 TR0: .WORD 0
7009 003626 000000 TR1: .WORD 0
7010 003630 000000 TR2: .WORD 0
7011 003632 000000 TR3: .WORD 0
7012 003634 000000 TR4: .WORD 0
7013 003636 000000 TR5: .WORD 0
7014 003640 000000 TR6: .WORD 0
7015 003642 000000 TCC: .WORD 0
7016
7017 ;TEST RESULTS - REGISTERS
7018 :
7019 003644 TRNR:
7020 003644 000000 TR0R: .WORD 0
7021 003646 000000 TR1R: .WORD 0
7022 003650 000000 TR2R: .WORD 0
7023 003652 000000 TR3R: .WORD 0
7024 003654 000000 TR4R: .WORD 0
7025 003656 000000 TR5R: .WORD 0
7026 003660 000000 TR6R: .WORD 0
7027
7028 ;TEST RESULTS - CONDITION CODES
7029 :
7030 003662 000000 TCCR: .WORD 0
7031
7032 ;EMULATION OPERANDS
7033 :
7034 003664 ERN:
7035 003664 000000 ER0: .WORD 0
7036 003666 000000 ER1: .WORD 0
7037 003670 000000 ER2: .WORD 0
7038 003672 000000 ER3: .WORD 0
7039 003674 000000 ER4: .WORD 0
7040 003676 000000 ER5: .WORD 0
7041 003700 000000 ER6: .WORD 0
7042
7043 ;EMULATION RESULTS - REGISTERS
7044 :
7045 003702 ERNR:
7046 003702 000000 ER0R: .WORD 0
7047 003704 000000 ER1R: .WORD 0
7048 003706 000000 ER2R: .WORD 0
7049 003710 000000 ER3R: .WORD 0
7050 003712 000000 ER4R: .WORD 0
7051 003714 000000 ER5R: .WORD 0
7052 003716 000000 ER6R: .WORD 0
7053
7054 ;EMULATION RESULTS - CONDITION CODES
7055 :
7056 003720 000000 ECCR: .WORD 0
```

| | | | .SBTTL | OCTAL CODING FOR EACH CIS INSTRUCTION | |
|------|--------|--------|--|---------------------------------------|------------------|
| | | | :OCTAL CODING FOR EACH CIS INSTRUCTION | | |
| | | | :OINST: | .WORD | : |
| 7058 | | | | 0 | : 0 - UNASSIGNED |
| 7059 | | | | 76030 | : 1 - MOVC |
| 7060 | | | | 76031 | : 2 - MOVRC |
| 7061 | 003722 | 000000 | | 76032 | : 3 - MOVTC |
| 7062 | 003724 | 076030 | | 76040 | : 4 - LOCC |
| 7063 | 003726 | 076031 | | 76041 | : 5 - SKPC |
| 7064 | 003730 | 076032 | | 76042 | : 6 - SCANC |
| 7065 | 003732 | 076040 | | 76043 | : 7 - SPANC |
| 7066 | 003734 | 076041 | | 76044 | :10 - CMPC |
| 7067 | 003736 | 076042 | | 76045 | :11 - MATCHC |
| 7068 | 003740 | 076043 | | 76050 | :12 - ADDN |
| 7069 | 003742 | 076044 | | 76051 | :13 - SUBN |
| 7070 | 003744 | 076045 | | 76052 | :14 - CMPN |
| 7071 | 003746 | 076050 | | 76053 | :15 - CVTNL |
| 7072 | 003750 | 076051 | | 76054 | :16 - CVTPN |
| 7073 | 003752 | 076052 | | 76055 | :17 - CVTNP |
| 7074 | 003754 | 076053 | | 76056 | :20 - ASHN |
| 7075 | 003756 | 076054 | | 76057 | :21 - CVTLN |
| 7076 | 003760 | 076055 | | 76070 | :22 - ADDP |
| 7077 | 003762 | 076056 | | 76071 | :23 - SUBP |
| 7078 | 003764 | 076057 | | 76072 | :24 - CMPP |
| 7079 | 003766 | 076070 | | 76073 | :25 - CVTPL |
| 7080 | 003770 | 076071 | | 76074 | :26 - MULP |
| 7081 | 003772 | 076072 | | 76075 | :27 - DIVP |
| 7082 | 003774 | 076073 | | 76076 | :30 - ASHP |
| 7083 | 003776 | 076074 | | 76077 | :31 - CVTLP |
| 7084 | 004000 | 076075 | | 76020 | :32 - L2DR |
| 7085 | 004002 | 076076 | | 76060 | :33 - L3DR |
| 7086 | 004004 | 076077 | | | |
| 7087 | 004006 | 076020 | | | |
| 7088 | 004010 | 076060 | | | |

7090
7091
7092
7093 004012 000000
7094 004014 000000
7095 004016 000000
7096 004020 000000
7097 004022 000003
7098 004024 000003
7099 004026 000003
7100 004030 000003
7101 004032 000000
7102 004034 000003
7103 004036 000001
7104 004040 000001
7105 004042 000003
7106 004044 000000
7107 004046 000001
7108 004050 000001
7109 004052 000001
7110 004054 000001
7111 004056 000001
7112 004060 000001
7113 004062 000003
7114 004064 000000
7115 004066 000001
7116 004070 000000
7117 004072 000001
7118 004074 000001
7119 004076 000000
7120 004100 000000
7121

.SBTTL CONDITION CODE USAGE RECORD
:CONDITION CODE USAGE RECORD
:LOW 4 BITS OF LOW BYTE = '1' STATE EXERCISED; LOW 4 BITS OF HIGH BYTE = '0' STATE EXERC
CCREC: .WORD 0 : 0 - UNASSIGNED
.WORD 0 : 1 - MOVC
.WORD 0 : 2 - MOVRC
.WORD 0 : 3 - MOVTC
.WORD 3 : 4 - LOCC
.WORD 3 : 5 - SKPC
.WORD 3 : 6 - SCANC
.WORD 3 : 7 - SPANC
.WORD 0 :10 - CMPC
.WORD 3 :11 - MATCHC
.WORD 1 :12 - ADDN
.WORD 1 :13 - SUBN
.WORD 3 :14 - CMPN
.WORD 0 :15 - CVTNL
.WORD 1 :16 - CVTPN
.WORD 1 :17 - CVTNP
.WORD 1 :20 - ASHN
.WORD 1 :21 - CVTLN
.WORD 1 :22 - ADDP
.WORD 1 :23 - SUBP
.WORD 3 :24 - CMPP
.WORD 0 :25 - CVTFL
.WORD 1 :26 - MULP
.WORD 0 :27 - DIVP
.WORD 1 :30 - ASHP
.WORD 1 :31 - CVTLP
.WORD 0 :32 - L2DR
.WORD 0 :33 - L3DR

7123
7124
7125
7126 004102 000000
7127 004104 000000
7128 004106 000000
7129 004110 000000
7130 004112 000000
7131 004114 000000
7132 004116 000000
7133 004120 000000
7134 004122 000000
7135 004124 000000
7136 004126 000000
7137 004130 000000
7138 004132 000000
7139 004134 000000
7140 004136 000000
7141 004140 000000
7142 004142 000000
7143 004144 000000
7144 004146 000000
7145 004150 000000
7146 004152 000000
7147 004154 000000
7148 004156 000000
7149 004160 000000
7150 004162 000000
7151 004164 000000
7152 004166 000000
7153 004170 000000

.SBTTL INTERRUPT LATENCY TABLE
: INTERRUPT LATENCY TABLE (WORST CASE VALUE FOR GIVEN INST)

:LATEN: .WORD 0 ; 0 - UNASSIGNED
 .WORD 0 ; 1 - MOVCL
 .WORD 0 ; 2 - MOVRC
 .WORD 0 ; 3 - MOVTC
 .WORD 0 ; 4 - LOCC
 .WORD 0 ; 5 - SKPC
 .WORD 0 ; 6 - SCANC
 .WORD 0 ; 7 - SPANC
 .WORD 0 ;10 - CMPC
 .WORD 0 ;11 - MATCHC
 .WORD 0 ;12 - ADDN
 .WORD 0 ;13 - SUBN
 .WORD 0 ;14 - CMPN
 .WORD 0 ;15 - CVTNL
 .WORD 0 ;16 - CVTPN
 .WORD 0 ;17 - CVTNP
 .WORD 0 ;20 - ASHN
 .WORD 0 ;21 - CVTLN
 .WORD 0 ;22 - ADDP
 .WORD 0 ;23 - SUBP
 .WORD 0 ;24 - CMPP
 .WORD 0 ;25 - CVTPL
 .WORD 0 ;26 - MULP
 .WORD 0 ;27 - DIVP
 .WORD 0 ;30 - ASHP
 .WORD 0 ;31 - CVTLP
 .WORD 0 ;32 - L2DR
LATEND: .WORD 0 ;33 - L3DR

```
7155      .SBTTL      RANDOM EXERCISE MODE MASK TABLE POINTERS
7156      :RANDOM EXERCISE MODE MASK TABLE POINTERS
7157      :
7158 004172 000000      .MINST: .WORD 0      : 0 - UNASSIGNED
7159 004174 006064      .WORD MMOVC      : 1 - MOV C
7160 004176 006064      .WORD MMOVRC      : 2 - MOV RC
7161 004200 006100      .WORD MMOVTC      : 3 - MOV TC
7162 004202 006134      .WORD MLOCC      : 4 - LOCC
7163 004204 006134      .WORD MSKPC      : 5 - SKPC
7164 004206 006144      .WORD MSCANC      : 6 - SCANC
7165 004210 006144      .WORD MSPANC      : 7 - SPANC
7166 004212 006064      .WORD MCMPC      :10 - CMPC
7167 004214 006160      .WORD MMTCHC      :11 - MATCHC
7168 004216 006240      .WORD MADDN      :12 - ADDN
7169 004220 006240      .WORD MSUBN      :13 - SUBN
7170 004222 006266      .WORD MCMPN      :14 - CMPN
7171 004224 006222      .WORD MCVTNL      :15 - CVTNL
7172 004226 006174      .WORD MCVTPN      :16 - CVTPN
7173 004230 006174      .WORD MCVTNP      :17 - CVTNP
7174 004232 006312      .WORD MASHN      :20 - ASHN
7175 004234 006210      .WORD MCVTLN      :21 - CVTLN
7176 004236 006240      .WORD MADDP      :22 - ADDP
7177 004240 006240      .WORD MSUBP      :23 - SUBP
7178 004242 006266      .WORD MCMPP      :24 - CMPP
7179 004244 006222      .WORD MCVTPL      :25 - CVTPL
7180 004246 006240      .WORD MMULP      :26 - MULP
7181 004250 006240      .WORD MDIVP      :27 - DIVP
7182 004252 006312      .WORD MASHP      :30 - ASHP
7183 004254 006210      .WORD MCVTLP      :31 - CVTLP
```


| | | | .SBTTL | DECIMAL INST DATA TYPE CONTROL WORDS | | |
|------|--------|--------|----------|--------------------------------------|--------|-------------|
| | | | :DECIMAL | INSTRUCTION DATA TYPE CONTROL WORDS | | |
| 7185 | | | | | | |
| 7186 | | | | | | |
| 7187 | | | | | | |
| 7188 | 004256 | 004210 | DECTYP: | .WORD DECTTB-50 | | |
| 7189 | | | | | | |
| 7190 | 004260 | 001400 | DECTTB: | .WORD 1400 | :ADDN | NDESC,PKPTW |
| 7191 | 004262 | 002000 | | .WORD 2000 | : | ZPM,SXTYPE |
| 7192 | 004264 | 001400 | | .WORD 1400 | :SUBN | |
| 7193 | 004266 | 002000 | | .WORD 2000 | | |
| 7194 | 004270 | 001000 | | .WORD 1000 | :CMPN | |
| 7195 | 004272 | 002000 | | .WORD 2000 | | |
| 7196 | 004274 | 000400 | | .WORD 0400 | :CVTNL | |
| 7197 | 004276 | 002000 | | .WORD 2000 | | |
| 7198 | 004300 | 001006 | | .WORD 1006 | :CVTPN | |
| 7199 | 004302 | 004006 | | .WORD 4006 | | |
| 7200 | 004304 | 001060 | | .WORD 1060 | :CVTNP | |
| 7201 | 004306 | 003060 | | .WORD 3060 | | |
| 7202 | 004310 | 001000 | | .WORD 1000 | :ASHN | |
| 7203 | 004312 | 002000 | | .WORD 2000 | | |
| 7204 | 004314 | 000400 | | .WORD 0400 | :CVTLN | |
| 7205 | 004316 | 002000 | | .WORD 2000 | | |
| 7206 | 004320 | 001400 | | .WORD 1400 | :ADDP | |
| 7207 | 004322 | 001777 | | .WORD 1777 | | |
| 7208 | 004324 | 001400 | | .WORD 1400 | :SUBP | |
| 7209 | 004326 | 001777 | | .WORD 1777 | | |
| 7210 | 004330 | 001000 | | .WORD 1000 | :CMPP | |
| 7211 | 004332 | 001777 | | .WORD 1777 | | |
| 7212 | 004334 | 000400 | | .WORD 0400 | :CVTPL | |
| 7213 | 004336 | 001777 | | .WORD 1777 | | |
| 7214 | 004340 | 001400 | | .WORD 1400 | :MULP | |
| 7215 | 004342 | 001777 | | .WORD 1777 | | |
| 7216 | 004344 | 001400 | | .WORD 1400 | :DIVP | |
| 7217 | 004346 | 001777 | | .WORD 1777 | | |
| 7218 | 004350 | 001000 | | .WORD 1000 | :ASHP | |
| 7219 | 004352 | 001777 | | .WORD 1777 | | |
| 7220 | 004354 | 000400 | | .WORD 0400 | :CVTLP | |
| 7221 | 004356 | 001777 | | .WORD 1777 | | |

| | | | .SBTTL | TYPE 0 INDIRECTLY SPECIFIED PARAMETERS | |
|------|--------|--------|---------------|--|-------------|
| | | | :TYPE 0 ENTRY | - INDIRECTLY SPECIFIED PARAMETERS | |
| 7223 | | | | | |
| 7224 | | | | | |
| 7225 | | | | | |
| 7226 | 004360 | 000000 | INSTID: | .WORD 0 | :UNASSIGNED |
| 7227 | 004362 | 006 | | .BYTE 6 | :MOVC |
| 7228 | 004363 | 000 | | .BYTE 0 | |
| 7229 | 004364 | 006 | | .BYTE 6 | :MOVRC |
| 7230 | 004365 | 000 | | .BYTE 0 | |
| 7231 | 004366 | 006 | | .BYTE 6 | :MOVTC |
| 7232 | 004367 | 015 | | .BYTE 15 | |
| 7233 | 004370 | 004 | | .BYTE 4 | :LOCC |
| 7234 | 004371 | 000 | | .BYTE 0 | |
| 7235 | 004372 | 004 | | .BYTE 4 | :SKPC |
| 7236 | 004373 | 000 | | .BYTE 0 | |
| 7237 | 004374 | 006 | | .BYTE 6 | :SCANC |
| 7238 | 004375 | 011 | | .BYTE 11 | |
| 7239 | 004376 | 006 | | .BYTE 6 | :SPANC |
| 7240 | 004377 | 011 | | .BYTE 11 | |
| 7241 | 004400 | 006 | | .BYTE 6 | :CMPC |
| 7242 | 004401 | 011 | | .BYTE 11 | |
| 7243 | 004402 | 006 | | .BYTE 6 | :MATCHC |
| 7244 | 004403 | 011 | | .BYTE 11 | |
| 7245 | 004404 | 007 | | .BYTE 7 | :ADDN |
| 7246 | 004405 | 012 | | .BYTE 12 | |
| 7247 | 004406 | 007 | | .BYTE 7 | :SUBN |
| 7248 | 004407 | 012 | | .BYTE 12 | |
| 7249 | 004410 | 006 | | .BYTE 6 | :CMPN |
| 7250 | 004411 | 011 | | .BYTE 11 | |
| 7251 | 004412 | 003 | | .BYTE 3 | :CVTNL |
| 7252 | 004413 | 000 | | .BYTE 0 | |
| 7253 | 004414 | 005 | | .BYTE 5 | :CVTPN |
| 7254 | 004415 | 000 | | .BYTE 0 | |
| 7255 | 004416 | 005 | | .BYTE 5 | :CVTNP |
| 7256 | 004417 | 000 | | .BYTE 0 | |
| 7257 | 004420 | 006 | | .BYTE 6 | :ASHN |
| 7258 | 004421 | 000 | | .BYTE 0 | |
| 7259 | 004422 | 000 | | .BYTE 0 | :CVTLN |
| 7260 | 004423 | 000 | | .BYTE 0 | |
| 7261 | 004424 | 007 | | .BYTE 7 | :ADDP |
| 7262 | 004425 | 012 | | .BYTE 12 | |
| 7263 | 004426 | 007 | | .BYTE 7 | :SUBP |
| 7264 | 004427 | 012 | | .BYTE 12 | |
| 7265 | 004430 | 006 | | .BYTE 6 | :CMPP |
| 7266 | 004431 | 011 | | .BYTE 11 | |
| 7267 | 004432 | 003 | | .BYTE 3 | :CVTPL |
| 7268 | 004433 | 000 | | .BYTE 0 | |
| 7269 | 004434 | 007 | | .BYTE 7 | :MULP |
| 7270 | 004435 | 012 | | .BYTE 12 | |
| 7271 | 004436 | 007 | | .BYTE 7 | :DIVP |
| 7272 | 004437 | 012 | | .BYTE 12 | |
| 7273 | 004440 | 006 | | .BYTE 6 | :ASHP |
| 7274 | 004441 | 000 | | .BYTE 0 | |
| 7275 | 004442 | 000 | | .BYTE 0 | :CVTLP |
| 7276 | 004443 | 000 | | .BYTE 0 | |

7278
7279
7280
7281
7282 004444
7284 004444 047515 041526 000000
7285 004452 072230
7286 004454 047515 041526 000061
7287 004462 072304
7288 004464 047515 041526 000062
7289 004472 072360
7290 004474 047514 041503 000000
7291 004502 072434
7292 004504 047514 041503 000061
7293 004512 072510
7294 004514 047514 041503 000062
7295 004522 072564
7296 004524 046503 041520 000000
7297 004532 072640
7298 004534 046503 041520 000061
7299 004542 072714
7300 004544 046503 041520 000062
7301 004552 072770
7302 004554 047515 051126 000103
7303 004562 073044
7304 004564 047515 051126 030503
7305 004572 073120
7306 004574 047515 051126 031103
7307 004602 073174
7308 004604 047515 052126 000103
7309 004612 073250
7310 004614 047515 052126 030503
7311 004622 073324
7312 004624 047515 052126 031103
7313 004632 073400
7314 004634 045523 041520 000000
7315 004642 073454
7316 004644 045523 041520 000061
7317 004652 073530
7318 004654 045523 041520 000062
7319 004662 073604
7320 004664 040515 041524 000000
7321 004672 073660
7322 004674 040515 041524 000061
7323 004702 073734
7324 004704 040515 041524 000062
7325 004712 074010
7326 004714 041523 047101 000103
7327 004722 074064
7328 004724 041523 047101 030503
7329 004732 074140
7330 004734 041523 047101 031103
7331 004742 074214
7332 004744 050123 047101 000103

.SBTTL ASCII TABLE FOR CIS INST NMEUMONICS
:ASCII TABLE FOR CIS INSTRUCTION NMEUMONICS
:ASZINS:
.ASCII /MOVC/<0><0>
.WORD IMOVC
.ASCII /MOVC1/<0>
.WORD IMOVC1
.ASCII /MOVC2/<0>
.WORD IMOVC2
.ASCII /LOCC/<0><0>
.WORD ILOCC
.ASCII /LOCC1/<0>
.WORD ILOCC1
.ASCII /LOCC2/<0>
.WORD ILOCC2
.ASCII /CMPC/<0><0>
.WORD ICMPC
.ASCII /CMPC1/<0>
.WORD ICMPC1
.ASCII /CMPC2/<0>
.WORD ICMPC2
.ASCII /MOVRC/<0>
.WORD IMOVR
.ASCII /MOVRC1/
.WORD IMOVR1
.ASCII /MOVRC2/
.WORD IMOVR2
.ASCII /MOVTC/<0>
.WORD IMOVT
.ASCII /MOVTC1/
.WORD IMOVT1
.ASCII /MOVTC2/
.WORD IMOVT2
.ASCII /SKPC/<0><0>
.WORD ISKPC
.ASCII /SKPC1/<0>
.WORD ISKPC1
.ASCII /SKPC2/<0>
.WORD ISKPC2
.ASCII /MATC/<0><0>
.WORD IMATC
.ASCII /MATC1/<0>
.WORD IMATC1
.ASCII /MATC2/<0>
.WORD IMATC2
.ASCII /SCANC/<0>
.WORD ISCAN
.ASCII /SCANC1/
.WORD ISCAN1
.ASCII /SCANC2/
.WORD ISCAN2
.ASCII /SPANC/<0>

| | | | | | |
|------|--------|--------|--------|--------|---------------------|
| 7333 | 004752 | 074270 | | | .WORD ISPAN |
| 7334 | 004754 | 050123 | 047101 | 030503 | .ASCII /SPANC1/ |
| 7335 | 004762 | 074344 | | | .WORD ISPAN1 |
| 7336 | 004764 | 050123 | 047101 | 031103 | .ASCII /SPANC2/ |
| 7337 | 004772 | 074420 | | | .WORD ISPAN2 |
| 7338 | 004774 | 053103 | 050124 | 000116 | .ASCII /CVTPN/<0> |
| 7339 | 005002 | 074474 | | | .WORD ICPZ |
| 7340 | 005004 | 053103 | 050124 | 030516 | .ASCII /CVTPN1/ |
| 7341 | 005012 | 074550 | | | .WORD ICPZ1 |
| 7342 | 005014 | 053103 | 047124 | 000120 | .ASCII /CVTNP/<0> |
| 7343 | 005022 | 074624 | | | .WORD ICPZ |
| 7344 | 005024 | 053103 | 047124 | 030520 | .ASCII /CVTNP1/ |
| 7345 | 005032 | 074700 | | | .WORD ICPZ1 |
| 7346 | 005034 | 053103 | 047124 | 031120 | .ASCII /CVTNP2/ |
| 7347 | 005042 | 074754 | | | .WORD ICPZ2 |
| 7348 | 005044 | 053103 | 046124 | 000120 | .ASCII /CVTLP/<0> |
| 7349 | 005052 | 075030 | | | .WORD ICLP |
| 7350 | 005054 | 053103 | 046124 | 030520 | .ASCII /CVTLP1/ |
| 7351 | 005062 | 075104 | | | .WORD ICLP1 |
| 7352 | 005064 | 053103 | 046124 | 031120 | .ASCII /CVTLP2/ |
| 7353 | 005072 | 075160 | | | .WORD ICLP2 |
| 7354 | 005074 | 053103 | 046124 | 000116 | .ASCII /CVTLN/<0> |
| 7355 | 005102 | 075234 | | | .WORD ICLZ |
| 7356 | 005104 | 053103 | 046124 | 030516 | .ASCII /CVTLN1/ |
| 7357 | 005112 | 075310 | | | .WORD ICLZ1 |
| 7358 | 005114 | 053103 | 050124 | 000114 | .ASCII /CVTPL/<0> |
| 7359 | 005122 | 075364 | | | .WORD ICPL |
| 7360 | 005124 | 053103 | 050124 | 030514 | .ASCII /CVTPL1/ |
| 7361 | 005132 | 075440 | | | .WORD ICPL1 |
| 7362 | 005134 | 053103 | 050124 | 031114 | .ASCII /CVTPL2/ |
| 7363 | 005142 | 075514 | | | .WORD ICPL2 |
| 7364 | 005144 | 053103 | 050124 | 031514 | .ASCII /CVTPL3/ |
| 7365 | 005152 | 075570 | | | .WORD ICPL3 |
| 7366 | 005154 | 053103 | 047124 | 000114 | .ASCII /CVTNL/<0> |
| 7367 | 005162 | 075644 | | | .WORD ICZL |
| 7368 | 005164 | 053103 | 047124 | 030514 | .ASCII /CVTNL1/ |
| 7369 | 005172 | 075720 | | | .WORD ICZL1 |
| 7370 | 005174 | 053103 | 047124 | 031114 | .ASCII /CVTNL2/ |
| 7371 | 005202 | 075774 | | | .WORD ICZL2 |
| 7372 | 005204 | 042101 | 050104 | 000000 | .ASCII /ADDP/<0><0> |
| 7373 | 005212 | 076050 | | | .WORD IADDP |
| 7374 | 005214 | 042101 | 050104 | 000061 | .ASCII /ADDP1/<0> |
| 7375 | 005222 | 076124 | | | .WORD IADDP1 |
| 7376 | 005224 | 042101 | 050104 | 000062 | .ASCII /ADDP2/<0> |
| 7377 | 005232 | 076200 | | | .WORD IADDP2 |
| 7378 | 005234 | 042101 | 050104 | 000063 | .ASCII /ADDP3/<0> |
| 7379 | 005242 | 076254 | | | .WORD IADDP3 |
| 7380 | 005244 | 042101 | 050104 | 000064 | .ASCII /ADDP4/<0> |
| 7381 | 005252 | 076330 | | | .WORD IADDP4 |
| 7382 | 005254 | 042101 | 047104 | 000000 | .ASCII /ADDN/<0><0> |
| 7383 | 005262 | 076404 | | | .WORD IADDN |
| 7384 | 005264 | 042101 | 047104 | 000061 | .ASCII /ADDN1/<0> |
| 7385 | 005272 | 076460 | | | .WORD IADDN1 |
| 7386 | 005274 | 042101 | 047104 | 000062 | .ASCII /ADDN2/<0> |

| | | | | | |
|------|--------|--------|--------|--------|---------------------|
| 7387 | 005302 | 076534 | | | .WORD IADDN2 |
| 7388 | 005304 | 042101 | 047104 | 000063 | .ASCII /ADDN3/<0> |
| 7389 | 005312 | 076610 | | | .WORD IADDN3 |
| 7390 | 005314 | 042101 | 047104 | 000064 | .ASCII /ADDN4/<0> |
| 7391 | 005322 | 076664 | | | .WORD IADDN4 |
| 7392 | 005324 | 052523 | 050102 | 000000 | .ASCII /SUBP/<0><0> |
| 7393 | 005332 | 077014 | | | .WORD ISUBP |
| 7394 | 005334 | 052523 | 050102 | 000061 | .ASCII /SUBP1/<0> |
| 7395 | 005342 | 077070 | | | .WORD ISUBP1 |
| 7396 | 005344 | 052523 | 050102 | 000062 | .ASCII /SUBP2/<0> |
| 7397 | 005352 | 077144 | | | .WORD ISUBP2 |
| 7398 | 005354 | 052523 | 050102 | 000063 | .ASCII /SUBP3/<0> |
| 7399 | 005362 | 077220 | | | .WORD ISUBP3 |
| 7400 | 005364 | 052523 | 050102 | 000064 | .ASCII /SUBP4/<0> |
| 7401 | 005372 | 077274 | | | .WORD ISUBP4 |
| 7402 | 005374 | 052523 | 047102 | 000000 | .ASCII /SUBN/<0><0> |
| 7403 | 005402 | 077350 | | | .WORD ISUBN |
| 7404 | 005404 | 052523 | 047102 | 000061 | .ASCII /SUBN1/<0> |
| 7405 | 005412 | 077424 | | | .WORD ISUBN1 |
| 7406 | 005414 | 052523 | 047102 | 000062 | .ASCII /SUBN2/<0> |
| 7407 | 005422 | 077500 | | | .WORD ISUBN2 |
| 7408 | 005424 | 052523 | 047102 | 000063 | .ASCII /SUBN3/<0> |
| 7409 | 005432 | 077554 | | | .WORD ISUBN3 |
| 7410 | 005434 | 052523 | 047102 | 000064 | .ASCII /SUBN4/<0> |
| 7411 | 005442 | 077630 | | | .WORD ISUBN4 |
| 7412 | 005444 | 046503 | 050120 | 000000 | .ASCII /CMPP/<0><0> |
| 7413 | 005452 | 077704 | | | .WORD ICMPP |
| 7414 | 005454 | 046503 | 050120 | 000061 | .ASCII /CMPP1/<0> |
| 7415 | 005462 | 077760 | | | .WORD ICMPP1 |
| 7416 | 005464 | 046503 | 050120 | 000062 | .ASCII /CMPP2/<0> |
| 7417 | 005472 | 100034 | | | .WORD ICMPP2 |
| 7418 | 005474 | 046503 | 050120 | 000063 | .ASCII /CMPP3/<0> |
| 7419 | 005502 | 100110 | | | .WORD ICMPP3 |
| 7420 | 005504 | 046503 | 050120 | 000064 | .ASCII /CMPP4/<0> |
| 7421 | 005512 | 100164 | | | .WORD ICMPP4 |
| 7422 | 005514 | 046503 | 047120 | 000000 | .ASCII /CMPN/<0><0> |
| 7423 | 005522 | 100240 | | | .WORD ICMPN |
| 7424 | 005524 | 046503 | 047120 | 000061 | .ASCII /CMPN1/<0> |
| 7425 | 005532 | 100314 | | | .WORD ICMPN1 |
| 7426 | 005534 | 046503 | 047120 | 000062 | .ASCII /CMPN2/<0> |
| 7427 | 005542 | 100370 | | | .WORD ICMPN2 |
| 7428 | 005544 | 046503 | 047120 | 000063 | .ASCII /CMPN3/<0> |
| 7429 | 005552 | 100444 | | | .WORD ICMPN3 |
| 7430 | 005554 | 046503 | 047120 | 000064 | .ASCII /CMPN4/<0> |
| 7431 | 005562 | 100520 | | | .WORD ICMPN4 |
| 7432 | 005564 | 051501 | 050110 | 000000 | .ASCII /ASHP/<0><0> |
| 7433 | 005572 | 100574 | | | .WORD IASHP |
| 7434 | 005574 | 051501 | 050110 | 000061 | .ASCII /ASHP1/<0> |
| 7435 | 005602 | 100650 | | | .WORD IASHP1 |
| 7436 | 005604 | 051501 | 050110 | 000062 | .ASCII /ASHP2/<0> |
| 7437 | 005612 | 100724 | | | .WORD IASHP2 |
| 7438 | 005614 | 051501 | 047110 | 000000 | .ASCII /ASHN/<0><0> |
| 7439 | 005622 | 101000 | | | .WORD IASHN |
| 7440 | 005624 | 051501 | 047110 | 000061 | .ASCII /ASHN1/<0> |

| | | | | | |
|------|--------|--------|--------|--------|-----------------------|
| 7441 | 005632 | 101054 | | | .WORD IASHN1 |
| 7442 | 005634 | 051501 | 047110 | 000062 | .ASCII /ASHN2/<0> |
| 7443 | 005642 | 101130 | | | .WORD IASHN2 |
| 7444 | 005644 | 052515 | 050114 | 000000 | .ASCII /MULP/<0><0> |
| 7445 | 005652 | 101204 | | | .WORD IMULP |
| 7446 | 005654 | 052515 | 050114 | 000061 | .ASCII /MULP1/<0> |
| 7447 | 005662 | 101260 | | | .WORD IMULP1 |
| 7448 | 005664 | 052515 | 050114 | 000062 | .ASCII /MULP2/<0> |
| 7449 | 005672 | 101334 | | | .WORD IMULP2 |
| 7450 | 005674 | 052515 | 050114 | 000063 | .ASCII /MULP3/<0> |
| 7451 | 005702 | 101410 | | | .WORD IMULP3 |
| 7452 | 005704 | 052515 | 050114 | 000064 | .ASCII /MULP4/<0> |
| 7453 | 005712 | 101464 | | | .WORD IMULP4 |
| 7454 | 005714 | 052515 | 050114 | 000065 | .ASCII /MULP5/<0> |
| 7455 | 005722 | 101540 | | | .WORD IMULP5 |
| 7456 | 005724 | 052515 | 050114 | 000066 | .ASCII /MULP6/<0> |
| 7457 | 005732 | 101614 | | | .WORD IMULP6 |
| 7458 | 005734 | 052515 | 050114 | 000067 | .ASCII /MULP7/<0> |
| 7459 | 005742 | 101670 | | | .WORD IMULP7 |
| 7460 | 005744 | 044504 | 050126 | 000000 | .ASCII /DIVP/<0><0> |
| 7461 | 005752 | 101744 | | | .WORD IDIVP |
| 7462 | 005754 | 044504 | 050126 | 000061 | .ASCII /DIVP1/<0> |
| 7463 | 005762 | 102020 | | | .WORD IDIVP1 |
| 7464 | 005764 | 044504 | 050126 | 000062 | .ASCII /DIVP2/<0> |
| 7465 | 005772 | 102074 | | | .WORD IDIVP2 |
| 7466 | 005774 | 044504 | 050126 | 000063 | .ASCII /DIVP3/<0> |
| 7467 | 006002 | 102150 | | | .WORD IDIVP3 |
| 7468 | 006004 | 044504 | 050126 | 000064 | .ASCII /DIVP4/<0> |
| 7469 | 006012 | 102224 | | | .WORD IDIVP4 |
| 7470 | 006014 | 044504 | 050126 | 000065 | .ASCII /DIVP5/<0> |
| 7471 | 006022 | 102300 | | | .WORD IDIVP5 |
| 7472 | 006024 | 044504 | 050126 | 000066 | .ASCII /DIVP6/<0> |
| 7473 | 006032 | 102354 | | | .WORD IDIVP6 |
| 7475 | 006034 | 031114 | 000104 | 000000 | .ASCII /L2D/<0><0><0> |
| 7476 | 006042 | 072100 | | | .WORD IL2D |
| 7477 | 006044 | 031514 | 000104 | 000000 | .ASCII /L3D/<0><0><0> |
| 7478 | 006052 | 072154 | | | .WORD IL3D |
| 7479 | 006054 | 000000 | | | .WORD 0 |
| 7480 | 006056 | 000000 | | | .WORD 0 |
| 7481 | 006060 | 000000 | | | .WORD 0 |
| 7482 | 006062 | 000000 | | | .WORD 0 |

```
7484      .SBTTL          RANDOM EXERCISE MASK TABLES
7485      :
7486      :RANDOM EXERCISING MASK TABLES
7487      :
7488
7489      006064      MMOVC:
7490      006064      MCMPC:
7491      006064      MMOVRC:
7493      006064      174000      .WORD BS2      :IP1 MASK      (LEN)
7494      006066      176000      .WORD BS4      :IP2      (ADR)
7495      006070      174000      .WORD BS2      :IP3      (LEN)
7496      006072      174000      .WORD BS2      :IP4      (ADR)
7504      006074      177400      .WORD BY      :IP5      (FILL)
7505      006076      125252      .WORD EOT
7506
7507
7508
7509
7510      006100      MMOVTC:
7512      006100      174000      .WORD BS2      :IP1 MASK      (LEN)
7513      006102      176000      .WORD BS4      :IP2      (ADR)
7514      006104      174000      .WORD BS2      :IP3      (LEN)
7515      006106      002000      .WORD MPO2000 :IP4      (ADR)
7523      006110      177400      .WORD BY      :IP5      (FILL)
7524      006112      177777      .WORD BSNULL   :IP6
7525      006114      177777      .WORD BSNULL   :IP7
7526      006116      177777      .WORD BSNULL   :IP10
7527      006120      177777      .WORD BSNULL   :IP11
7528      006122      177777      .WORD BSNULL   :IP12
7529      006124      177777      .WORD BSNULL   :IP13
7530      006126      177777      .WORD BSNULL   :IP14
7531      006130      022000      .WORD MPO2200 :IP15      (TABLE ADR)
7532      006132      125252      .WORD EOT
7533
```

```
7535  
7536  
7537  
7538  
7539 006134  
7540 006134  
7542 006134 174000  
7543 006136 174000  
7549 006140 177400  
7550 006142 125252  
7551  
7552  
7553  
7554 006144  
7555 006144  
7557 006144 174000  
7558 006146 174000  
7564 006150 177400  
7565 006152 177400  
7567 006154 174000  
7572 006156 125252  
7573  
7574  
7575 006160  
7577 006160 177000  
7578 006162 176000  
7579 006164 174000  
7580 006166 174000  
7588 006170 177400  
7589 006172 125252
```

;
:RANDOM EXERCISE MASK TABLES (CONTINUED)
;
MLOCC:
MSKPC:
 .WORD BS2 :IP1 MASK (LEN)
 .WORD BS2 :IP2 (ADR)
 .WORD BY :IP3 (CHAR)
 .WORD EOT

MSCANC:
MSPANC:
 .WORD BS2 :IP1 MASK (LEN)
 .WORD BS2 :IP2 (ADR)
 .WORD BY :IP3 (TABLE LEN)
 .WORD BY :IP4 (TABLE MASK)
 .WORD BS2 :IP5 (TABLE ADR)
 .WORD EOT

MMTCHC:
 .WORD BS8 :IP1 MASK (LEN)
 .WORD BS4 :IP2 (ADR)
 .WORD BS2 :IP3 (LEN)
 .WORD BS2 :IP4 (ADR)
 .WORD BY :IP5 (DATA)
 .WORD EOT


```
7591
7592
7593      ;RANDOM EXERCISE MODE MASK TABLES (CONTINUED)
7594      ;
7595
7596 006174 MCVTNP:
7597 006174 177740 MCVTPN: .WORD BS128      ;IP1 MASK      (LEN)
7599 006176 176000      .WORD BS4        ;IP2          (ADR)
7604 006200 177740      .WORD BS128     ;IP3          (LEN)
7606 006202 004000      .WORD MPO4000   ;IP4          (ADR)
7611 006204 152525      .WORD DSCPTR    ;IP5          (DESC POINTER)
7612 006206 125252      .WORD EOT
7613
7614
7615 006210 MCVTLP:
7616 006210 000000 MCVTLN: .WORD WD        ;IP1 MASK      (LONG-HIGH)
7617 006212 000000      .WORD WD        ;IP2          (LONG-LOW)
7618 006214 177740      .WORD BS128     ;IP3          (LEN)
7620 006216 174000      .WORD BS2       ;IP4          (ADR)
7625 006220 125252      .WORD EOT
7626
7627
7628 006222 MCVTPL:
7629 006222 177740 MCVTNL: .WORD BS128     ;IP1 MASK      (LEN)
7631 006224 174000      .WORD BS2       ;IP2          (ADR)
7636 006226 152525      .WORD DSCPTR    ;IP3          (DESC POINTER)
7637 006230 177777      .WORD BSNULL    ;IP4
7638 006232 177777      .WORD BSNULL    ;IP5
7639 006234 177400      .WORD BY        ;IP6          (DATA)
7640 006236 125252      .WORD EOT
```

```
7642  
7643  
7644  
7645  
7646  
7647 006240  
7648 006240  
7649 006240  
7650 006240  
7651 006240  
7652 006240  
7653 006240 177740  
7655 006242 177000  
7656 006244 177740  
7657 006246 002000  
7658 006250 177740  
7659 006252 004040  
7668 006254 152525  
7669 006256 177777  
7670 006260 177777  
7671 006262 152525  
7672 006264 125252  
7673  
7674  
7675 006266  
7676 006266  
7677 006266 177740  
7679 006270 176000  
7680 006272 177740  
7681 006274 004000  
7688 006276 177400  
7689 006300 152525  
7690 006302 177777  
7691 006304 177777  
7692 006306 152525  
7693 006310 125252
```

;
;RANDOM EXERCISE MODE MASK TABLES (CONTINUED)
;
MADDP:
MADDN:
MSUBP:
MSUBN:
MMULP:
MDIVP:

| | | |
|---------------|-----------|----------------|
| .WORD BS128 | :IP1 MASK | (LEN) |
| .WORD BS8 | :IP2 | (ADR) |
| .WORD BS128 | :IP3 | (LEN) |
| .WORD MPO2000 | :IP4 | (ADR) |
| .WORD BS128 | :IP5 | (LEN) |
| .WORD MPO4040 | :IP6 | (ADR) |
| .WORD DSCPTR | :IP7 | (DESC POINTER) |
| .WORD BSNUL | :IP10 | |
| .WORD BSNUL | :IP11 | |
| .WORD DSCPTR | :IP12 | (DESC POINTER) |
| .WORD EOT | | |

MCMPN:
MCMPP:

| | | |
|---------------|-----------|----------------|
| .WORD BS128 | :IP1 MASK | (LEN) |
| .WORD BS4 | :IP2 | (ADR) |
| .WORD BS128 | :IP3 | (LEN) |
| .WORD MPO4000 | :IP4 | (ADR) |
| .WORD BY | :IP5 | (DATA) |
| .WORD DSCPTR | :IP6 | (DESC POINTER) |
| .WORD BSNUL | :IP7 | |
| .WORD BSNUL | :IP10 | |
| .WORD DSCPTR | :IP11 | (DESC POINTER) |
| .WORD EOT | | |

7695
7696
7697
7698
7699
7700 006312
7701 006312
7702 006312 177740
7704 006314 176000
7705 006316 170000
7706 006320 177740
7707 006322 004000
7715 006324 152525
7716 006326 125252

:
:RANDOM EXERCISE MODE MASK TABLES (CONTINUED)
:

MASHP:
MASHN:
.WORD BS128 :IP1 MASK (LEN)
.WORD BS4 :IP2 (ADR)
.WORD BS0 :IP3 (RD,SC)
.WORD BS128 :IP4 (LEN)
.WORD MPO4000 :IP5 (ADR)
.WORD DSCPTR :IP6 (DESC POINTER)
.WORD EOT

```
7746  
7747  
7748  
7749  
7750 006330  
7751 006330  
7752 006330 151600  
7753 006332 010100  
7754 006334 010201  
7755 006336 010302  
7756 006340 010504  
7757 006342 020403  
7758 006344 031001  
7759 006346 031303  
7760 006350 041001  
7761 006352 041323  
7762 006354 051300  
7763 006356 060000  
7764 006360 071123  
7765 006362 001312  
7766 006364 070601  
7767 006366 001007  
7768 006370 100000  
7769 006372 111300  
7770 006374 122400  
7771 006376 001300  
7772 006400 000000  
7773 006402 007000  
7774 006404 137000  
7775 006406 140000  
7776  
7777
```

```
.SBTTL          CIS INST FLOW TABLES  
:CIS INSTRUCTION FLOW TABLES  
:  
:XMOVC:          :MOV C FLOW TABLE S1=SRC, S2=DST  
:XMOVRC:         :MOVRC FLOW TABLE S1=SRC, S2=DST  
:                :LOAD SPECIAL HANDLING WORD FROM PTP16  
:                :LOAD TR0 FROM PTP01  
:                :LOAD TR1 FROM PTP02  
:                :LOAD TR2 FROM PTP03  
:                :LOAD TR4 FROM PTP05  
:                :GENERATE TR3 FROM PTP04  
:                :VERIFY THAT S1.ADR-S1.SURR.LEN >=20  
:                :VERIFY THAT S2.ADR-S2.SURR.LEN >= 20  
:                :VERIFY THAT S1.ADR+S1.LEN+S1.SURR.LEN < TBLEN  
:                :VERIFY THAT S2.ADR+S2.LEN+S2.SURR.LEN < TBLEN  
:                :ADD TEST BUFFER ADDRESS TO TR1 AND TR3  
:                :INITIALIZE TEST BUFFER  
:                :INSERT S2 & S2.SURR STRINGS IN TEST BUFFER  
:  
:                :INSERT S1 & S1.SURR STRINGS IN TEST BUFFER  
:  
:                :COPY TEST BUFFER INTO EMULATION BUFFER  
:                :SETUP EMULATION OPERANDS & EMULATE INST  
:                :SETUP CC & REGS AND EXECUTE CIS INST.  
:  
:  
:                :COMPARE RESULTS  
:                :UPDATE PTRS FOR NEXT TEST CONDITION.  
:                :RETURN TO START EXECUTING NEXT TEST CONDITION.
```

| | | | | | |
|------|--------|--------|--------------|--|--|
| 7779 | | | | | |
| 7780 | 006410 | | XMOVTC: | :MOVRC FLOW TABLE S1=SRC, S2=DST | |
| 7781 | 006410 | 151600 | .WORD 151600 | :LOAD SPECIAL HANDLING WORD FROM PTP16 | |
| 7782 | 006412 | 010100 | .WORD 010100 | :LOAD TR0 FROM PTP01 | |
| 7783 | 006414 | 010201 | .WORD 010201 | :LOAD TR1 FROM PTP02 | |
| 7784 | 006416 | 010302 | .WORD 010302 | :LOAD TR2 FROM PTP03 | |
| 7785 | 006420 | 010504 | .WORD 010504 | :LOAD TR4 FROM PTP05 | |
| 7786 | 006422 | 011505 | .WORD 011505 | :LOAD TR5 FROM PTP15 (TRANSLATION TABLE) | |
| 7787 | 006424 | 020403 | .WORD 020403 | :GENERATE TR3 FROM PTP04 | |
| 7788 | 006426 | 031001 | .WORD 031001 | :VERIFY THAT S1.ADR-S1.SURR.LEN >=20 | |
| 7789 | 006430 | 031303 | .WORD 031303 | :VERIFY THAT S2.ADR-S2.SURR.LEN >= 20 | |
| 7790 | 006432 | 041001 | .WORD 041001 | :VERIFY THAT S1.ADR+S1.LEN+S1.SURR.LEN < TBLN | |
| 7791 | 006434 | 041323 | .WORD 041323 | :VERIFY THAT S2.ADR+S2.LEN+S2.SURR.LEN < TBLN | |
| 7792 | 006436 | 051300 | .WORD 051300 | :ADD TEST BUFFER ADDRESS TO TR1 AND TR3 | |
| 7793 | 006440 | 060000 | .WORD 060000 | :INITIALIZE TEST BUFFER | |
| 7794 | 006442 | 071123 | .WORD 071123 | :INSERT S2 & S2.SURR STRINGS IN TEST BUFFER | |
| 7795 | 006444 | 001312 | .WORD 001312 | | |
| 7796 | 006446 | 070601 | .WORD 070601 | :INSERT S1 & S1.SURR STRINGS IN TEST BUFFER | |
| 7797 | 006450 | 001007 | .WORD 001007 | | |
| 7798 | 006452 | 100000 | .WORD 100000 | :COPY TEST BUFFER INTO EMULATION BUFFER | |
| 7799 | 006454 | 111300 | .WORD 111300 | :SETUP EMULATION OPERANDS & EMULATE INST | |
| 7800 | 006456 | 122450 | .WORD 122450 | :SETUP CC & REGS AND EXECUTE CIS INST. | |
| 7801 | 006460 | 001300 | .WORD 001300 | | |
| 7802 | 006462 | 000000 | .WORD 000000 | | |
| 7803 | 006464 | 007000 | .WORD 007000 | | |
| 7804 | 006466 | 137000 | .WORD 137000 | :COMPARE RESULTS | |
| 7805 | 006470 | 140000 | .WORD 140000 | :UPDATE PTRS FOR NEXT TEST CONDITION. | |
| 7806 | | | | : RETURN TO START EXECUTING NEXT TEST CONDITION. | |
| 7807 | | | | | |

| | | | | |
|------|--------|--------|--------------|--|
| 7809 | | | | |
| 7810 | | | | |
| 7811 | 006472 | | XCMPC: | :CMPC FLOW TABLE S1=SRC1, S2=SRC2 |
| 7812 | 006472 | 151500 | .WORD 151500 | :LOAD SPECIAL HANDLING WORD FROM PTP15 |
| 7813 | 006474 | 010100 | .WORD 010100 | :LOAD TR0 FROM PTP01 |
| 7814 | 006476 | 010201 | .WORD 010201 | :LOAD TR1 FROM PTP02 |
| 7815 | 006500 | 010302 | .WORD 010302 | :LOAD TR2 FROM PTP03 |
| 7816 | 006502 | 010504 | .WORD 010504 | :LOAD TR4 FROM PTP05 |
| 7817 | 006504 | 020403 | .WORD 020403 | :GENERATE TR3 FROM PTP04 |
| 7818 | 006506 | 031001 | .WORD 031001 | :VERIFY THAT S1.ADR-S1.SURR.LEN >=20 |
| 7819 | 006510 | 031303 | .WORD 031303 | :VERIFY THAT S2.ADR-S2.SURR.LEN >= 20 |
| 7820 | 006512 | 041001 | .WORD 041001 | :VERIFY THAT S1.ADR+S1.LEN+S1.SURR.LEN < TBLEN |
| 7821 | 006514 | 041323 | .WORD 041323 | :VERIFY THAT S2.ADR+S2.LEN+S2.SURR.LEN < TBLEN |
| 7822 | 006516 | 051300 | .WORD 051300 | :ADD TEST BUFFER ADDRESS TO TR1 AND TR3 |
| 7823 | 006520 | 060000 | .WORD 060000 | :INITIALIZE TEST BUFFER |
| 7824 | 006522 | 071123 | .WORD 071123 | :INSERT S2 & S2.SURR STRINGS IN TEST BUFFER |
| 7825 | 006524 | 001312 | .WORD 001312 | |
| 7826 | 006526 | 070601 | .WORD 070601 | :INSERT S1 & S1.SURR STRINGS IN TEST BUFFER |
| 7827 | 006530 | 001007 | .WORD 001007 | |
| 7828 | 006532 | 100000 | .WORD 100000 | :COPY TEST BUFFER INTO EMULATION BUFFER |
| 7829 | 006534 | 111300 | .WORD 111300 | :SETUP EMULATION OPERANDS & EMULATE INST |
| 7830 | 006536 | 122400 | .WORD 122400 | :SETUP CC & REGS AND EXECUTE CIS INST. |
| 7831 | 006540 | 001300 | .WORD 001300 | |
| 7832 | 006542 | 000000 | .WORD 000000 | |
| 7833 | 006544 | 007000 | .WORD 007000 | |
| 7834 | 006546 | 131370 | .WORD 131370 | :COMPARE RESULTS |
| 7835 | 006550 | 007000 | .WORD 007000 | |
| 7836 | 006552 | 140000 | .WORD 140000 | :UPDATE PTRS FOR NEXT TEST CONDITION. |
| 7837 | | | | : RETURN TO START EXECUTING NEXT TEST CONDITION. |

| | | | | | |
|------|--------|--------|----------|--------|--|
| 7839 | | | | | |
| 7840 | 006554 | | XMATCHC: | | :MATCHC FLOW TABLE S1=SRC, S2=OBJ |
| 7841 | 006554 | 151500 | .WORD | 151500 | :LOAD SPECIAL HANDLING WORD FROM PTP15 |
| 7842 | 006556 | 010300 | .WORD | 010300 | :LOAD TR0 FROM PTP03 |
| 7843 | 006560 | 010102 | .WORD | 010102 | :LOAD TR2 FROM PTP01 |
| 7844 | 006562 | 010203 | .WORD | 010203 | :LOAD TR3 FROM PTP02 |
| 7845 | 006564 | 010504 | .WORD | 010504 | :LOAD TR4 FROM PTP05 |
| 7846 | 006566 | 020401 | .WORD | 020401 | :GENERATE TR1 FROM PTP04 |
| 7847 | 006570 | 031301 | .WORD | 031301 | :VERIFY THAT S1.ADR-S1.SURR.LEN >=20 |
| 7848 | 006572 | 031003 | .WORD | 031003 | :VERIFY THAT S2.ADR-S2.SURR.LEN >= 20 |
| 7849 | 006574 | 041301 | .WORD | 041301 | :VERIFY THAT S1.ADR+S1.LEN+S1.SURR.LEN < TBLEN |
| 7850 | 006576 | 041023 | .WORD | 041023 | :VERIFY THAT S2.ADR+S2.LEN+S2.SURR.LEN < TBLEN |
| 7851 | 006600 | 051300 | .WORD | 051300 | :ADD TEST BUFFER ADDRESS TO TR1 AND TR3 |
| 7852 | 006602 | 060000 | .WORD | 060000 | :INITIALIZE TEST BUFFER |
| 7853 | 006604 | 070623 | .WORD | 070623 | :INSERT S2 & S2.SURR STRINGS IN TEST BUFFER |
| 7854 | 006606 | 001007 | .WORD | 001007 | |
| 7855 | 006610 | 071101 | .WORD | 071101 | :INSERT S1 & S1.SURR STRINGS IN TEST BUFFER |
| 7856 | 006612 | 001312 | .WORD | 001312 | |
| 7857 | 006614 | 100000 | .WORD | 100000 | :COPY TEST BUFFER INTO EMULATION BUFFER |
| 7858 | 006616 | 111300 | .WORD | 111300 | :SETUP EMULATION OPERANDS & EMULATE INST |
| 7859 | 006620 | 122000 | .WORD | 122000 | :SETUP CC & REGS AND EXECUTE CIS INST. |
| 7860 | 006622 | 001300 | .WORD | 001300 | |
| 7861 | 006624 | 000000 | .WORD | 000000 | |
| 7862 | 006626 | 000170 | .WORD | 000170 | |
| 7863 | 006630 | 131370 | .WORD | 131370 | :COMPARE RESULTS |
| 7864 | 006632 | 001700 | .WORD | 001700 | |
| 7865 | 006634 | 140000 | .WORD | 140000 | :UPDATE PTRS FOR NEXT TEST CONDITION. |
| 7866 | | | | | : RETURN TO START EXECUTING NEXT TEST CONDITION. |

7868
7869
7870 006636
7871 006636
7872 006636 151500
7873 006640 010110
7874 006642 010201
7875 006644 010312
7876 006646 010504
7877 006650 020403
7878 006652 031001
7879 006654 031303
7880 006656 041001
7881 006660 041323
7882 006662 051300
7883 006664 060000
7884 006666 075123
7885 006670 074601
7886 006672 100000
7887 006674 111300
7888 006676 122000
7889 006700 001300
7890 006702 000000
7891 006704 007000
7892 006706 137000
7893 006710 140000
7894

XCMPN:
XCMPN:
.WORD 151500
.WORD 010110
.WORD 010201
.WORD 010312
.WORD 010504
.WORD 020403
.WORD 031001
.WORD 031303
.WORD 041001
.WORD 041323
.WORD 051300
.WORD 060000
.WORD 075123
.WORD 074601
.WORD 100000
.WORD 111300
.WORD 122000
.WORD 001300
.WORD 000000
.WORD 007000
.WORD 137000
.WORD 140000

:CMPP FLOW TABLE S1=SRC1, S2=SRC2
:CMPN FLOW TABLE S1=SRC1, S2=SRC2
:LOAD SPECIAL HANDLING FROM PTP15
:LOAD TR0 FROM PTP01
:LOAD TR1 FROM PTP02
:LOAD TR2 FROM PTP03
:LOAD TR4 FROM PTP05
:GENERATE TR3 FROM PTP04
:VERIFY THAT S1.ADR-S1.SURR.LEN >=20
:VERIFY THAT S2.ADR-S2.SURR.LEN >= 20
:VERIFY THAT S1.ADR+S1.LEN+S1.SURR.LEN < TBLN
:VERIFY THAT S2.ADR+S2.LEN+S2.SURR.LEN < TBLN
:ADD TEST BUFFER ADDRESS TO TR1 AND TR3
:INITIALIZE TEST BUFFER
:INSERT S2 & S2.SURR STRINGS IN TEST BUFFER
:INSERT S1 & S1.SURR STRINGS IN TEST BUFFER
:COPY TEST BUFFER INTO EMULATION BUFFER
:SETUP EMULATION OPERANDS & EMULATE INST
:SETUP CC & REGS AND EXECUTE CIS INST.

:COMPARE RESULTS
:UPDATE PTRS FOR NEXT TEST CONDITION.
: RETURN TO START EXECUTING NEXT TEST CONDITION.

7896
7897 006712
7898 006712
7899 006712 151500
7900 006714 010110
7901 006716 010201
7902 006720 010422
7903 006722 010304
7904 006724 020503
7905 006726 031001
7906 006730 031303
7907 006732 041001
7908 006734 041323
7909 006736 051300
7910 006740 060000
7911 006742 071123
7912 006744 001312
7913 006746 074601
7914 006750 100000
7915 006752 111300
7916 006754 122400
7917 006756 001300
7918 006760 000000
7919 006762 007000
7920 006764 133700
7921 006766 007000
7922 006770 140000
7923
7924

XASHP:
XASHN:

.WORD 151500
.WORD 010110
.WORD 010201
.WORD 010422
.WORD 010304
.WORD 020503
.WORD 031001
.WORD 031303
.WORD 041001
.WORD 041323
.WORD 051300
.WORD 060000
.WORD 071123
.WORD 001312
.WORD 074601
.WORD 100000
.WORD 111300
.WORD 122400
.WORD 001300
.WORD 000000
.WORD 007000
.WORD 133700
.WORD 007000
.WORD 140000

:ASHP FLOW TABLE
:ASHN FLOW TABLE
:LOAD SPECIAL HANDLING FROM PTP15
:LOAD TR0 FROM PTP01
:LOAD TR1 FROM PTP02
:LOAD TR2 FROM PTP04
:LOAD TR4 FROM PTP03
:GENERATE TR3 FROM PTP05
:VERIFY THAT SRC.ADR-SRC.SURR.LEN >= 20
:VERIFY THAT DST.ADR-DST.SURR.LEN >= 20
:VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN <TBLEN
:VERIFY THAT DST.ADR+DST.LEN+DST.SURR.LEN <TBLEN
:ADD TEST BUFFER ADDRESS TO TR1 & TR5
:INITIALIZE TEST BUFFER
:INSERT DST & DST.SURR STRINGS IN TEST BUFFER

:INSERT SRC IN TST BUFFER
:COPY TEST BUFFER INTO EMULATION BUFFER
:SETUP EMULATION OPERANDS & EXECUTE INST
:SETUP CC & REGS AND EXECUTE CIS INST

:COMPARE RESULTS

:UPDATE PTRS FOR NEXT TEST CONDITION
: AND RETURN TO START EXECUTING NEXT
: TEST CONDITION

| | | | | |
|------|--------|--------|--------|---|
| 7926 | | | | |
| 7927 | | | | |
| 7928 | | | | |
| 7929 | 006772 | | XLOCC: | :LOCC FLOW TABLE |
| 7930 | 006772 | | XSKPC: | :SKPC FLOW TABLE |
| 7931 | 006772 | 150700 | .WORD | :LOAD SPECIAL HANDLING WORD FROM PTP07 |
| 7932 | 006774 | 010100 | .WORD | :LOAD TR0 FROM PTP01 |
| 7933 | 006776 | 010201 | .WORD | :LOAD TR1 FROM PTP02 |
| 7934 | 007000 | 010304 | .WORD | :LOAD TR4 FROM PTP03 |
| 7935 | 007002 | 030601 | .WORD | :VERIFY THAT SRC.ADR-SRC.SURR.LEN >=20 |
| 7936 | 007004 | 040601 | .WORD | :VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN < TBLEN |
| 7937 | 007006 | 051000 | .WORD | :ADD TEST BUFFER ADDRESS TO TR1 |
| 7938 | 007010 | 060000 | .WORD | :INITIALIZE TEST BUFFER |
| 7939 | 007012 | 070401 | .WORD | :INSERT SRC & SRC SURR STRINGS IN BUFFER |
| 7940 | 007014 | 000605 | .WORD | |
| 7941 | 007016 | 100000 | .WORD | :COPY TEST BUFFER INTO EMULATION BUFFER |
| 7942 | 007020 | 111000 | .WORD | :SETUP EMULATION OPERANDS & EMULATE INST. |
| 7943 | 007022 | 121400 | .WORD | :SETUP CC & REGS AND EXECUTE CIS INST |
| 7944 | 007024 | 001000 | .WORD | |
| 7945 | 007026 | 000000 | .WORD | |
| 7946 | 007030 | 000170 | .WORD | |
| 7947 | 007032 | 131700 | .WORD | :COMPARE RESULTS |
| 7948 | 007034 | 001700 | .WORD | |
| 7949 | 007036 | 140000 | .WORD | :UPDATE POINTERS AND RETURN FOR NEXT |
| 7950 | | | | : TEST CONDITION. |
| 7951 | | | | |

| | | | | |
|------|--------|--------|--------------|---|
| 7953 | | | | |
| 7954 | 007040 | | XADDP: | :ADDP FLOW TABLE |
| 7955 | 007040 | | XADDN: | :ADDN FLOW TABLE |
| 7956 | 007040 | | XSUBP: | :SUBP FLOW TABLE |
| 7957 | 007040 | | XSUBN: | :SUBN FLOW TABLE |
| 7958 | 007040 | | XMULP: | :MULP FLOW TABLE |
| 7959 | 007040 | | XDIVP: | :DIVP FLOW TABLE |
| 7960 | 007040 | 152100 | .WORD 152100 | :LOAD SPECIAL HANDLING FROM PTP21 |
| 7961 | 007042 | 010110 | .WORD 010110 | :LOAD TR0 FROM PTP01 |
| 7962 | 007044 | 010201 | .WORD 010201 | :LOAD TR1 FROM PTP02 |
| 7963 | 007046 | 010322 | .WORD 010322 | :LOAD TR2 FROM PTP03 |
| 7964 | 007050 | 010534 | .WORD 010534 | :LOAD TR4 FROM PTP05 |
| 7965 | 007052 | 010605 | .WORD 010605 | :TYPE 0 USE ONLY - LOAD TR5 FROM PTP06 |
| 7966 | 007054 | 020403 | .WORD 020403 | :GENERATE TR3 & TR5 FROM PTP04 |
| 7967 | 007056 | 031101 | .WORD 031101 | :VERIFY THAT SRC1.ADR-SRC1.SURR.LEN >= 20 |
| 7968 | 007060 | 031403 | .WORD 031403 | :VERIFY THAT SRC2.ADR-SRC2.SURR.LEN >= 20 |
| 7969 | 007062 | 031705 | .WORD 031705 | :VERIFY THAT DST.ADR-DST.SURR.LEN >=20 |
| 7970 | 007064 | 041101 | .WORD 041101 | :VERIFY THAT SRC1.ADR+SRC1.LEN+SRC1.SURR.LEN < TBLN |
| 7971 | 007066 | 041423 | .WORD 041423 | :VERIFY THAT SRC2.ADR+SRC2.LEN+SRC2.SURR.LEN < TBLN |
| 7972 | 007070 | 041745 | .WORD 041745 | :VERIFY THAT DST.ADR+DST.LEN+DST.SURR.LEN < TBLN |
| 7973 | 007072 | 051350 | .WORD 051350 | :ADD TEST BUFFER ADDRESS TO TR1,TR3, & TR5 |
| 7974 | 007074 | 060000 | .WORD 060000 | :INITIALIZE TEST BUFFER |
| 7975 | 007076 | 071545 | .WORD 071545 | :INSERT DST & DST.SURR STRINGS IN TEST BUFFER |
| 7976 | 007100 | 001716 | .WORD 001716 | |
| 7977 | 007102 | 074701 | .WORD 074701 | :INSERT SRC1 IN TEST BUFFER |
| 7978 | 007104 | 075223 | .WORD 075223 | :INSERT SRC2 IN TEST BUFFFER |
| 7979 | 007106 | 100000 | .WORD 100000 | :COPY TEST BUFFER INTO EMULATION BUFFER |
| 7980 | 007110 | 111350 | .WORD 111350 | :SETUP EMULATION OPERANDS & EMULATE INST |
| 7981 | 007112 | 123000 | .WORD 123000 | :SETUP CC & REGS AND EXECUTE CIS INST. |
| 7982 | 007114 | 001350 | .WORD 001350 | |
| 7983 | 007116 | 000000 | .WORD 000000 | |
| 7984 | 007120 | 007000 | .WORD 007000 | |
| 7985 | 007122 | 135700 | .WORD 135700 | :COMPARE RESULTS |
| 7986 | 007124 | 007000 | .WORD 007000 | |
| 7987 | 007126 | 140000 | .WORD 140000 | :UPDATE POINTERS FOR NEXT TEST CONDITION |
| 7988 | | | | : AND RETURN TO START EXECUTING NEXT |
| 7989 | | | | : TEST CONDITION |

| | | | |
|------|--------|--------|--------------|
| 7991 | | | |
| 7992 | 007130 | | |
| 7993 | 007130 | | XSCANC: |
| 7994 | 007130 | 151500 | XSPANC: |
| 7995 | 007132 | 010100 | .WORD 151500 |
| 7996 | 007134 | 010201 | .WORD 010100 |
| 7997 | 007136 | 010302 | .WORD 010201 |
| 7998 | 007140 | 010303 | .WORD 010302 |
| 7999 | 007142 | 010404 | .WORD 010303 |
| 8000 | 007144 | 020505 | .WORD 010404 |
| 8001 | 007146 | 031001 | .WORD 020505 |
| 8002 | 007150 | 031305 | .WORD 031001 |
| 8003 | 007152 | 041001 | .WORD 031305 |
| 8004 | 007154 | 041325 | .WORD 041001 |
| 8005 | 007156 | 051500 | .WORD 041325 |
| 8006 | 007160 | 060000 | .WORD 051500 |
| 8007 | 007162 | 071125 | .WORD 060000 |
| 8008 | 007164 | 070601 | .WORD 071125 |
| 8009 | 007166 | 001007 | .WORD 070601 |
| 8010 | 007170 | 100000 | .WORD 001007 |
| 8011 | 007172 | 111500 | .WORD 100000 |
| 8012 | 007174 | 122000 | .WORD 111500 |
| 8013 | 007176 | 001500 | .WORD 122000 |
| 8014 | 007200 | 000000 | .WORD 001500 |
| 8015 | 007202 | 000170 | .WORD 000000 |
| 8016 | 007204 | 131570 | .WORD 000170 |
| 8017 | 007206 | 001700 | .WORD 131570 |
| 8018 | 007210 | 140000 | .WORD 001700 |
| 8019 | | | .WORD 140000 |

```
;SCANC FLOW TABLE
;SPANC FLOW TABLE
;LOAD SPECIAL HANDLING WORD FROM PTP15
;LOAD TR0 FROM PTP01
;LOAD TR1 FROM PTP02
;LOAD TR2 FROM PTP03
;LOAD TR3 FROM PTP03 (TR2 & TR3 CONTAIN TABLE LEN)
;LOAD TR4 FROM PTP04
;GENERATE TR5 FROM PTP05
;VERIFY THAT SRC.ADR - SRC.SURR.LEN > = 20.
;VERIFY THAT TABLE.ADR - TABLE.SURR.LEN > = 20
;VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN<TBLEN
;VERIFY THAT TABLE.ADR+256+TABLE SURR LEN<TBLEN
;ADD TEST BUFFER ADDRESS TO TR1 AND TR5
;INITIALIZE TEST BUFFER
;INSERT TABLE & TABLE SURR IN BUFFER
;INSERT SRC & SRC SURR STRINGS IN BUFFER

;COPY TEST BUFFER INTO EMULATION BUFFER
;SETUP EMULATION OPERANDS & EMULATE INST
;SETUP CC REGS & EXECUTE CIS INST

;COMPARE RESULTS

;UPDATE PTRS FROM NEXT TEST CONDITION
```

| | | | | |
|------|--------|--------|--------------|---|
| 8021 | | | | |
| 8022 | | | | |
| 8023 | 007212 | | XCVTPN: | :CVTPN FLOW TABLE |
| 8024 | 007212 | | XCVTNP: | :CVTNP FLOW TABLE |
| 8025 | 007212 | 151400 | .WORD 151400 | :LOAD SPECIAL HANDLING FROM PTP14 |
| 8026 | 007214 | 010110 | .WORD 010110 | :LOAD TR0 FROM PTP01 |
| 8027 | 007216 | 010201 | .WORD 010201 | :LOAD TR1 FROM PTP02 |
| 8028 | 007220 | 010322 | .WORD 010322 | :LOAD TR2 FROM PTP03 |
| 8029 | 007222 | 020403 | .WORD 020403 | :GENERATE TR3 FROM PTP04 |
| 8030 | 007224 | 030701 | .WORD 030701 | :VERIFY THAT SRC.ADR-SRC.SURR.LEN>=20 |
| 8031 | 007226 | 031203 | .WORD 031203 | :VERIFY THAT DST.ADR-DST.SURR.LEN>=20 |
| 8032 | 007230 | 040701 | .WORD 040701 | :VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN<TBLEN |
| 8033 | 007232 | 041223 | .WORD 041223 | :VERIFY THAT DST.ADR+DST.LEN+DST.SURR.LEN<TBLEN |
| 8034 | 007234 | 051300 | .WORD 051300 | :ADD TEST BUFFER ADDRESS TO TR1 & TR3 |
| 8035 | 007236 | 060000 | .WORD 060000 | :INITIALIZE TEST BUFFER |
| 8036 | 007240 | 071023 | .WORD 071023 | :INSERT DST & DST SURR STRINGS IN BUFFER |
| 8037 | 007242 | 001211 | .WORD 001211 | |
| 8038 | 007244 | 074501 | .WORD 074501 | :INSERT SRC STRING IN BUFFER |
| 8039 | 007246 | 100000 | .WORD 100000 | :COPY TEST BUFFER INTO EMULATION BUFFER |
| 8040 | 007250 | 111300 | .WORD 111300 | :SETUP EMULATION OPERANDS & EMULAT INST. |
| 8041 | 007252 | 122000 | .WORD 122000 | :SETUP CC REGS & EXECUTE CIS INST. |
| 8042 | 007254 | 001300 | .WORD 001300 | |
| 8043 | 007256 | 000000 | .WORD 000000 | |
| 8044 | 007260 | 007000 | .WORD 007000 | |
| 8045 | 007262 | 133700 | .WORD 133700 | :COMPARE RESULTS |
| 8046 | 007264 | 007000 | .WORD 007000 | |
| 8047 | 007266 | 140000 | .WORD 140000 | :UPDATE PTRS FROM NEXT TEST CONDITION |
| 8048 | | | | |

8050
8051
8052 007270
8053 007270
8054 007270 151000
8055 007272 010102
8056 007274 010203
8057 007276 010310
8058 007300 010401
8059 007302 030701
8060 007304 040701
~~8061 007306 051000~~
8062 007310 060000
8063 007312 070501
8064 007314 000706
8065 007316 100000
8066 007320 111000
8067 007322 122000
8068 007324 001300
8069 007326 000000
8070 007330 007000
8071 007332 131700
8072 007334 007000
8073 007336 140000
8074
8075
8076

XCVTLP:
XCVTLN:
.WORD 151000
.WORD 010102
.WORD 010203
.WORD 010310
.WORD 010401
.WORD 030701
.WORD 040701

~~.WORD 051000~~
.WORD 060000
.WORD 070501
.WORD 000706
.WORD 100000
.WORD 111000
.WORD 122000
.WORD 001300
.WORD 000000
.WORD 007000
.WORD 131700
.WORD 007000
.WORD 140000

;CVTLP FLOW TABLE
;CVTLN FLOW TABLE
;LOAD SPECIAL HANDLING FROM PTP10
;LOAD TR2 FROM PTP01
;LOAD TR3 FROM PTP02
;LOAD TR0 FROM PTP03
;LOAD TR1 FROM PTP04
;VERIFY THAT DST.ADR-DST.SURR.LEN>=20
;VERIFY THAT DST.ADR+DST.LEN+DST.SURR.LEN<TBLEN

;ADD TEST BUFFER ADDRESS TO TR1
;INITIALIZE TEST BUFFER
;INSERT DST & DST SURR STRINGS IN BUFFER

;COPY TEST BUFFER INTO EMULATION BUFFER
;SETUP EMULATION OPERANDS & EMULATE INST
;SETUP CC REGS & EXECUTE CIS INST

;COMPARE RESULTS

;UPDATE PTRS FOR NEXT TEST CONDITION

| | | | | | |
|------|--------|--------|---------|---|--|
| 8078 | | | | | |
| 8079 | | | | | |
| 8080 | 007340 | | XCVTPL: | :CVTPL FLOW TABLE | |
| 8081 | 007340 | | XCVTNL: | :CVTNL FLOW TABLE | |
| 8082 | 007340 | 150700 | .WORD | :LOAD SPECIAL HANDLING FROM PTP07 | |
| 8083 | 007342 | 010110 | .WORD | :LOAD TR0 FROM PTP01 | |
| 8084 | 007344 | 010201 | .WORD | :LOAD TR1 FROM PTP02 | |
| 8085 | 007346 | 010604 | .WORD | :LOAD TR4 FROM PTP06 | |
| 8086 | 007350 | 030501 | .WORD | :VERIFY THAT SRC.ADR-SRC.SURR.LEN.+20 | |
| 8087 | 007352 | 040501 | .WORD | :VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN<TBLEN | |
| 8088 | 007354 | 051000 | .WORD | :ADD TEST BUFFER ADDRESS TO TR1 | |
| 8089 | 007356 | 060000 | .WORD | :INITIALIZE TEST BUFFER | |
| 8090 | 007360 | 074301 | .WORD | :INSERT SRC STRING IN TEST BUFFER | |
| 8091 | 007362 | 100000 | .WORD | :COPY TEST BUFFER INTO EMULATION BUFFER | |
| 8092 | 007364 | 111000 | .WORD | :SETUP EMULATION OPERANDS & EMULATE INST. | |
| 8093 | 007366 | 122000 | .WORD | :SETUP CC REGS & EXECUTE CIS INST | |
| 8094 | 007370 | 001000 | .WORD | | |
| 8095 | 007372 | 007300 | .WORD | | |
| 8096 | 007374 | 007000 | .WORD | | |
| 8097 | 007376 | 137000 | .WORD | :COMPARE RESULTS | |
| 8098 | 007400 | 140000 | .WORD | :UPDATE PTRS FOR NEXT TEST CONDITION | |
| 8099 | | | | | |
| 8100 | | | | | |

8102 007402
8103 007402
8104 007402 010100
8105 007404 010201
8106 007406 010302
8107 007410 010403
8108 007412 010504
8109 007414 010605
8110 007416 060000
8111 007420 020700
8112 007422 100000
8113 007424 110000
8114 007426 120000
8115 007430 137000
8116 007432 140000

XL2D:
XL3D:

.WORD 010100
.WORD 010201
.WORD 010302
.WORD 010403
.WORD 010504
.WORD 010605
.WORD 060000
.WORD 020700
.WORD 100000
.WORD 110000
.WORD 120000
.WORD 137000
.WORD 140000

:L2DR FLOW TABLE
:L3DR FLOW TABLE
:LOAD TR0 FROM PTP01
:LOAD TR1 FROM PTP02
:LOAD TR2 FROM PTP03
:LOAD TR3 FROM PTP04
:LOAD TR4 FROM PTP05
:LOAD TR5 FROM PTP06
:INITIALIZE TEST BUFFER
:GENERATE TRN FROM PTP07
:COPY TEST BUFFER INTO EMUL. BUFFER
:SETUP EMUL. OPERANDS & EMUL INST.
:SETUP CC & REGS & EXECUTE CIS INST.
:COMPARE RESULTS
:UPDATE PTRS FOR NEXT TEST CONDITION.


```

8118      .SBTTL GLOBAL TEXT SECTION
8119
8120      ; FORMAT STATEMENTS USED IN PRINT CALLS
8121      :
8122      :
8123      :
8124      :
8124      007434 040445 046440 053117 AMOVC: .ASCIZ /%A MOVC/
           007442 000103
8125      007444 040445 046440 053117 AMOVRC: .ASCIZ /%A MOVRC/
           007452 041522 000
8126      007455 045 020101 047515 AMOVTC: .ASCIZ /%A MOVTC/
           007462 052126 000103
8127      007466 040445 046040 041517 ALOCC: .ASCIZ /%A LOCC/
           007474 000103
8128      007476 040445 051440 050113 ASKPC: .ASCIZ /%A SKPC/
           007504 000103
8129      007506 040445 051440 040503 ASCANC: .ASCIZ /%A SCANC/
           007514 041516 000
8130      007517 045 020101 050123 ASPANC: .ASCIZ /%A SPANC/
           007524 047101 000103
8131      007530 040445 041440 050115 ACMPC: .ASCIZ /%A CMPC/
           007536 000103
8132      007540 040445 046440 052101 AMATCHC: .ASCIZ /%A MATC/
           007546 000103
8133      007550 040445 040440 042104 AADDN: .ASCIZ /%A ADDN/
           007556 000116
8134      007560 040445 051440 041125 ASUBN: .ASCIZ /%A SUBN/
           007566 000116
8135      007570 040445 041440 050115 ACMPN: .ASCIZ /%A CMPN/
           007576 000116
8136      007600 040445 041440 052126 ACVTNL: .ASCIZ /%A CVTNL/
           007606 046116 000
8137      007611 045 020101 053103 ACVTPN: .ASCIZ /%A CVTPN/
           007616 050124 000116
8138      007622 040445 041440 052126 ACVTNP: .ASCIZ /%A CVTNP/
           007630 050116 000
8139      007633 045 020101 051501 AASHN: .ASCIZ /%A ASHN/
           007640 047110 000
8140      007643 045 020101 053103 ACVTLN: .ASCIZ /%A CVTLN/
           007650 046124 000116
8141      007654 040445 040440 042104 AADDP: .ASCIZ /%A ADDP/
           007662 000120
8142      007664 040445 051440 041125 ASUBP: .ASCIZ /%A SUBP/
           007672 000120
8143      007674 040445 041440 050115 ACMPP: .ASCIZ /%A CMPP/
           007702 000120
8144      007704 040445 041440 052126 ACVTPL: .ASCIZ /%A CVTPL/
           007712 046120 000
8145      007715 045 020101 052515 AMULP: .ASCIZ /%A MULP/
           007722 050114 000
8146      007725 045 020101 044504 ADIVP: .ASCIZ /%A DIVP/
           007732 050126 000
8147      007735 045 020101 051501 AASHP: .ASCIZ /%A ASHP/
           007742 050110 000

```

| | | | | | | |
|------|--------|--------|--------|--------|----------------|---|
| 8148 | 007745 | 045 | 020101 | 053103 | ACVTLP: .ASCIZ | /%A CVTLP/ |
| | 007752 | 046124 | 000120 | | | |
| 8149 | 007756 | 040445 | 046040 | 042062 | AL2D: .ASCIZ | /%A L2D%01/ |
| | 007764 | 047445 | 000061 | | | |
| 8150 | 007770 | 040445 | 046040 | 042063 | AL3D: .ASCIZ | /%A L3D%01/ |
| | 007776 | 047445 | 000061 | | | |
| 8151 | | | | | | |
| 8152 | 010002 | 051445 | 031462 | 040445 | FORM1: .ASCIZ | /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S5%AF%S24%ANZVC%N/ |
| | 010010 | 046123 | 051445 | 022465 | | |
| | 010016 | 051501 | 022501 | 032523 | | |
| | 010024 | 040445 | 046104 | 051445 | | |
| | 010032 | 022465 | 042101 | 022501 | | |
| | 010040 | 032523 | 040445 | 022506 | | |
| | 010046 | 031123 | 022464 | 047101 | | |
| | 010054 | 053132 | 022503 | 000116 | | |
| 8153 | 010062 | 051445 | 031462 | 040445 | FORM2: .ASCIZ | /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S5%AF%S6%AT%S15%ANZVC%N/ |
| | 010070 | 046123 | 051445 | 022465 | | |
| | 010076 | 051501 | 022501 | 032523 | | |
| | 010104 | 040445 | 046104 | 051445 | | |
| | 010112 | 022465 | 042101 | 022501 | | |
| | 010120 | 032523 | 040445 | 022506 | | |
| | 010126 | 033123 | 040445 | 022524 | | |
| | 010134 | 030523 | 022465 | 047101 | | |
| | 010142 | 053132 | 022503 | 000116 | | |
| 8154 | 010150 | 051445 | 031462 | 040445 | FORM3: .ASCIZ | /%S23%ASL%S5%ASA%S23%ACHAR%S21%ANZVC%N/ |
| | 010156 | 046123 | 051445 | 022465 | | |
| | 010164 | 051501 | 022501 | 031123 | | |
| | 010172 | 022463 | 041501 | 040510 | | |
| | 010200 | 022522 | 031123 | 022461 | | |
| | 010206 | 047101 | 053132 | 022503 | | |
| | 010214 | 000116 | | | | |
| 8155 | 010216 | 051445 | 031462 | 040445 | FORM4: .ASCIZ | /%S23%ASL%S5%ASA%S23%AMASK%S3%AT%S15%ANZVC%N/ |
| | 010224 | 046123 | 051445 | 022465 | | |
| | 010232 | 051501 | 022501 | 031123 | | |
| | 010240 | 022463 | 046501 | 051501 | | |
| | 010246 | 022513 | 031523 | 040445 | | |
| | 010254 | 022524 | 030523 | 022465 | | |
| | 010262 | 047101 | 053132 | 022503 | | |
| | 010270 | 000116 | | | | |
| 8156 | 010272 | 051445 | 031462 | 040445 | FORM5: .ASCIZ | /%S23%AS1L%S4%AS1A%S4%AS2L%S4%AS2A%S4%AF%S24%ANZVC%N/ |
| | 010300 | 030523 | 022514 | 032123 | | |
| | 010306 | 040445 | 030523 | 022501 | | |
| | 010314 | 032123 | 040445 | 031123 | | |
| | 010322 | 022514 | 032123 | 040445 | | |
| | 010330 | 031123 | 022501 | 032123 | | |
| | 010336 | 040445 | 022506 | 031123 | | |
| | 010344 | 022464 | 047101 | 053132 | | |
| | 010352 | 022503 | 000116 | | | |
| 8157 | 010356 | 051445 | 031462 | 040445 | FORM6: .ASCIZ | /%S23%ASL%S5%ASA%S5%AOL%S5%A0A%S32%ANZVC%N/ |
| | 010364 | 046123 | 051445 | 022465 | | |
| | 010372 | 051501 | 022501 | 032523 | | |
| | 010400 | 040445 | 046117 | 051445 | | |
| | 010406 | 022465 | 047501 | 022501 | | |
| | 010414 | 031523 | 022462 | 047101 | | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| 8158 | 010422 | 053132 | 022503 | 000116 | |
| | 010430 | 051445 | 031462 | 040445 | FORM7: .ASCIZ /%S23%AS1L%S4%AS1A%S4%AS2L%S4%AS2A%S4%ADL%S5%ADA%S14%ANZVC%/ |
| | 010436 | 030523 | 022514 | 032123 | |
| | 010444 | 040445 | 030523 | 022501 | |
| | 010452 | 032123 | 040445 | 031123 | |
| | 010460 | 022514 | 032123 | 040445 | |
| | 010466 | 031123 | 022501 | 032123 | |
| | 010474 | 040445 | 046104 | 051445 | |
| | 010502 | 022465 | 042101 | 022501 | |
| | 010510 | 030523 | 022464 | 047101 | |
| 8159 | 010516 | 053132 | 022503 | 000116 | |
| | 010524 | 051445 | 031462 | 040445 | FORM8: .ASCIZ /%S23%AS1L%S4%AS1A%S4%AS2L%S4%AS2A%S32%ANZVC%/ |
| | 010532 | 030523 | 022514 | 032123 | |
| | 010540 | 040445 | 030523 | 022501 | |
| | 010546 | 032123 | 040445 | 031123 | |
| | 010554 | 022514 | 032123 | 040445 | |
| | 010562 | 031123 | 022501 | 031523 | |
| | 010570 | 022462 | 047101 | 053132 | |
| | 010576 | 022503 | 000116 | | |
| 8160 | 010602 | 051445 | 031462 | 040445 | FORM9: .ASCIZ /%S23%ASL%S5%ASA%S5%AD.H%S4%AD.L%S32%ANZVC%/ |
| | 010610 | 046123 | 051445 | 022465 | |
| | 010616 | 051501 | 022501 | 032523 | |
| | 010624 | 040445 | 027104 | 022510 | |
| | 010632 | 032123 | 040445 | 027104 | |
| | 010640 | 022514 | 031523 | 022462 | |
| | 010646 | 047101 | 053132 | 022503 | |
| | 010654 | 000116 | | | |
| 8161 | 010656 | 051445 | 031462 | 040445 | FORM10: .ASCIZ /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S32%ANZVC%/ |
| | 010664 | 046123 | 051445 | 022465 | |
| | 010672 | 051501 | 022501 | 032523 | |
| | 010700 | 040445 | 046104 | 051445 | |
| | 010706 | 022465 | 042101 | 022501 | |
| | 010714 | 031523 | 022462 | 047101 | |
| | 010722 | 053132 | 022503 | 000116 | |
| 8162 | 010730 | 051445 | 031462 | 040445 | FORM11: .ASCIZ /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S5%AR,S%S22%ANZVC%/ |
| | 010736 | 046123 | 051445 | 022465 | |
| | 010744 | 051501 | 022501 | 032523 | |
| | 010752 | 040445 | 046104 | 051445 | |
| | 010760 | 022465 | 042101 | 022501 | |
| | 010766 | 032523 | 040445 | 026122 | |
| | 010774 | 022523 | 031123 | 022462 | |
| | 011002 | 047101 | 053132 | 022503 | |
| | 011010 | 000116 | | | |
| 8163 | 011012 | 051445 | 031462 | 040445 | FORM12: .ASCIZ /%S23%ADL%S5%ADA%S5%AS.H%S4%AS.L%S32%ANZVC%/ |
| | 011020 | 046104 | 051445 | 022465 | |
| | 011026 | 042101 | 022501 | 032523 | |
| | 011034 | 040445 | 027123 | 022510 | |
| | 011042 | 032123 | 040445 | 027123 | |
| | 011050 | 022514 | 031523 | 022462 | |
| | 011056 | 047101 | 053132 | 022503 | |
| | 011064 | 000116 | | | |
| 8164 | 011066 | 040445 | 044440 | 050116 | INREG: .ASCIZ +%A INPUT R0-R6,CC/ + |
| | 011074 | 052125 | 020040 | 051040 | |
| | 011102 | 026460 | 033122 | 041454 | |

| | | | | | | |
|------|--------|--------|--------|--------|----------------|--|
| 8165 | 011110 | 027503 | 000040 | | | |
| | 011114 | 040445 | 044440 | 050116 | INMEM: .ASCIZ | +%A INPUTS IN MEMORY/ + |
| | 011122 | 052125 | 020123 | 047111 | | |
| | 011130 | 046440 | 046505 | 051117 | | |
| | 011136 | 027531 | 000040 | | | |
| 8166 | 011142 | 040445 | 042440 | 050130 | EMOUT: .ASCIZ | +%A EXP OUT R0-R6,CC/ + |
| | 011150 | 047440 | 052125 | 051040 | | |
| | 011156 | 026460 | 033122 | 041454 | | |
| | 011164 | 027503 | 000040 | | | |
| 8167 | 011170 | 047445 | 022466 | 030523 | FORM13: .ASCIZ | /%06%S1%06%S1%06%S1%06%S1%06%S1%06%S1%Y4%N/ |
| | 011176 | 047445 | 022466 | 030523 | | |
| | 011204 | 047445 | 022466 | 030523 | | |
| | 011212 | 047445 | 022466 | 030523 | | |
| | 011220 | 047445 | 022466 | 030523 | | |
| | 011226 | 047445 | 022466 | 030523 | | |
| | 011234 | 047445 | 022466 | 030523 | | |
| | 011242 | 054445 | 022464 | 000116 | | |
| 8168 | 011250 | 047445 | 022466 | 030523 | FORM14: .ASCIZ | /%06%S1%06%S1%06%S1%06%S1%06%S1%06%S1%Y4%N/ |
| | 011256 | 047445 | 022466 | 030523 | | |
| | 011264 | 047445 | 022466 | 030523 | | |
| | 011272 | 047445 | 022466 | 030523 | | |
| | 011300 | 047445 | 022466 | 030523 | | |
| | 011306 | 047445 | 022466 | 030523 | | |
| | 011314 | 047445 | 022466 | 030523 | | |
| | 011322 | 054445 | 022464 | 000116 | | |
| 8169 | 011330 | 040445 | 040440 | 052103 | ACOUT: .ASCIZ | +%A ACT OUT R0-R6,CC/ + |
| | 011336 | 047440 | 052125 | 051040 | | |
| | 011344 | 026460 | 033122 | 041454 | | |
| | 011352 | 027503 | 000040 | | | |
| 8170 | 011356 | 047445 | 022466 | 030523 | FORM15: .ASCIZ | /%06%S1/ |
| | 011364 | 000 | | | | |
| 8171 | 011365 | 045 | 033523 | 000 | FORM16: .ASCIZ | /%S7/ |
| 8172 | 011371 | 045 | 033123 | 022461 | FORM17: .ASCIZ | /%S61%Y4/ |
| | 011376 | 032131 | 000 | | | |
| 8173 | 011401 | 045 | 032131 | 000 | FORM18: .ASCIZ | /%Y4/ |
| 8174 | 011405 | 045 | 022516 | 020101 | EBUFO: .ASCIZ | +%N%A EXP BUFFER %06%A/ %03%N+ |
| | 011412 | 054105 | 020120 | 052502 | | |
| | 011420 | 043106 | 051105 | 022440 | | |
| | 011426 | 033117 | 040445 | 020057 | | |
| | 011434 | 047445 | 022463 | 000116 | | |
| 8175 | 011442 | 040445 | 040440 | 052103 | ABUFO: .ASCIZ | +%A ACT BUFFER %06%A/ %03+ |
| | 011450 | 041040 | 043125 | 042506 | | |
| | 011456 | 020122 | 047445 | 022466 | | |
| | 011464 | 027501 | 022440 | 031517 | | |
| | 011472 | 000 | | | | |
| 8176 | 011473 | 015 | 041412 | 041475 | QDISP: .ASCIZ | <(R)<LF>/C=CONT.;R=REPEAT TEST;S=RESTART;D=DISPLAY MEMORY;H=REPEAT & HAL |
| | 011500 | 047117 | 027124 | 051073 | | |
| | 011506 | 051075 | 050105 | 040505 | | |
| | 011514 | 020124 | 042524 | 052123 | | |
| | 011522 | 051473 | 051075 | 051505 | | |
| | 011530 | 040524 | 052122 | 042073 | | |
| | 011536 | 042075 | 051511 | 046120 | | |
| | 011544 | 054501 | 046440 | 046505 | | |
| | 011552 | 051117 | 035531 | 036510 | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| | 011560 | 042522 | 042520 | 052101 | |
| | 011566 | 023040 | 044040 | 046101 | |
| | 011574 | 020124 | 052101 | 041440 | |
| | 011602 | 051511 | 044440 | 051516 | |
| | 011610 | 037524 | 000 | | |
| 8177 | 011613 | 045 | 022516 | 041501 | STKM1: .ASCIZ /%N%ACIS INST EXECUTION USED MORE THAN 64 LOCS ON STACK%N/ |
| | 011620 | 051511 | 044440 | 051516 | |
| | 011626 | 020124 | 054105 | 041505 | |
| | 011634 | 052125 | 047511 | 020116 | |
| | 011642 | 051525 | 042105 | 046440 | |
| | 011650 | 051117 | 020105 | 044124 | |
| | 011656 | 047101 | 033040 | 020064 | |
| | 011664 | 047514 | 051503 | 047440 | |
| | 011672 | 020116 | 052123 | 041501 | |
| | 011700 | 022513 | 000116 | | |
| 8178 | 011704 | 047045 | 040445 | 044503 | STKM2: .ASCIZ /%N%ACIS INST EXECUTION DESTROYED CONTENTS OF WORD AT STACK+2%N/ |
| | 011712 | 020123 | 047111 | 052123 | |
| | 011720 | 042440 | 042530 | 052503 | |
| | 011726 | 044524 | 047117 | 042040 | |
| | 011734 | 051505 | 051124 | 054517 | |
| | 011742 | 042105 | 041440 | 047117 | |
| | 011750 | 042524 | 052116 | 020123 | |
| | 011756 | 043117 | 053440 | 051117 | |
| | 011764 | 020104 | 052101 | 051440 | |
| | 011772 | 040524 | 045503 | 031053 | |
| | 012000 | 047045 | 000 | | |
| 8179 | 012003 | 045 | 022516 | 052101 | CISQ: .ASCIZ /%N%ATRAP TO 10 OCCURRED ON CIS INST IN TEST #1. IS CISP PRESENT?%N/ |
| | 012010 | 040522 | 020120 | 047524 | |
| | 012016 | 030440 | 020060 | 041517 | |
| | 012024 | 052503 | 051122 | 042105 | |
| | 012032 | 047440 | 020116 | 044503 | |
| | 012040 | 020123 | 047111 | 052123 | |
| | 012046 | 044440 | 020116 | 042524 | |
| | 012054 | 052123 | 021440 | 027061 | |
| | 012062 | 044440 | 020123 | 044503 | |
| | 012070 | 050123 | 050040 | 042522 | |
| | 012076 | 042523 | 052116 | 022477 | |
| | 012104 | 000116 | | | |
| 8180 | 012106 | 047045 | 040445 | 051124 | MMVMSG: .ASCIZ /%N%ATRAP TO 250/ |
| | 012114 | 050101 | 052040 | 020117 | |
| | 012122 | 032462 | 000060 | | |
| 8181 | 012126 | 047045 | 040445 | 051124 | TRAP4: .ASCIZ /%N%ATRAP TO 4/ |
| | 012134 | 050101 | 052040 | 020117 | |
| | 012142 | 000064 | | | |
| 8182 | 012144 | 047045 | 040445 | 051124 | TRAP10: .ASCIZ /%N%ATRAP TO 10/ |
| | 012152 | 050101 | 052040 | 020117 | |
| | 012160 | 030061 | 000 | | |
| 8183 | 012163 | 045 | 022516 | 042501 | HLTMSG: .ASCIZ /%N%AERROR HALT/ |
| | 012170 | 051122 | 051117 | 044040 | |
| | 012176 | 046101 | 000124 | | |
| 8184 | 012202 | 047045 | 040445 | 044503 | NOPROG: .ASCIZ /%N%ACIS INST IS NOT MAKING PROGRESS%N/ |
| | 012210 | 020123 | 047111 | 052123 | |
| | 012216 | 044440 | 020123 | 047516 | |
| | 012224 | 020124 | 040515 | 044513 | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| | 012232 | 043516 | 050040 | 047522 | |
| | 012240 | 051107 | 051505 | 022523 | |
| | 012246 | 000116 | | | |
| 8185 | 012250 | 047045 | 040445 | 040514 | LATEXC: .ASCIZ /%N%ALATENCY EXCESSIVE%N/ |
| | 012256 | 042524 | 041516 | 020131 | |
| | 012264 | 054105 | 042503 | 051523 | |
| | 012272 | 053111 | 022505 | 000116 | |
| 8186 | 012300 | 040445 | 020040 | 052101 | TRPINF: .ASCIZ /%A AT:%06%A MODE:%01%A D-EN:%B1%A INST:%06%A INST CT:%D2%S1%D5%N/ |
| | 012306 | 022472 | 033117 | 040445 | |
| | 012314 | 020040 | 047515 | 042504 | |
| | 012322 | 022472 | 030517 | 040445 | |
| | 012330 | 020040 | 026504 | 047105 | |
| | 012336 | 022472 | 030502 | 040445 | |
| | 012344 | 020040 | 047111 | 052123 | |
| | 012352 | 022472 | 033117 | 040445 | |
| | 012360 | 020040 | 047111 | 052123 | |
| | 012366 | 041440 | 035124 | 042045 | |
| | 012374 | 022462 | 030523 | 042045 | |
| | 012402 | 022465 | 000116 | | |
| 8187 | 012406 | 047045 | 040445 | 047105 | ASK: .ASCIZ /%N%AENTER INSTRUCTION TO TEST <ALL> / |
| | 012414 | 042524 | 020122 | 047111 | |
| | 012422 | 052123 | 052522 | 052103 | |
| | 012430 | 047511 | 020116 | 047524 | |
| | 012436 | 052040 | 051505 | 020124 | |
| | 012444 | 040474 | 046114 | 020076 | |
| | 012452 | 020040 | 000 | | |
| 8188 | 012455 | 045 | 022516 | 051101 | ASKRM: .ASCIZ /%N%ARANDOM EXERCISE MODE (Y OR N) ? / |
| | 012462 | 047101 | 047504 | 020115 | |
| | 012470 | 054105 | 051105 | 044503 | |
| | 012476 | 042523 | 046440 | 042117 | |
| | 012504 | 020105 | 054450 | 047440 | |
| | 012512 | 020122 | 024516 | 037440 | |
| | 012520 | 020040 | 000 | | |
| 8189 | 012523 | 045 | 022516 | 050101 | ASKMOD: .ASCIZ /%N%APROCESSOR TEST MODE (K=KERNEL,S=SUPV,U=USER)? / |
| | 012530 | 047522 | 042503 | 051523 | |
| | 012536 | 051117 | 052040 | 051505 | |
| | 012544 | 020124 | 047515 | 042504 | |
| | 012552 | 024040 | 036513 | 042513 | |
| | 012560 | 047122 | 046105 | 051454 | |
| | 012566 | 051475 | 050125 | 026126 | |
| | 012574 | 036525 | 051525 | 051105 | |
| | 012602 | 037451 | 020040 | 000 | |
| 8190 | 012607 | 045 | 022516 | 046501 | ASKMM: .ASCIZ /%N%AMEMORY MANAGEMENT (N=OFF,D=D-SPACE ENABLED,H=D-SPACE DISABLED)? / |
| | 012614 | 046505 | 051117 | 020131 | |
| | 012622 | 040515 | 040516 | 042507 | |
| | 012630 | 042515 | 052116 | 024040 | |
| | 012636 | 036516 | 043117 | 026106 | |
| | 012644 | 036504 | 026504 | 050123 | |
| | 012652 | 041501 | 020105 | 047105 | |
| | 012660 | 041101 | 042514 | 026104 | |
| | 012666 | 036510 | 026504 | 050123 | |
| | 012674 | 041501 | 020105 | 044504 | |
| | 012702 | 040523 | 046102 | 042105 | |
| | 012710 | 037451 | 020040 | 000 | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 8191 | 012715 | 045 | 022516 | 052101 | ASKINT: .ASCIZ /%N%ATEST INTERRUPTABILITY OF CIS INSTRUCTIONS (KW11-P REQUIRED) (Y OR N |
| | 012722 | 051505 | 020124 | 047111 | |
| | 012730 | 042524 | 051122 | 050125 | |
| | 012736 | 040524 | 044502 | 044514 | |
| | 012744 | 054524 | 047440 | 020106 | |
| | 012752 | 044503 | 020123 | 047111 | |
| | 012760 | 052123 | 052522 | 052103 | |
| | 012766 | 047511 | 051516 | 024040 | |
| | 012774 | 053513 | 030461 | 050055 | |
| | 013002 | 051040 | 050505 | 044525 | |
| | 013010 | 042522 | 024504 | 024040 | |
| | 013016 | 020131 | 051117 | 047040 | |
| | 013024 | 020051 | 000077 | | |
| 8192 | 013030 | 047045 | 040445 | 047111 | ASKSRC: .ASCIZ /%N%AINTR SOURCE (R=LTC,N=KW11-P @100KHZ,C=KW11-P @10KHZ,Y=KW11-P EXT OS |
| | 013036 | 051124 | 051440 | 052517 | |
| | 013044 | 041522 | 020105 | 051050 | |
| | 013052 | 046075 | 041524 | 047054 | |
| | 013060 | 045475 | 030527 | 026461 | |
| | 013066 | 020120 | 030500 | 030060 | |
| | 013074 | 044113 | 026132 | 036503 | |
| | 013102 | 053513 | 030461 | 050055 | |
| | 013110 | 040040 | 030061 | 044113 | |
| | 013116 | 026132 | 036531 | 053513 | |
| | 013124 | 030461 | 050055 | 042440 | |
| | 013132 | 052130 | 047440 | 041523 | |
| | 013140 | 037451 | 000 | | |
| 8193 | 013143 | 045 | 022516 | 040501 | ASKDI: .ASCIZ /%N%AALLOW INTERRUPTING THE CIS INST EXECUTED DURING NORMAL INTR SERVICE |
| | 013150 | 046114 | 053517 | 044440 | |
| | 013156 | 052116 | 051105 | 052522 | |
| | 013164 | 052120 | 047111 | 020107 | |
| | 013172 | 044124 | 020105 | 044503 | |
| | 013200 | 020123 | 047111 | 052123 | |
| | 013206 | 042440 | 042530 | 052503 | |
| | 013214 | 042524 | 020104 | 052504 | |
| | 013222 | 044522 | 043516 | 047040 | |
| | 013230 | 051117 | 040515 | 020114 | |
| | 013236 | 047111 | 051124 | 051440 | |
| | 013244 | 051105 | 044526 | 042503 | |
| | 013252 | 000077 | | | |
| 8194 | 013254 | 047045 | 040445 | 040503 | NOLAT: .ASCIZ /%N%ACAN'T TEST LATENCY - NEED 2ND KW11-P/ |
| | 013262 | 023516 | 020124 | 042524 | |
| | 013270 | 052123 | 046040 | 052101 | |
| | 013276 | 047105 | 054503 | 026440 | |
| | 013304 | 047040 | 042505 | 020104 | |
| | 013312 | 047062 | 020104 | 053513 | |
| | 013320 | 030461 | 050055 | 000 | |
| 8195 | 013325 | 045 | 022516 | 041501 | NOINT: .ASCIZ /%N%ACAN'T TEST INTERRUPTABILITY - NO CLOCK/ |
| | 013332 | 047101 | 052047 | 052040 | |
| | 013340 | 051505 | 020124 | 047111 | |
| | 013346 | 042524 | 051122 | 050125 | |
| | 013354 | 040524 | 044502 | 044514 | |
| | 013362 | 054524 | 026440 | 047040 | |
| | 013370 | 020117 | 046103 | 041517 | |
| | 013376 | 000113 | | | |

| | | | | | | | | | | | | | | | | |
|------|--------|--------|--------|--------|---------|--------|--|---|---|---|---|---|---|---|---|----|
| 8196 | 013400 | 047045 | 040445 | 053523 | SWNG: | .ASCIZ | /N%ASWITCH ON CIS MODULE ACTS LIKE IT IS IN THE INCORRECT POSITION%/ | | | | | | | | | |
| | 013406 | 052111 | 044103 | 047440 | | | | | | | | | | | | |
| | 013414 | 020116 | 044503 | 020123 | | | | | | | | | | | | |
| | 013422 | 047515 | 052504 | 042514 | | | | | | | | | | | | |
| | 013430 | 040440 | 052103 | 020123 | | | | | | | | | | | | |
| | 013436 | 044514 | 042513 | 044440 | | | | | | | | | | | | |
| | 013444 | 020124 | 051511 | 044440 | | | | | | | | | | | | |
| | 013452 | 020116 | 044124 | 020105 | | | | | | | | | | | | |
| | 013460 | 047111 | 047503 | 051122 | | | | | | | | | | | | |
| | 013466 | 041505 | 020124 | 047520 | | | | | | | | | | | | |
| | 013474 | 044523 | 044524 | 047117 | | | | | | | | | | | | |
| | 013502 | 047045 | 000 | | | | | | | | | | | | | |
| 8197 | 013505 | 045 | 022516 | 041501 | NOABO: | .ASCIZ | /N%ACIS INST FAILED TO ABORT%/ | | | | | | | | | |
| | 013512 | 051511 | 044440 | 051516 | | | | | | | | | | | | |
| | 013520 | 020124 | 040506 | 046111 | | | | | | | | | | | | |
| | 013526 | 042105 | 052040 | 020117 | | | | | | | | | | | | |
| | 013534 | 041101 | 051117 | 022524 | | | | | | | | | | | | |
| | 013542 | 000116 | | | | | | | | | | | | | | |
| 8198 | 013544 | 047045 | 040445 | 044514 | KW11L: | .ASCIZ | /N%ALINE CLOCK WILL BE USED FOR INTERRUPT SOURCE%/ | | | | | | | | | |
| | 013552 | 042516 | 041440 | 047514 | | | | | | | | | | | | |
| | 013560 | 045503 | 053440 | 046111 | | | | | | | | | | | | |
| | 013566 | 020114 | 042502 | 052440 | | | | | | | | | | | | |
| | 013574 | 042523 | 020104 | 047506 | | | | | | | | | | | | |
| | 013602 | 020122 | 047111 | 042524 | | | | | | | | | | | | |
| | 013610 | 051122 | 050125 | 020124 | | | | | | | | | | | | |
| | 013616 | 047523 | 051125 | 042503 | | | | | | | | | | | | |
| | 013624 | 047045 | 000 | | | | | | | | | | | | | |
| 8199 | 013627 | 015 | 040412 | 042104 | AST: | .ASCIZ | <CR><LF>/ADDR(S)?/ | | | | | | | | | |
| | 013634 | 024122 | 024523 | 000077 | | | | | | | | | | | | |
| 8200 | 013642 | 047445 | 022466 | 027501 | FORM19: | .ASCIZ | +%06%A/%S3%03%S2%03%S2%03%S2%03+ | | | | | | | | | |
| | 013650 | 051445 | 022463 | 031517 | | | | | | | | | | | | |
| | 013656 | 051445 | 022462 | 031517 | | | | | | | | | | | | |
| | 013664 | 051445 | 022462 | 031517 | | | | | | | | | | | | |
| | 013672 | 051445 | 022462 | 031517 | | | | | | | | | | | | |
| | 013700 | 000 | | | | | | | | | | | | | | |
| 8201 | 013701 | 045 | 020101 | 020040 | ADDHDR: | .ASCIZ | /A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | N/ |
| | 013706 | 020040 | 020040 | 020040 | | | | | | | | | | | | |
| | 013714 | 020040 | 020060 | 020040 | | | | | | | | | | | | |
| | 013722 | 030440 | 020040 | 020040 | | | | | | | | | | | | |
| | 013730 | 020062 | 020040 | 031440 | | | | | | | | | | | | |
| | 013736 | 020040 | 020040 | 020064 | | | | | | | | | | | | |
| | 013744 | 020040 | 032440 | 020040 | | | | | | | | | | | | |
| | 013752 | 020040 | 020066 | 020040 | | | | | | | | | | | | |
| | 013760 | 033440 | 047045 | 000 | | | | | | | | | | | | |
| 8202 | 013765 | 045 | 031123 | 047445 | FORM20: | .ASCIZ | /S2%03%S2%03%S2%03%S2%03%/ | | | | | | | | | |
| | 013772 | 022463 | 031123 | 047445 | | | | | | | | | | | | |
| | 014000 | 022463 | 031123 | 047445 | | | | | | | | | | | | |
| | 014006 | 022463 | 031123 | 047445 | | | | | | | | | | | | |
| | 014014 | 022463 | 000116 | | | | | | | | | | | | | |
| 8203 | 014020 | 047045 | 000 | | FORM21: | .ASCIZ | /N/ | | | | | | | | | |
| 8204 | 014023 | 045 | 022516 | 020101 | FORM22: | .ASCIZ | /N%A SRC%S13/ | | | | | | | | | |
| | 014030 | 051123 | 022503 | 030523 | | | | | | | | | | | | |
| | 014036 | 000063 | | | | | | | | | | | | | | |
| 8205 | 014040 | 047045 | 040445 | 051440 | FORM23: | .ASCIZ | /N%A SRC1%S12/ | | | | | | | | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| | 014046 | 041522 | 022461 | 030523 | |
| | 014054 | 000062 | | | |
| 8206 | 014056 | 047045 | 040445 | 051440 | FORM24: .ASCIZ /%N%A SRC2%S12/ |
| | 014064 | 041522 | 022462 | 030523 | |
| | 014072 | 000062 | | | |
| 8207 | 014074 | 047045 | 040445 | 042440 | FORM25: .ASCIZ /%N%A EXP RESULT%S4/ |
| | 014102 | 050130 | 051040 | 051505 | |
| | 014110 | 046125 | 022524 | 032123 | |
| | 014116 | 000 | | | |
| 8208 | 014117 | 045 | 022516 | 020101 | FORM26: .ASCIZ /%N%A ACT RESULT%S4/ |
| | 014124 | 041501 | 020124 | 042522 | |
| | 014132 | 052523 | 052114 | 051445 | |
| | 014140 | 000064 | | | |
| 8209 | 014142 | 040445 | 000040 | | FORM27: .ASCIZ /%A / |
| 8210 | 014146 | 051445 | 022462 | 044501 | FORM30: .ASCIZ /%S2%A INST CNT:%D2%S1%D5/ |
| | 014154 | 051516 | 020124 | 047103 | |
| | 014162 | 035124 | 042045 | 022462 | |
| | 014170 | 030523 | 042045 | 000065 | |
| 8211 | 014176 | 040445 | 000111 | | FORM31: .ASCIZ /%AI/ |
| 8212 | 014202 | 040445 | 020040 | 044523 | FORM32: .ASCIZ /%A SIGN BYTE=(/ |
| | 014210 | 047107 | 041040 | 052131 | |
| | 014216 | 036505 | 000050 | | |
| 8213 | 014222 | 040445 | 000051 | | FORM33: .ASCIZ /%A)/ |
| 8214 | 014226 | 040445 | 025440 | 000 | FORM34: .ASCIZ /%A +/ |
| 8215 | 014233 | 045 | 020101 | 000055 | FORM35: .ASCIZ /%A -/ |
| 8216 | 014240 | 047045 | 047045 | 040445 | FORM36: .ASCIZ /%N%N%RANDOM # GENERATOR SEED%S5%06%S2%06%S2%06%N/ |
| | 014246 | 040522 | 042116 | 046517 | |
| | 014254 | 021440 | 043440 | 047105 | |
| | 014262 | 051105 | 052101 | 051117 | |
| | 014270 | 051440 | 042505 | 022504 | |
| | 014276 | 032523 | 047445 | 022466 | |
| | 014304 | 031123 | 047445 | 022466 | |
| | 014312 | 031123 | 047445 | 022466 | |
| | 014320 | 000116 | | | |
| 8217 | 014322 | 047045 | 040445 | 051105 | FORM37: .ASCIZ /%N%A ERROR #D6/ |
| | 014330 | 047522 | 020122 | 022443 | |
| | 014336 | 033104 | 000 | | |
| 8218 | 014341 | 045 | 022516 | 042501 | FORM38: .ASCIZ /%N%A ERROR IN UNUSED REGISTER SET: USED SET:%01/ |
| | 014346 | 051122 | 051117 | 044440 | |
| | 014354 | 020116 | 047125 | 051525 | |
| | 014362 | 042105 | 051040 | 043505 | |
| | 014370 | 051511 | 042524 | 020122 | |
| | 014376 | 042523 | 035124 | 052440 | |
| | 014404 | 042523 | 020104 | 042523 | |
| | 014412 | 035124 | 047445 | 000061 | |
| 8219 | 014420 | 047045 | 040445 | 054105 | FORM39: .ASCIZ /%N%A EXP:%06%A ACT R0-R5 %06%S1%06%S1%06%S1%06%S1%06%S1%06/ |
| | 014426 | 035120 | 047445 | 022466 | |
| | 014434 | 020101 | 040440 | 052103 | |
| | 014442 | 051040 | 026460 | 032522 | |
| | 014450 | 022440 | 033117 | 051445 | |
| | 014456 | 022461 | 033117 | 051445 | |
| | 014464 | 022461 | 033117 | 051445 | |
| | 014472 | 022461 | 033117 | 051445 | |
| | 014500 | 022461 | 033117 | 051445 | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| 8220 | 014506 | 022461 | 033117 | 000 | |
| | 014513 | 045 | 020101 | 047111 | FORM40: .ASCIZ /%A INTR CNT:%D4%A REG SET:%01%A MODE:/' |
| | 014520 | 051124 | 041440 | 052116 | |
| | 014526 | 022472 | 032104 | 040445 | |
| | 014534 | 051040 | 043505 | 051440 | |
| | 014542 | 052105 | 022472 | 030517 | |
| | 014550 | 040445 | 046440 | 042117 | |
| | 014556 | 035105 | 000 | | |
| 8221 | 014561 | 045 | 020101 | 042040 | FRM40A: .ASCIZ /%A D-EN:/' |
| | 014566 | 042455 | 035116 | 000 | |
| 8222 | 014573 | 045 | 031123 | 022463 | FORM41: .ASCIZ /%S23%AR0%S5%AR1%S5%AR2%S5%AR3%S5%AR4%S5%AR5%S5%AR6%S5%ANZVC%N/' |
| | 014600 | 051101 | 022460 | 032523 | |
| | 014606 | 040445 | 030522 | 051445 | |
| | 014614 | 022465 | 051101 | 022462 | |
| | 014622 | 032523 | 040445 | 031522 | |
| | 014630 | 051445 | 022465 | 051101 | |
| | 014636 | 022464 | 032523 | 040445 | |
| | 014644 | 032522 | 051445 | 022465 | |
| | 014652 | 051101 | 022466 | 032523 | |
| | 014660 | 040445 | 055116 | 041526 | |
| | 014666 | 047045 | 000 | | |
| 8223 | 014671 | 045 | 022516 | 041501 | FORM42: .ASCIZ /%N%ACIS INSTRUCTION WAS SUSPENDED TO SERVICE INTERRUPT/' |
| | 014676 | 051511 | 044440 | 051516 | |
| | 014704 | 051124 | 041525 | 044524 | |
| | 014712 | 047117 | 053440 | 051501 | |
| | 014720 | 051440 | 051525 | 042520 | |
| | 014726 | 042116 | 042105 | 052040 | |
| | 014734 | 020117 | 042523 | 053122 | |
| | 014742 | 041511 | 020105 | 047111 | |
| | 014750 | 042524 | 051122 | 050125 | |
| | 014756 | 000124 | | | |
| 8224 | 014760 | 047045 | 040445 | 051520 | FORM43: .ASCIZ /%N%APSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT%N/' |
| | 014766 | 020127 | 044502 | 020124 | |
| | 014774 | 020070 | 044123 | 052517 | |
| | 015002 | 042114 | 044040 | 053101 | |
| | 015010 | 020105 | 042502 | 047105 | |
| | 015016 | 051440 | 052105 | 041040 | |
| | 015024 | 052125 | 053440 | 051501 | |
| | 015032 | 047040 | 052117 | 047045 | |
| | 015040 | 000 | | | |
| 8225 | 015041 | 045 | 022516 | 041101 | FORM44: .ASCIZ /%N%ABIT 8 OF PSW SET WITH PC < CIS INST PC/' |
| | 015046 | 052111 | 034040 | 047440 | |
| | 015054 | 020106 | 051520 | 020127 | |
| | 015062 | 042523 | 020124 | 044527 | |
| | 015070 | 044124 | 050040 | 020103 | |
| | 015076 | 020074 | 044503 | 020123 | |
| | 015104 | 047111 | 052123 | 050040 | |
| | 015112 | 000103 | | | |
| 8226 | 015114 | 047045 | 040445 | 052523 | FORM45: .ASCIZ /%N%ASUSPECT THAT CIS INST BACKED UP PC TOO FAR/' |
| | 015122 | 050123 | 041505 | 020124 | |
| | 015130 | 044124 | 052101 | 041440 | |
| | 015136 | 051511 | 044440 | 051516 | |
| | 015144 | 020124 | 040502 | 045503 | |
| | 015152 | 042105 | 052440 | 020120 | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| | 015160 | 041520 | 052040 | 047517 | |
| | 015166 | 043040 | 051101 | 000 | |
| 8227 | 015173 | 045 | 022516 | 004501 | FORM46: .ASCIZ /%N%A WHEN EXITING TO SERVICE INTERRUPT%N/ |
| | 015200 | 044127 | 047105 | 042440 | |
| | 015206 | 044530 | 044524 | 043516 | |
| | 015214 | 052040 | 020117 | 042523 | |
| | 015222 | 053122 | 041511 | 020105 | |
| | 015230 | 047111 | 042524 | 051122 | |
| | 015236 | 050125 | 022524 | 000116 | |
| 8228 | 015244 | 047045 | 040445 | 044503 | FORM47: .ASCIZ /%N%ACIS INST COMPLETED BUT PSW BIT 8 STILL SET%N/ |
| | 015252 | 020123 | 047111 | 052123 | |
| | 015260 | 041440 | 046517 | 046120 | |
| | 015266 | 052105 | 042105 | 041040 | |
| | 015274 | 052125 | 050040 | 053523 | |
| | 015302 | 041040 | 052111 | 034040 | |
| | 015310 | 051440 | 044524 | 046114 | |
| | 015316 | 051440 | 052105 | 047045 | |
| | 015324 | 000 | | | |
| 8229 | 015325 | 045 | 022516 | 044501 | FORM48: .ASCIZ /%N%AIN-LINE CIS INSTRUCTION COMPLETED WITH PC/ |
| | 015332 | 026516 | 044514 | 042516 | |
| | 015340 | 041440 | 051511 | 044440 | |
| | 015346 | 051516 | 051124 | 041525 | |
| | 015354 | 044524 | 047117 | 041440 | |
| | 015362 | 046517 | 046120 | 052105 | |
| | 015370 | 042105 | 053440 | 052111 | |
| | 015376 | 020110 | 041520 | 000 | |
| 8230 | 015403 | 045 | 022516 | 020101 | FORM49: .ASCIZ /%N%A POINTING AT IN-LINE OPERANDS RATHER THAN NEXT INST%N/ |
| | 015410 | 050011 | 044517 | 052116 | |
| | 015416 | 047111 | 020107 | 052101 | |
| | 015424 | 044440 | 026516 | 044514 | |
| | 015432 | 042516 | 047440 | 042520 | |
| | 015440 | 040522 | 042116 | 020123 | |
| | 015446 | 040522 | 044124 | 051105 | |
| | 015454 | 052040 | 040510 | 020116 | |
| | 015462 | 042516 | 052130 | 044440 | |
| | 015470 | 051516 | 022524 | 000116 | |
| 8231 | 015476 | 006477 | 040412 | 042104 | QUES: .ASCIZ /?/<CR><LF>/ADDR(S)?/ |
| | 015504 | 024122 | 024523 | 000077 | |
| 8232 | 015512 | 005015 | 000 | | XCR LF: .ASCIZ <CR><LF> |
| 8233 | 015515 | 057 | 005015 | 000 | SLCRLF: .ASCIZ <SL><CR><LF> |
| 8235 | 015521 | 015 | 042412 | 042116 | ENDP: .ASCIZ <CR><LF>/END OF PASS (EXECUTION OF TABLED TEST CASES COMPLETE)/ |
| | 015526 | 047440 | 020106 | 040520 | |
| | 015534 | 051523 | 024040 | 054105 | |
| | 015542 | 041505 | 052125 | 047511 | |
| | 015550 | 020116 | 043117 | 052040 | |
| | 015556 | 041101 | 042514 | 020104 | |
| | 015564 | 042524 | 052123 | 041440 | |
| | 015572 | 051501 | 051505 | 041440 | |
| | 015600 | 046517 | 046120 | 052105 | |
| | 015606 | 024505 | 000 | | |
| 8240 | 015611 | 015 | 042412 | 042116 | ENDQP: .ASCIZ <CR><LF>/END OF QUICK VERIFY PASS/ |
| | 015616 | 047440 | 020106 | 052521 | |
| | 015624 | 041511 | 020113 | 042526 | |
| | 015632 | 044522 | 054506 | 050040 | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| 8241 | 015640 | 051501 | 000123 | | |
| | 015644 | 005015 | 047111 | 052123 | FSHDR: .ASCIZ <CR><LF>/INST UNDER TEST WILL BE DISPLAYED AT THE START OF TESTING FOR E |
| | 015652 | 052440 | 042116 | 051105 | |
| | 015660 | 052040 | 051505 | 020124 | |
| | 015666 | 044527 | 046114 | 041040 | |
| | 015674 | 020105 | 044504 | 050123 | |
| | 015702 | 040514 | 042531 | 020104 | |
| | 015710 | 052101 | 052040 | 042510 | |
| | 015716 | 051440 | 040524 | 052122 | |
| | 015724 | 047440 | 020106 | 042524 | |
| | 015732 | 052123 | 047111 | 020107 | |
| | 015740 | 047506 | 020122 | 040505 | |
| | 015746 | 044103 | 044440 | 051516 | |
| | 015754 | 000124 | | | |
| 8242 | 015756 | 005015 | 052521 | 041511 | QVHDR: .ASCIZ <CR><LF>/QUICK VERIFY PASS TIME: APPROX. 3 MINUTES/ |
| | 015764 | 020113 | 042526 | 044522 | |
| | 015772 | 054506 | 050040 | 051501 | |
| | 016000 | 020123 | 044524 | 042515 | |
| | 016006 | 020072 | 040440 | 050120 | |
| | 016014 | 047522 | 027130 | 031440 | |
| | 016022 | 046440 | 047111 | 052125 | |
| | 016030 | 051505 | 000 | | |
| 8243 | 016033 | 045 | 022516 | 042501 | ACCSEED: .ASCIZ /%N%AENTER 3 RANDOM NUMBER GEN. SEED CONSTANTS: %N/ |
| | 016040 | 052116 | 051105 | 031440 | |
| | 016046 | 051040 | 047101 | 047504 | |
| | 016054 | 020115 | 052516 | 041115 | |
| | 016062 | 051105 | 043440 | 047105 | |
| | 016070 | 020056 | 042523 | 042105 | |
| | 016076 | 041440 | 047117 | 052123 | |
| | 016104 | 047101 | 051524 | 020072 | |
| | 016112 | 022440 | 000116 | | |
| 8245 | 016116 | 005015 | 040520 | 051523 | FSHDR1: .ASCIZ <CR><LF>+PASS TIME: 11/44 APPROX 30 MIN+ |
| | 016124 | 052040 | 046511 | 035105 | |
| | 016132 | 030440 | 027461 | 032064 | |
| | 016140 | 040440 | 050120 | 047522 | |
| | 016146 | 020130 | 030063 | 046440 | |
| | 016154 | 047111 | 000 | | |
| 8250 | 016157 | 015 | 042412 | 052116 | FSHDR2: .ASCIZ <CR><LF>/ENTERING RANDOM TEST MODE/ |
| | 016164 | 051105 | 047111 | 020107 | |
| | 016172 | 040522 | 042116 | 046517 | |
| | 016200 | 052040 | 051505 | 020124 | |
| | 016206 | 047515 | 042504 | 000 | |
| 8251 | 016213 | 015 | 047012 | 020117 | FSHDR3: .ASCIZ <CR><LF>/NO FURTHER END OF PASS MESSAGES WILL BE ISSUED/ |
| | 016220 | 052506 | 052122 | 042510 | |
| | 016226 | 020122 | 047105 | 020104 | |
| | 016234 | 043117 | 050040 | 051501 | |
| | 016242 | 020123 | 042515 | 051523 | |
| | 016250 | 043501 | 051505 | 053440 | |
| | 016256 | 046111 | 020114 | 042502 | |
| | 016264 | 044440 | 051523 | 042525 | |
| | 016272 | 000104 | | | |
| 8252 | 016274 | 005015 | 040522 | 042116 | FSHDR4: .ASCIZ <CR><LF>/RANDOM # GENERATOR SEED CONSTANTS WILL BE PRINTED/ |
| | 016302 | 046517 | 021440 | 043440 | |
| | 016310 | 047105 | 051105 | 052101 | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| | 016316 | 051117 | 051440 | 042505 | |
| | 016324 | 020104 | 047503 | 051516 | |
| | 016332 | 040524 | 052116 | 020123 | |
| | 016340 | 044527 | 046114 | 041040 | |
| | 016346 | 020105 | 051120 | 047111 | |
| | 016354 | 042524 | 000104 | | |
| 8253 | 016360 | 005015 | 020040 | 020040 | FSHDR5: .ASCIZ <CR><LF>/ EVERY 1024 CIS INSTRUCTION TESTS/ |
| | 016366 | 053105 | 051105 | 020131 | |
| | 016374 | 030061 | 032062 | 041440 | |
| | 016402 | 051511 | 044440 | 051516 | |
| | 016410 | 051124 | 041525 | 044524 | |
| | 016416 | 047117 | 052040 | 051505 | |
| | 016424 | 051524 | 000 | | |
| 8257 | 016427 | 045 | 026501 | 026455 | DASH: .ASCIZ /%A-----%N/ |
| | 016434 | 026455 | 026455 | 026455 | |
| | 016442 | 026455 | 026455 | 026455 | |
| | 016450 | 026455 | 026455 | 026455 | |
| | 016456 | 026455 | 026455 | 026455 | |
| | 016464 | 026455 | 026455 | 026455 | |
| | 016472 | 026455 | 026455 | 026455 | |
| | 016500 | 026455 | 026455 | 026455 | |
| | 016506 | 026455 | 026455 | 026455 | |
| | 016514 | 026455 | 026455 | 026455 | |
| | 016522 | 026455 | 047045 | 000 | |
| 8258 | 016527 | 045 | 022516 | 026501 | SDASH: .ASCIZ /%N%A-----/ |
| | 016534 | 026455 | 026455 | 026455 | |
| | 016542 | 026455 | 026455 | 026455 | |
| | 016550 | 000 | | | |
| 8260 | 016551 | 015 | 041412 | 045532 | PNAME: .ASCIZ <CR><LF>/CZKEEA0 PDP-11 CIS INSTRUCTION EXERCISER / |
| | 016556 | 042505 | 030101 | 050040 | |
| | 016564 | 050104 | 030455 | 020061 | |
| | 016572 | 044503 | 020123 | 047111 | |
| | 016600 | 052123 | 052522 | 052103 | |
| | 016606 | 047511 | 020116 | 054105 | |
| | 016614 | 051105 | 044503 | 042523 | |
| | 016622 | 020122 | 000 | | |
| 8265 | 016625 | 015 | 030412 | 027461 | MPT34: .ASCIZ <CR><LF>+11/34 TYPE MEM MGMT ON SYSTEM UNDER TEST+ |
| | 016632 | 032063 | 052040 | 050131 | |
| | 016640 | 020105 | 042515 | 020115 | |
| | 016646 | 043515 | 052115 | 047440 | |
| | 016654 | 020116 | 054523 | 052123 | |
| | 016662 | 046505 | 052440 | 042116 | |
| | 016670 | 051105 | 052040 | 051505 | |
| | 016676 | 000124 | | | |
| 8266 | | | | | |
| 8267 | | | | | |

```
8269 .SBTTL GLOBAL SUBROUTINES SECTION
8270
8271 .SBTTL CIS EMULATOR
8272 : FUNCTIONAL DESCRIPTION:
8273 : CIS EMULATOR CONTAINS ROUTINES TO EMULATE EACH OF THE
8274 : CIS INSTRUCTIONS USING STANDARD PDP-11 INSTRUCTIONS
8275
8276 : INPUTS: CIS INSTRUCTION TO EMULATE
8277 : CIS INSTRUCTION OPERANDS (LENGTHS,ADDRESSES,ETC)
8278 : STARTING ADDRESS FOR REGISTER RESULTS
8279 : ADDRESS FOR CONDITION CODE RESULTS
8280
8281 ; IMPLICIT INPUTS: CHARACTER OR DECIMAL STRINGS SETUP IN EMULATION BUFFER
8282
8283 : OUTPUTS: CONDITION CODES
8284 : GENERAL PURPOSE REGISTERS
8285 : STRINGS IN EMULATION BUFFER
8286
8287 ; IMPLICIT OUTPUTS:
8288
8289 ; SUBORDINATE ROUTINES USED:
8290
8291 ; FUNCTIONAL SIDE EFFECTS:
8292
8293 : CALLING SEQUENCE: JSR PC,EMULATE
8294 : XXXXXX ;OCTAL ENCODING OF CIS INST
8295 : YYYYYY ;POINTER TO REGISTER OPERANDS
8296 : ZZZZZZ ;POINTER TO REGISTER RESULTS
8297 : WWWWWW ;POINTER TO CONDITION CODE RESULTS
8298
8299 177776 EPSW = 177776
8300 .EVEN
```

| | | | | | | |
|------|--------|--------|--------|---------|------------------|-------------------------------------|
| 8302 | | | | | .SBTTL | INSTRUCTION DECODER |
| 8303 | 016700 | 010037 | 017072 | EMULAT: | MOV R0,ESTORE | ;SAVE OLD REGISTER VALUES |
| 8304 | 016704 | 010137 | 017074 | | MOV R1,ESTORE+2 | |
| 8305 | 016710 | 010237 | 017076 | | MOV R2,ESTORE+4 | |
| 8306 | 016714 | 010337 | 017100 | | MOV R3,ESTORE+6 | |
| 8307 | 016720 | 010437 | 017102 | | MOV R4,ESTORE+10 | |
| 8308 | 016724 | 010537 | 017104 | | MOV R5,ESTORE+12 | |
| 8309 | 016730 | 005037 | 035744 | | CLR EZDF | ;CLEAR ZERO DIVIDE FLAG |
| 8310 | 016734 | 011600 | | | MOV (SP),R0 | ;GRAB DATA POINTERS |
| 8311 | 016736 | 012037 | 017110 | | MOV (R0)+,EINST | ;CIS INSTRUCTION BEING CALLED |
| 8312 | 016742 | 012037 | 017112 | | MOV (R0)+,EIRSTK | ;FAKE INPUT GPRS |
| 8313 | 016746 | 012037 | 017114 | | MOV (R0)+,EORSTK | ;FAKE OUTPUT GPRS |
| 8314 | 016752 | 012037 | 017116 | | MOV (R0)+,EOPSW | ;FAKE PSW |
| 8315 | 016756 | 010016 | | | MOV R0,(SP) | ;SUB RETURN ADDRESS |
| 8316 | 016760 | 012700 | 017140 | | MOV #ELISTA,R0 | ;CIS COMMAND LIST POINTER |
| 8317 | 016764 | 005710 | | 1\$: | TST (R0) | ;EXIT IF INSTRUCTION CANT DECODE |
| 8318 | 016766 | 001001 | | | BNE 2\$ | ;ELSE CONTINUE SEARCH |
| 8319 | 016770 | 000207 | | | RTS PC | |
| 8320 | 016772 | 023720 | 017110 | 2\$: | CMP EINST,(R0)+ | ;LOOK AT TABLE FOR MATCH |
| 8321 | 016776 | 001372 | | | BNE 1\$ | ;KEEP TRYING |
| 8322 | 017000 | 162700 | 017142 | | SUB #ELISTA+2,R0 | ;AT LAST, SO FIND HANDLER FOR IT |
| 8323 | 017004 | 062700 | 017264 | | ADD #ELISTB,R0 | ;HANDLER ADDRESS IN TABLE B |
| 8324 | 017010 | 011037 | 017106 | | MOV (R0),EROUT | ;HOLD ADDRESS WHILE I FIND THE DATA |
| 8325 | 017014 | 012700 | 017120 | | MOV #E0,R0 | ;POINTER TO EMULATE GPRS |
| 8326 | 017020 | 013701 | 017112 | | MOV EIRSTK,R1 | ;POINTER TO REGISTER DATA |
| 8327 | 017024 | 012120 | | 3\$: | MOV (R1)+,(R0)+ | ;BEAM OVER THE DATA |
| 8328 | 017026 | 022700 | 017136 | | CMP #E6+2,R0 | ;EXIT LOOP AFTER R6 LOADED |
| 8329 | 017032 | 001374 | | | BNE 3\$ | ;ELSE LOAD NEXT |
| 8330 | 017034 | 004777 | 000046 | | JSR PC,@EROUT | ;EXECUTE EMULATED CIS INSTRUCTION |
| 8331 | 017040 | 013700 | 017072 | | MOV ESTORE,R0 | ;ON RETURN RESTORE REGISTERS |
| 8332 | 017044 | 013701 | 017074 | | MOV ESTORE+2,R1 | |
| 8333 | 017050 | 013702 | 017076 | | MOV ESTORE+4,R2 | |
| 8334 | 017054 | 013703 | 017100 | | MOV ESTORE+6,R3 | |
| 8335 | 017060 | 013704 | 017102 | | MOV ESTORE+10,R4 | |
| 8336 | 017064 | 013705 | 017104 | | MOV ESTORE+12,R5 | |
| 8337 | 017070 | 000207 | | | RTS PC | ;RETURN TO MAIN PROGRAM |

| 8339 | | | | .SBTTL | |
|------|--------|--------|---------|--------|--------|
| 8340 | 017072 | 000006 | ESTORE: | .BLKW | 6 |
| 8341 | 017106 | 000000 | EROUT: | .WORD | 0 |
| 8342 | 017110 | 000000 | EINST: | .WORD | 0 |
| 8343 | 017112 | 000000 | EIRSTK: | .WORD | 0 |
| 8344 | 017114 | 000000 | EORSTK: | .WORD | 0 |
| 8345 | 017116 | 000000 | EOPSW: | .WORD | 0 |
| 8346 | 017120 | 000000 | E0: | .WORD | 0 |
| 8347 | 017122 | 000000 | E1: | .WORD | 0 |
| 8348 | 017124 | 000000 | E2: | .WORD | 0 |
| 8349 | 017126 | 000000 | E3: | .WORD | 0 |
| 8350 | 017130 | 000000 | E4: | .WORD | 0 |
| 8351 | 017132 | 000000 | E5: | .WORD | 0 |
| 8352 | 017134 | 000000 | E6: | .WORD | 0 |
| 8353 | 017136 | 000000 | TEMP: | .WORD | 0 |
| 8354 | 017140 | 076020 | ELISTA: | .WORD | 076020 |
| 8355 | 017142 | 076021 | | .WORD | 076021 |
| 8356 | 017144 | 076022 | | .WORD | 076022 |
| 8357 | 017146 | 076023 | | .WORD | 076023 |
| 8358 | 017150 | 076024 | | .WORD | 076024 |
| 8359 | 017152 | 076025 | | .WORD | 076025 |
| 8360 | 017154 | 076026 | | .WORD | 076026 |
| 8361 | 017156 | 076027 | | .WORD | 076027 |
| 8362 | 017160 | 076030 | | .WORD | 76030 |
| 8363 | 017162 | 076031 | | .WORD | 76031 |
| 8364 | 017164 | 076032 | | .WORD | 76032 |
| 8365 | 017166 | 076040 | | .WORD | 76040 |
| 8366 | 017170 | 076041 | | .WORD | 76041 |
| 8367 | 017172 | 076042 | | .WORD | 76042 |
| 8368 | 017174 | 076043 | | .WORD | 76043 |
| 8369 | 017176 | 076044 | | .WORD | 76044 |
| 8370 | 017200 | 076045 | | .WORD | 76045 |
| 8371 | 017202 | 076050 | | .WORD | 76050 |
| 8372 | 017204 | 076051 | | .WORD | 76051 |
| 8373 | 017206 | 076052 | | .WORD | 76052 |
| 8374 | 017210 | 076053 | | .WORD | 76053 |
| 8375 | 017212 | 076054 | | .WORD | 76054 |
| 8376 | 017214 | 076055 | | .WORD | 76055 |
| 8377 | 017216 | 076056 | | .WORD | 76056 |
| 8378 | 017220 | 076057 | | .WORD | 76057 |
| 8379 | 017222 | 076060 | | .WORD | 076060 |
| 8380 | 017224 | 076061 | | .WORD | 076061 |
| 8381 | 017226 | 076062 | | .WORD | 076062 |
| 8382 | 017230 | 076063 | | .WORD | 076063 |
| 8383 | 017232 | 076064 | | .WORD | 076064 |
| 8384 | 017234 | 076065 | | .WORD | 076065 |
| 8385 | 017236 | 076066 | | .WORD | 076066 |
| 8386 | 017240 | 076067 | | .WORD | 076067 |
| 8387 | 017242 | 076070 | | .WORD | 76070 |
| 8388 | 017244 | 076071 | | .WORD | 76071 |
| 8389 | 017246 | 076072 | | .WORD | 76072 |
| 8390 | 017250 | 076073 | | .WORD | 76073 |
| 8391 | 017252 | 076074 | | .WORD | 76074 |
| 8392 | 017254 | 076075 | | .WORD | 76075 |

DATA STORAGE

| | | | | |
|------|--------|--------|-------|--------|
| 8393 | 017256 | 076076 | .WORD | 76076 |
| 8394 | 017260 | 076077 | .WORD | 76077 |
| 8395 | 017262 | 000000 | .WORD | 0 |
| 8396 | 017264 | 035752 | .WORD | EL2D0 |
| 8397 | 017266 | 036056 | .WORD | EL2D1 |
| 8398 | 017270 | 036064 | .WORD | EL2D2 |
| 8399 | 017272 | 036072 | .WORD | EL2D3 |
| 8400 | 017274 | 036100 | .WORD | EL2D4 |
| 8401 | 017276 | 036114 | .WORD | EL2D5 |
| 8402 | 017300 | 036130 | .WORD | EL2D6 |
| 8403 | 017302 | 036174 | .WORD | EL2D7 |
| 8404 | 017304 | 017406 | .WORD | EMOVC |
| 8405 | 017306 | 020112 | .WORD | EMOVRC |
| 8406 | 017310 | 017630 | .WORD | EMOVTC |
| 8407 | 017312 | 021002 | .WORD | ELOCC |
| 8408 | 017314 | 021034 | .WORD | ESKPC |
| 8409 | 017316 | 020724 | .WORD | ESCNC |
| 8410 | 017320 | 020646 | .WORD | ESPNC |
| 8411 | 017322 | 020334 | .WORD | ECMPC |
| 8412 | 017324 | 020536 | .WORD | EMTCHC |
| 8413 | 017326 | 023146 | .WORD | EADDN |
| 8414 | 017330 | 023214 | .WORD | ESUBN |
| 8415 | 017332 | 032050 | .WORD | ECMPN |
| 8416 | 017334 | 026522 | .WORD | ECVTNL |
| 8417 | 017336 | 027266 | .WORD | ECVTPN |
| 8418 | 017340 | 027700 | .WORD | ECVTNP |
| 8419 | 017342 | 030452 | .WORD | EASHN |
| 8420 | 017344 | 024724 | .WORD | ECVTLN |
| 8421 | 017346 | 036234 | .WORD | EL3D0 |
| 8422 | 017350 | 036310 | .WORD | EL3D1 |
| 8423 | 017352 | 036316 | .WORD | EL3D2 |
| 8424 | 017354 | 036324 | .WORD | EL3D3 |
| 8425 | 017356 | 036332 | .WORD | EL3D4 |
| 8426 | 017360 | 036340 | .WORD | EL3D5 |
| 8427 | 017362 | 036346 | .WORD | EL3D6 |
| 8428 | 017364 | 036426 | .WORD | EL3D7 |
| 8429 | 017366 | 023136 | .WORD | EADDP |
| 8430 | 017370 | 023154 | .WORD | ESUBP |
| 8431 | 017372 | 031774 | .WORD | ECMPP |
| 8432 | 017374 | 026530 | .WORD | ECVTPL |
| 8433 | 017376 | 032634 | .WORD | EMULP |
| 8434 | 017400 | 034676 | .WORD | EDIVP |
| 8435 | 017402 | 030376 | .WORD | EASHP |
| 8436 | 017404 | 024732 | .WORD | ECVTLP |

ELISTB:

| | | | | | | | |
|------|--------|--------|--------|--------|---------|------------------|-------------------------------------|
| 8438 | | | | | | .SBTTL | MOVE STRING |
| 8439 | 017406 | 013700 | 017122 | | EMOVC: | MOV E1,R0 | :FIND END OF SOURCE STRING |
| 8440 | 017412 | 063700 | 017120 | | | ADD E0,R0 | |
| 8441 | 017416 | 013701 | 017126 | | | MOV E3,R1 | :FIND END OF DEST. STRING |
| 8442 | 017422 | 063701 | 017124 | | | ADD E2,R1 | |
| 8443 | 017426 | 023737 | 017120 | 017124 | | CMP E0,E2 | :WICH STRING IS LONGER |
| 8444 | 017434 | 101003 | | | | BHI 1\$ | :SOURCE |
| 8445 | 017436 | 103416 | | | | BLO 2\$ | :DEST. |
| 8446 | 017440 | 010102 | | | | MOV R1,R2 | :THEIR THE SAME |
| 8447 | 017442 | 000420 | | | | BR 3\$ | |
| 8448 | 017444 | 013700 | 017122 | | 1\$: | MOV E1,R0 | :SHORTEN SOURCE STRING |
| 8449 | 017450 | 063700 | 017124 | | | ADD E2,R0 | |
| 8450 | 017454 | 010102 | | | | MOV R1,R2 | :DEST USED END = REAL END |
| 8451 | 017456 | 013703 | 017120 | | | MOV E0,R3 | :CALCULATE # OF CHARS |
| 8452 | 017462 | 163703 | 017124 | | | SUB E2,R3 | : NOT TRANSFERRED. |
| 8453 | 017466 | 010377 | 177422 | | | MOV R3,@EORSTK | :SAVE RESULT |
| 8454 | 017472 | 000406 | | | | BR 4\$ | |
| 8455 | 017474 | 013702 | 017126 | | 2\$: | MOV E3,R2 | :USED END < REAL END |
| 8456 | 017500 | 063702 | 017120 | | | ADD E0,R2 | |
| 8457 | 017504 | 005077 | 177404 | | 3\$: | CLR @EORSTK | :ALL CHAR. TRANSFERED TO DEST. |
| 8458 | 017510 | 023737 | 017122 | 017126 | 4\$: | CMP E1,E3 | :WICH STRING IS IN HIGH CORE |
| 8459 | 017516 | 103410 | | | | BLO EFORWD | :DEST STRING IS |
| 8460 | 017520 | 013703 | 017122 | | EBACK: | MOV E1,R3 | :START ADDRESS OF SOURCE |
| 8461 | 017524 | 013704 | 017126 | | | MOV E3,R4 | :START ADDRESS OF DEST. |
| 8462 | 017530 | 020402 | | | 1\$: | CMP R4,R2 | :IS TRANSFER COMPLETE ? |
| 8463 | 017532 | 001410 | | | | BEQ EFILL | :YES |
| 8464 | 017534 | 112324 | | | | MOVB (R3)+,(R4)+ | :XFER CHAR. |
| 8465 | 017536 | 000774 | | | | BR 1\$ | |
| 8466 | 017540 | 010203 | | | EFORWD: | MOV R2,R3 | :DEST STRING POINTER |
| 8467 | 017542 | 023700 | 017122 | | 1\$: | CMP E1,R0 | :IS XFER COMPLETE ? |
| 8468 | 017546 | 001402 | | | | BEQ EFILL | :YES |
| 8469 | 017550 | 114043 | | | | MOVB -(R0),-(R3) | :XFER CHAR. |
| 8470 | 017552 | 000773 | | | | BR 1\$ | |
| 8471 | 017554 | 020102 | | | EFILL: | CMP R1,R2 | :ADD FILL CHARS. TO COMPLETE STRING |
| 8472 | 017556 | 001403 | | | | BEQ 2\$ | |
| 8473 | 017560 | 113722 | 017130 | | | MOVB E4,(R2)+ | :XFER FILL |
| 8474 | 017564 | 000773 | | | | BR EFILL | |
| 8475 | 017566 | 013700 | 017114 | | 2\$: | MOV EORSTK,R0 | :RETURN CLEAN UP |
| 8476 | 017572 | 005720 | | | | TST (R0)+ | :R0 = R0 |
| 8477 | 017574 | 005020 | | | | CLR (R0)+ | :R1 = 0 |
| 8478 | 017576 | 005020 | | | | CLR (R0)+ | :R2 = 0 |
| 8479 | 017600 | 005020 | | | | CLR (R0)+ | :R3 = 0 |
| 8480 | 017602 | 013720 | 017130 | | | MOV E4,(R0)+ | :R4 = R4 |
| 8481 | 017606 | 013710 | 017132 | | | MOV E5,(R0) | :R5 = R5 |
| 8482 | 017612 | 023737 | 017120 | 017124 | | CMP E0,E2 | :SET PSW CC BITS |
| 8483 | 017620 | 013777 | 177776 | 177270 | | MOV EPSW,@EOPSW | :STORE RESULT |
| 8484 | 017626 | 000207 | | | | RTS PC | |

| | | | | | | | |
|------|--------|--------|--------|--------|------|------------------|------------------------------|
| 8486 | | | | | | .SBTTL | MOVE STRING TRANSLATE |
| 8487 | 017630 | 013700 | 017122 | | | MOV E1,R0 | :FIND END OF SOURCE |
| 8488 | 017634 | 063700 | 017120 | | | ADD E0,R0 | |
| 8489 | 017640 | 013701 | 017126 | | | MOV E3,R1 | :FIND END OF DEST. |
| 8490 | 017644 | 063701 | 017124 | | | ADD E2,R1 | |
| 8491 | 017650 | 023737 | 017120 | 017124 | | CMP E0,E2 | :WHICH STRING IS LONGER |
| 8492 | 017656 | 101003 | | | | BHI 1\$ | :SOURCE |
| 8493 | 017660 | 103416 | | | | BLO 2\$ | :DEST. |
| 8494 | 017662 | 010102 | | | | MOV R1,R2 | :SAME |
| 8495 | 017664 | 000420 | | | | BR 3\$ | |
| 8496 | 017666 | 013700 | 017122 | | 1\$: | MOV E1,R0 | :SHORTEN SOURCE STRING |
| 8497 | 017672 | 063700 | 017124 | | | ADD E2,R0 | |
| 8498 | 017676 | 010102 | | | | MOV R1,R2 | |
| 8499 | 017700 | 013703 | 017120 | | | MOV E0,R3 | :CALCULATE NO. OF CHARS. |
| 8500 | 017704 | 163703 | 017124 | | | SUB E2,R3 | : NOT TRANSFERRED |
| 8501 | 017710 | 010377 | 177200 | | | MOV R3,@EORSTK | :STORE RESULT |
| 8502 | 017714 | 000406 | | | | BR 4\$ | |
| 8503 | 017716 | 013702 | 017126 | | 2\$: | MOV E3,R2 | :MARK REAL END OF DEST |
| 8504 | 017722 | 063702 | 017120 | | | ADD E0,R2 | |
| 8505 | 017726 | 005077 | 177162 | | 3\$: | CLR @EORSTK | |
| 8506 | 017732 | 023737 | 017122 | 017126 | 4\$: | CMP E1,E3 | :WHO'S HIGHER IN MEMORY |
| 8507 | 017740 | 103420 | | | | BLO EMTFRD | :MOVE FORWARD |
| 8508 | 017742 | 013703 | 017122 | | | MOV E1,R3 | :START OF SOURCE |
| 8509 | 017746 | 013704 | 017126 | | | MOV E3,R4 | :START OF DEST. |
| 8510 | 017752 | 020402 | | | 1\$: | CMP R4,R2 | :XFER COMPLETE YET |
| 8511 | 017754 | 001430 | | | | BEQ EMTFIL | :YES |
| 8512 | 017756 | 112337 | 017136 | | | MOVB (R3)+,TEMP | :CAL. INDEX INTO TABLE |
| 8513 | 017762 | 105037 | 017137 | | | CLRB TEMP+1 | |
| 8514 | 017766 | 063737 | 017132 | 017136 | | ADD E5,TEMP | |
| 8515 | 017774 | 117724 | 177136 | | | MOVB @TEMP,(R4)+ | :MOVE TABLE VALUE TO DEST. |
| 8516 | 020000 | 000764 | | | | BR 1\$ | |
| 8517 | 020002 | 010203 | | | | MOV R2,R3 | :DEST MOVE POINTER |
| 8518 | 020004 | 023700 | 017122 | | 1\$: | CMP E1,R0 | :XFER COMPLETE ? |
| 8519 | 020010 | 001412 | | | | BEQ EMTFIL | |
| 8520 | 020012 | 114037 | 017136 | | | MOVB -(R0),TEMP | :CAL INDEX INTO TABLE |
| 8521 | 020016 | 105037 | 017137 | | | CLRB TEMP+1 | |
| 8522 | 020022 | 063737 | 017132 | 017136 | | ADD E5,TEMP | |
| 8523 | 020030 | 117743 | 177102 | | | MOVB @TEMP,-(R3) | :MOVE TABLE VALUE INTO DEST. |
| 8524 | 020034 | 000763 | | | | BR 1\$ | |
| 8525 | 020036 | 020102 | | | | CMP R1,R2 | :COMPLETE |
| 8526 | 020040 | 001403 | | | | BEQ 2\$ | :YES |
| 8527 | 020042 | 113722 | 017130 | | | MOVB E4,(R2)+ | :XFER FILL |
| 8528 | 020046 | 000773 | | | | BR EMTFIL | |
| 8529 | 020050 | 013700 | 017114 | | 2\$: | MOV EORSTK,R0 | :RETURN CLEAN UP |
| 8530 | 020054 | 005720 | | | | TST (R0)+ | :R0 = R0 |
| 8531 | 020056 | 005020 | | | | CLR (R0)+ | :R1 = 0 |
| 8532 | 020060 | 005020 | | | | CLR (R0)+ | :R2 = 0 |
| 8533 | 020062 | 005020 | | | | CLR (R0)+ | :R3 = 0 |
| 8534 | 020064 | 013720 | 017130 | | | MOV E4,(R0)+ | :R4 = R4 |
| 8535 | 020070 | 013710 | 017132 | | | MOV E5,(R0) | :R5 = R5 |
| 8536 | 020074 | 023737 | 017120 | 017124 | | CMP E0,E2 | :SET CC BITS |
| 8537 | 020102 | 013777 | 177776 | 177006 | | MOV EPSW,@EOPSW | :RETURN STATUS |
| 8538 | 020110 | 000207 | | | | RTS PC | |

| | | | | | | | |
|------|--------|--------|--------|--------|----------|------------------|---------------------------|
| 8541 | | | | | | .SBTTL | MOVE REVERSE STRING |
| 8542 | 020112 | 013700 | 017122 | | EMOVRTC: | MOV E1,R0 | :FIND END OF SOURCE |
| 8543 | 020116 | 063700 | 017120 | | | ADD E0,R0 | |
| 8544 | 020122 | 013701 | 017126 | | | MOV E3,R1 | :FIND END OF DEST. |
| 8545 | 020126 | 063701 | 017124 | | | ADD E2,R1 | |
| 8546 | 020132 | 023737 | 017120 | 017124 | | CMP E0,E2 | :WHICH STRING IS LARGER |
| 8547 | 020140 | 101004 | | | | BHI 1\$ | :SOURCE |
| 8548 | 020142 | 103421 | | | | BLO 2\$ | :DEST. |
| 8549 | 020144 | 013702 | 017126 | | | MOV E3,R2 | :SAME |
| 8550 | 020150 | 000421 | | | | BR 3\$ | |
| 8551 | 020152 | 010037 | 017122 | | 1\$: | MOV R0,E1 | :SHORTEN SOURCE |
| 8552 | 020156 | 163737 | 017124 | 017122 | | SUB E2,E1 | |
| 8553 | 020164 | 013702 | 017126 | | | MOV E3,R2 | :DEST. REAL START |
| 8554 | 020170 | 013703 | 017120 | | | MOV E0,R3 | :CALCULATE NO OF CHARS |
| 8555 | 020174 | 163703 | 017124 | | | SUB E2,R3 | : NOT TRANSFERRED |
| 8556 | 020200 | 010377 | 176710 | | | MOV R3,@EORSTK | :STORE RESULT |
| 8557 | 020204 | 000405 | | | | BR 4\$ | |
| 8558 | 020206 | 010102 | | | 2\$: | MOV R1,R2 | :MARK REAL START OF DEST. |
| 8559 | 020210 | 163702 | 017120 | | | SUB E0,R2 | |
| 8560 | 020214 | 005077 | 176674 | | 3\$: | CLR @EORSTK | :ALL CHARS. TRANSFERED |
| 8561 | 020220 | 023702 | 017122 | | 4\$: | CMP E1,R2 | :WHO'S IN HIGH MEMORY |
| 8562 | 020224 | 103407 | | | | BLO EMRFRWD | :MOVE FORWARD |
| 8563 | 020226 | 013703 | 017122 | | EMRDKD: | MOV E1,R3 | :SOURCE START POINTER |
| 8564 | 020232 | 010204 | | | | MOV R2,R4 | :DEST. START POINTER |
| 8565 | 020234 | 020401 | | | 1\$: | CMP R4,R1 | :XFER COMPLETE |
| 8566 | 020236 | 001406 | | | | BEQ EMRFRIL | :YES |
| 8567 | 020240 | 112324 | | | | MOVB (R3)+,(R4)+ | :XFER CHAR. |
| 8568 | 020242 | 000774 | | | | BR 1\$ | |
| 8569 | 020244 | 020102 | | | EMRFRWD: | CMP R1,R2 | :XFER COMPLETE |
| 8570 | 020246 | 001402 | | | | BEQ EMRFRIL | :YES |
| 8571 | 020250 | 114041 | | | | MOVB -(R0),-(R1) | :XFER CHAR. |
| 8572 | 020252 | 000774 | | | | BR EMRFRWD | |
| 8573 | 020254 | 013703 | 017126 | | EMRFRIL: | MOV E3,R3 | :ADD FILLER |
| 8574 | 020260 | 020302 | | | 2\$: | CMP R3,R2 | :FILL COMPLETE |
| 8575 | 020262 | 001403 | | | | BEQ 1\$ | :YES |
| 8576 | 020264 | 113723 | 017130 | | | MOVB E4,(R3)+ | :XFER FILL |
| 8577 | 020270 | 000773 | | | | BR 2\$ | |
| 8578 | 020272 | 013700 | 017114 | | 1\$: | MOV EORSTK,R0 | :RETURN CLEAN UP |
| 8579 | 020276 | 005720 | | | | TST (R0)+ | :R0 = R0 |
| 8580 | 020300 | 005020 | | | | CLR (R0)+ | :R1 = 0 |
| 8581 | 020302 | 005020 | | | | CLR (R0)+ | :R2 = 0 |
| 8582 | 020304 | 005020 | | | | CLR (R0)+ | :R3 = 0 |
| 8583 | 020306 | 013720 | 017130 | | | MOV E4,(R0)+ | :R4 = R4 |
| 8584 | 020312 | 013710 | 017132 | | | MOV E5,(R0) | :R5 = R5 |
| 8585 | 020316 | 023737 | 017120 | 017124 | | CMP E0,E2 | :SET CC BITS |
| 8586 | 020324 | 013777 | 177776 | 176564 | | MOV EPSW,@EOPSW | :RETURN TO USER |
| 8587 | 020332 | 000207 | | | | RTS PC | |

| | | | | | | |
|------|--------|--------|--------|---------|------------------|-----------------------------|
| 8589 | | | | | .SBTTL | COMPARE STRING |
| 8590 | 020334 | 013700 | 017120 | ECMPC: | MOV E0,R0 | ;CAL. END OF SCR1 |
| 8591 | 020340 | 063700 | 017122 | | ADD E1,R0 | |
| 8592 | 020344 | 013701 | 017124 | | MOV E2,R1 | ;CAL. END OF SCR2 |
| 8593 | 020350 | 063701 | 017126 | | ADD E3,R1 | |
| 8594 | 020354 | 013702 | 017122 | | MOV E1,R2 | ;START OF SCR1 |
| 8595 | 020360 | 013703 | 017126 | | MOV E3,R3 | ;START OF SCR2 |
| 8596 | 020364 | 020002 | | 1\$: | CMP R0,R2 | ;END OF SCR1 |
| 8597 | 020366 | 001427 | | | BEQ ENDA | ;YES |
| 8598 | 020370 | 020103 | | | CMP R1,R3 | ;END OF SCR2 |
| 8599 | 020372 | 001445 | | | BEQ ENDB | ;YES |
| 8600 | 020374 | 121213 | | | CMPB (R2),(R3) | ;SET CC BITS |
| 8601 | 020376 | 013777 | 177776 | 176512 | MOV EPSW,@EOPSW | ;STORE STATUS |
| 8602 | 020404 | 122322 | | | CMPB (R3)+,(R2)+ | ;FIND NON MATCHING CHARS. |
| 8603 | 020406 | 001766 | | | BEQ 1\$ | ;KEEP TRYING |
| 8604 | 020410 | 005303 | | | DEC R3 | ;ADJ SCR1 POINTER |
| 8605 | 020412 | 005302 | | | DEC R2 | ;ADJ SCR2 POINTER |
| 8606 | 020414 | 160301 | | ECMOUT: | SUB R3,R1 | ;NO. OF CHARS. LEFT IN SCR2 |
| 8607 | 020416 | 160200 | | | SUB R2,R0 | ;NO. OF CHARS. LEFT IN SCR1 |
| 8608 | 020420 | 013704 | 017114 | EMATOT: | MOV EORSTK,R4 | ;REGISTER DUMP POINTER |
| 8609 | 020424 | 010024 | | | MOV R0,(R4)+ | ;R0 = SCR1 LEN |
| 8610 | 020426 | 010224 | | | MOV R2,(R4)+ | ;R1 = SCR1 START |
| 8611 | 020430 | 010124 | | | MOV R1,(R4)+ | ;R2 = SCR2 LEN |
| 8612 | 020432 | 010324 | | | MOV R3,(R4)+ | ;R3 = SCR2 START |
| 8613 | 020434 | 013724 | 017130 | | MOV E4,(R4)+ | ;R4 = R4 |
| 8614 | 020440 | 013724 | 017132 | | MOV E5,(R4)+ | ;R5 = R5 |
| 8615 | 020444 | 000207 | | | RTS PC | |
| 8616 | 020446 | 020103 | | ENDDA: | CMP R1,R3 | ;END SCR2 ? |
| 8617 | 020450 | 001004 | | | BNE 1\$ | |
| 8618 | 020452 | 013777 | 177776 | 176436 | MOV EPSW,@EOPSW | ;YES - STORE STATUS |
| 8619 | 020460 | 000755 | | | BR ECMOUT | |
| 8620 | 020462 | 123713 | 017130 | 1\$: | CMPB E4,(R3) | ;SET CC BITS |
| 8621 | 020466 | 013777 | 177776 | 176422 | MOV EPSW,@EOPSW | ;STORE RESULT |
| 8622 | 020474 | 123723 | 017130 | | CMPB E4,(R3)+ | ;FIND NON MATCHING CHARS. |
| 8623 | 020500 | 001762 | | | BEQ ENDA | ;KEEP TRYING |
| 8624 | 020502 | 005303 | | | DEC R3 | ;ADJ SCR2 POINTER |
| 8625 | 020504 | 000743 | | | BR ECMOUT | |
| 8626 | 020506 | 020002 | | ENDB: | CMP R0,R2 | ;END SCR1 |
| 8627 | 020510 | 001741 | | | BEQ ECMOUT | ;YES |
| 8628 | 020512 | 121237 | 017130 | | CMPB (R2),E4 | ;SET CC BITS |
| 8629 | 020516 | 013777 | 177776 | 176372 | MOV EPSW,@EOPSW | ;SAVE RESULT |
| 8630 | 020524 | 123722 | 017130 | | CMPB E4,(R2)+ | ;FIND NON MATCHING CHARS. |
| 8631 | 020530 | 001766 | | | BEQ ENDB | ;KEEP TRYING |
| 8632 | 020532 | 005302 | | | DEC R2 | ;ADJ POINTER |
| 8633 | 020534 | 000727 | | | BR ECMOUT | |

| | | | | | | | | |
|------|--------|--------|--------|--------|-----------------------|--|-----------------------|--------|
| 8635 | | | | | .SBTTL | | MATCH STRING | |
| 8636 | 020536 | 013700 | 017120 | | MOV E0,R0 | | ;CALCULATE END OF SRC | |
| 8637 | 020542 | 063700 | 017122 | | ADD E1,R0 | | | |
| 8638 | 020546 | 013701 | 017124 | | MOV E2,R1 | | ;CALCULATE END OF OBJ | |
| 8639 | 020552 | 063701 | 017126 | | ADD E3,R1 | | | |
| 8640 | 020556 | 013702 | 017122 | | MOV E1,R2 | | ;START OF SRC | |
| 8641 | 020562 | 013703 | 017126 | | MOV E3,R3 | | ;START OF OBJ | |
| 8642 | 020566 | 010204 | | 4\$: | MOV R2,R4 | | ;SAVE START OF SRC | |
| 8643 | 020570 | 020103 | | 1\$: | CMP R1,R3 | | ;OBJ FOUND IN STRING? | |
| 8644 | 020572 | 001411 | | | BEQ 3\$ | | ;YES | |
| 8645 | 020574 | 020200 | | | CMP R2, 020632 005700 | | | TST R0 |
| 8659 | 020634 | 013777 | 177776 | 176254 | MOV EPSW,@EPSW | | ;SAVE RESULT | |
| 8660 | 020642 | 000137 | 020420 | | JMP EMATOT | | | |

| | | | |
|------|--------|--------|---------------|
| 8662 | | | |
| 8663 | 020646 | 013700 | 017122 |
| 8664 | 020652 | 063700 | 017120 |
| 8665 | 020656 | 013701 | 017122 |
| 8666 | 020662 | 020100 | |
| 8667 | 020664 | 001476 | |
| 8668 | 020666 | 112137 | 017136 |
| 8669 | 020672 | 105037 | 017137 |
| 8670 | 020676 | 063737 | 017132 017136 |
| 8671 | 020704 | 117702 | 176226 |
| 8672 | 020710 | 133702 | 017130 |
| 8673 | 020714 | 001362 | |
| 8674 | 020716 | 005301 | |
| 8675 | 020720 | 000137 | 021062 |

ESPNC:

1\$:

017136

```
.SBTTL
MOV E1,R0
ADD E0,R0
MOV E1,R1
CMP R1,R0
BEQ ESPND
MOVB (R1)+,TEMP
CLRB TEMP+1
ADD E5,TEMP
MOVB @TEMP,R2
BITB E4,R2
BNE 1$
DEC R1
JMP ESPND
```

```
SEARCH FOR NON GROUP CHARS.
;CALCULATE END OF SOURCE
;
;START OF SOURCE
;COMPLETE
;YES
;TABLE OFFSET
;LOWER BYTE ONLY
;TABLE DATA
;AND MASK
;IF = 0 END SEARCH
;ADJ POINTER
```

| | | | | | | |
|------|--------|--------|--------|--------|-----------------|--------------------------|
| 8677 | | | | | .SBTTL | SEARCH FOR GROUP CHARS. |
| 8678 | 020724 | 013700 | 017122 | | MOV E1,R0 | ;CALCULATE END OF SOURCE |
| 8679 | 020730 | 063700 | 017120 | | ADD E0,R0 | |
| 8680 | 020734 | 013701 | 017122 | | MOV E1,R1 | ;START OF SOURCE |
| 8681 | 020740 | 020100 | | | CMP R1,R0 | ;COMPLETE ? |
| 8682 | 020742 | 001447 | | | BEQ ESPND | ;YES |
| 8683 | 020744 | 112137 | 017136 | | MOVB (R1)+,TEMP | ;TABLE OFFSET |
| 8684 | 020750 | 105037 | 017137 | | CLRB TEMP+1 | ;LOWER BYTE ONLY |
| 8685 | 020754 | 063737 | 017132 | 017136 | ADD E5,TEMP | ;TABLE DATA |
| 8686 | 020762 | 117702 | 176150 | | MOVB @TEMP,R2 | |
| 8687 | 020766 | 135702 | 017130 | | BITB E4,R2 | ;AND MASK |
| 8688 | 020772 | 001762 | | | BEQ 1\$ | ;IF <>0 END SEARCH |
| 8689 | 020774 | 005301 | | | DEC R1 | ;ADJ POINTER |
| 8690 | 020776 | 000137 | 021062 | | JMP ESPND | |

| | | | |
|------|--------|--------|--------|
| 8692 | | | |
| 8693 | 021002 | 013700 | 017122 |
| 8694 | 021006 | 010001 | |
| 8695 | 021010 | 063700 | 017120 |
| 8696 | 021014 | 020001 | |
| 8697 | 021016 | 001421 | |
| 8698 | 021020 | 122137 | 017130 |
| 8699 | 021024 | 001373 | |
| 8700 | 021026 | 005301 | |
| 8701 | 021030 | 000137 | 021062 |

ELOCC:

1\$:

```
.SBTTL  
MOV E1,R0  
MOV R0,R1  
ADD E0,R0  
CMP R0,R1  
BEQ ESPND  
CMPB (R1)+,E4  
BNE 1$  
DEC R1  
JMP ESPND
```

```
LOCC INSTRUCTION  
;CALCULATE END ADDRESS  
;SEARCH POINTER  
  
;HAS SEARCH FAILED  
;YES  
;LOOK FOR CHAR.  
;NOT FOUND YET  
;ADJUST POINTER  
;RETURN
```

| | | | | | | |
|------|--------|--------|---------------|--------|-----------------|------------------------|
| 8703 | | | | | .SBTTL | SKPC INSTRUCTION |
| 8704 | 021034 | 013700 | 017122 | ESKPC: | MOV E1,R0 | :CALCULATE END ADDRESS |
| 8705 | 021040 | 010001 | | | MOV R0,R1 | :START ADDRESS |
| 8706 | 021042 | 063700 | 017120 | | ADD E0,R0 | |
| 8707 | 021046 | 020001 | | 1\$: | CMP R0,R1 | :SEARCH FOR PASS |
| 8708 | 021050 | 001404 | | | BEQ ESPND | :FOUND IT |
| 8709 | 021052 | 122137 | 017130 | | CMPB (R1)+,E4 | :LOOK FOR NOT CHAR. |
| 8710 | 021056 | 001773 | | | BEQ 1\$ | |
| 8711 | 021060 | 005301 | | | DEC R1 | |
| 8712 | 021062 | 160100 | | ESPND: | SUB R1,R0 | :NO. OF CHARS. |
| 8713 | 021064 | 013702 | 017114 | | MOV EORSTK,R2 | :REGISTER DUMP POINTER |
| 8714 | 021070 | 010022 | | | MOV R0,(R2)+ | :R0 = LEN |
| 8715 | 021072 | 010122 | | | MOV R1,(R2)+ | :R1 = POS. |
| 8716 | 021074 | 013722 | 017124 | | MOV E2,(R2)+ | :R2 = R2 |
| 8717 | 021100 | 013722 | 017126 | | MOV E3,(R2)+ | :R3 = R3 |
| 8718 | 021104 | 013722 | 017130 | | MOV E4,(R2)+ | :R4 = R4 |
| 8719 | 021110 | 013722 | 017132 | | MOV E5,(R2)+ | :R5 = R5 |
| 8720 | 021114 | 005700 | | | TST R0 | :SET CC BITS |
| 8721 | 021116 | 013777 | 177776 175772 | | MOV EPSW,@EOPSW | :STORE RESULTS |
| 8722 | 021124 | 000207 | | | RTS PC | |

.SBTTL DIGIT (SIGN) RETRIEVER

```

*****
ROUTINE TO RETREIVE A PACKED OR ZONED DIGIT (OR SIGN) FROM A DECIMAL STRING.

INPUTS: R0 = STRING ADDRESS
        R1 = TYPE & POSITION OF DIGIT REQUESTED WITHIN STRING
        ELSD = TYPE & STRING LENGTH
        EODD = ODD SIZE INDICATOR

OUTPUT: ERSNEG = SIGN (IS SIGN WAS REQUESTED)
        = 0 IF REQUEST IS FOR UNSIGNED STRING SIGN.
        R2 = REQUESTED DIGIT (0 IF SIGN WAS REQUESTED)
        SGNBYT = SIGN BYTE IF SIGN WAS REQUESTED
        R0,R1,ELSD,EODD RETURNED UNDISTURBED

USAGE: (MSD=MOST SIGNIFICANT DIGIT;LSD=LEAST SIGNIF DIGIT)
        TO REQUEST MSD SET R1 = 0
        .. .. LSD SET R1 = STRING LENGTH -1
        .. .. SIGN SET R1 = STRING LENGTH
*****

```

```

8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746
8747 021126 005737 026056
8748 021132 100403
8749 021134 004737 021352
8750 021140 000402
8751 021142 004737 021150
8752 021146 000207
8753
8754
8755 021150 010037 031664
8756 021154 010137 031666
8757 021160 042701 070000
8758 021164 005737 024646
8759 021170 100401
8760 021172 005201
8761 021174 010137 024630
8762 021200 006237 024630
8763 021204 042737 000001 024630
8764 021212 060037 024630
8765 021216 013700 024630
8766 021222 032701 000002
8767 021226 001403
8768 021230 116002 000001
8769 021234 000401
8770 021236 111002
8771 021240 110237 024722
8772 021244 032701 000001
8773 021250 001433
8774 021252 042702 177760
8775 021256 123737 024664 031666
8776 021264 001020
8777 021266 005037 024616

```

```

ESNK:      TST EPAK           ;PACKED OR ZONED
           BMI 1$
           JSR PC,EFNDTZ     ;ZONED
           BR 2$
1$:        JSR PC,EFINDT     ;PACKED
2$:        RTS PC

;PACKED STRING DIGIT RETRIEVER
EFINDT:    MOV R0,ETMPRO     ;SAVE INPUT REGISTERS
           MOV R1,ETMPR1
           BIC #070000,R1   ;CLEAR TYPE FIELD
           TST EODD        ;POSITION CORRECT FOR ODD SIZE NUMBERS
           BMI 10$
           INC R1           ;NUMBER IS EVEN
10$:       MOV R1,EFINDA     ;FIND WORD TO NIBBLE
           ASR EFINDA
           BIC #1,EFINDA
           ADD R0,EFINDA    ;ADD OFFSET TO POSITION
           MOV EFINDA,R0   ;PLACE IN R0 FOR USE
           BIT #2,R1       ;WHICH BYTE
           BEQ 1$
           MOVB 1(R0),R2    ;GRAB BYTE
           BR 2$
1$:        MOVB (R0),R2     ;GRAB LOWER BYTE
2$:        MOVB R2,SGNBYT   ;SAVE SIGN BYTE FOR POSSIBLE ERROR PRINT
           BIT #1,R1       ;WHICH NIBBLE
           BEQ 3$         ;HIGH
4$:        BIC #177760,R2  ;LOW
           CMPB ELSD,ETMPR1 ;REQUEST FOR SIGN?
           BNE 5$         ;BRANCH IF NO
           CLR ERSNEG

```

| | | | | | | | |
|------|--------|--------|--------|--------|------|-------------------------------|-------------------------------------|
| 8778 | 021272 | 020227 | 000013 | | | CMP R2,#13 | :IS RETRIEVED SIGN NEG (1011)? |
| 8779 | 021276 | 001403 | | | | BEQ 7\$ | :BRANCH IF YES |
| 8780 | 021300 | 020227 | 000015 | | | CMP R2,#15 | :IS RETRIEVED SIGN NEG (1101)? |
| 8781 | 021304 | 001007 | | | | BNE 6\$ | :BRANCH IF NO |
| 8782 | 021306 | 122737 | 000160 | 031667 | 7\$: | CMPB #160,ETMPR1+1 | :IS INST UNSIGNED PACKED? |
| 8783 | 021314 | 001403 | | | | BEQ 6\$ | :BRANCH IF YES |
| 8784 | 021316 | 012737 | 177777 | 024616 | | MOV #177777,ERSNEG | :SET NEG FLAG |
| 8785 | 021324 | 005002 | | | 6\$: | CLR R2 | |
| 8786 | 021326 | 013700 | 031664 | | 5\$: | MOV ETMPRO,R0 | :RESTORE REGISTERS |
| 8787 | 021332 | 013701 | 031666 | | | MOV ETMPR1,R1 | |
| 8788 | 021336 | 000207 | | | | RTS PC | :RETURN |
| 8789 | 021340 | 006202 | | | 3\$: | ASR R2 | :SELECT UPPER NIBBLE |
| 8790 | 021342 | 006202 | | | | ASR R2 | |
| 8791 | 021344 | 006202 | | | | ASR R2 | |
| 8792 | 021346 | 006202 | | | | ASR R2 | |
| 8793 | 021350 | 000740 | | | | BR 4\$ | |
| 8794 | | | | | | | |
| 8795 | | | | | | | |
| 8796 | 021352 | 010037 | 031664 | | | :ZONED STRING DIGIT RETRIEVER | |
| 8797 | 021356 | 010137 | 031666 | | | EFNDTZ: MOV R0,ETMPRO | :SAVE REGISTER |
| 8798 | 021362 | 013737 | 024664 | 024656 | | MOV R1,ETMPR1 | |
| 8799 | 021370 | 042701 | 070000 | | | MOV ELSD,ESLSD | |
| 8800 | 021374 | 123737 | 024664 | 031666 | | BIC #070000,R1 | |
| 8801 | 021402 | 001442 | | | | CMPB ELSD,ETMPR1 | :REQUEST FOR SIGN? |
| 8802 | 021404 | 122737 | 000040 | 031667 | 6\$: | BEQ 1\$ | :BRANCH IF YES |
| 8803 | 021412 | 001405 | | | | CMPB #040,ETMPR1+1 | :IS DESC TYPE TRAILING OVERPUNCH? |
| 8804 | 021414 | 122737 | 000060 | 031667 | | BEQ 4\$ | :BRANCH IF YES |
| 8805 | 021422 | 001411 | | | | CMPB #060,ETMPR1+1 | :IS DESC TYPE LEADING OVERPUNCH? |
| 8806 | 021424 | 000415 | | | | BEQ 5\$ | :BRANCH IF YES |
| 8807 | | | | | | BR 3\$ | |
| 8808 | 021426 | 005337 | 024664 | | 4\$: | DEC ELSD | :TYPE = TRAILING OVERPUNCH |
| 8809 | 021432 | 123701 | 024664 | | | CMPB ELSD,R1 | :IS DIGIT REQUESTED PART OF THE |
| 8810 | | | | | | | : ENCODED SIGN DIGIT? |
| 8811 | 021436 | 001010 | | | | BNE 3\$ | :BRANCH IF NO |
| 8812 | 021440 | 004737 | 021764 | | | JSR PC,DECZO | :DECODE OVERPUNCH BYTE FOR DIGIT |
| 8813 | 021444 | 000407 | | | | BR 2\$ | |
| 8814 | | | | | | | |
| 8815 | 021446 | | | | 5\$: | | :TYPE = LEADING OVERPUNCH |
| 8816 | 021446 | 005701 | | | | TST R1 | :IS DIGIT REQUESTED PART OF ENCODED |
| 8817 | | | | | | | : SIGN DIGIT? |
| 8818 | 021450 | 001003 | | | | BNE 3\$ | :BRANCH IF NO |
| 8819 | 021452 | 004737 | 021764 | | | JSR PC,DECZO | :DECODE OVERPUNCH BYTE |
| 8820 | 021456 | 000402 | | | | BR 2\$ | |
| 8821 | | | | | | | |
| 8822 | 021460 | 060100 | | | 3\$: | ADD R1,R0 | :BYTE ADDRESS |
| 8823 | 021462 | 111002 | | | | MOVB (R0),R2 | :DATA |
| 8824 | 021464 | 042702 | 177760 | | 2\$: | BIC #177760,R2 | :MASK OFF JUNK |
| 8825 | 021470 | 013700 | 031664 | | | MOV ETMPRO,R0 | :RESTORE REGISTER |
| 8826 | 021474 | 013701 | 031666 | | | MOV ETMPR1,R1 | |
| 8827 | 021500 | 013737 | 024656 | 024664 | | MOV ESLSD,ELSD | |
| 8828 | 021506 | 000207 | | | | RTS PC | |
| 8829 | 021510 | | | | 1\$: | | :SIGN REQUESTED |
| 8830 | 021510 | 105737 | 024664 | | | TSTB ELSD | :IS STRING LEN = 0? |
| 8831 | 021514 | 001020 | | | | BNE 10\$ | :BRANCH IF NO |

| | | | | | | | |
|------|--------|--------|--------|--------|--------|--------------------|---|
| 8832 | 021516 | 122737 | 000100 | 031667 | | CMPB #100,ETMPR1+1 | ;TYPE = TRAILING SEPARATE? |
| 8833 | 021524 | 001002 | | | | BNE 103\$ | ;BRANCH IF NO |
| 8834 | 021526 | 000137 | 021710 | | | JMP 14\$ | ;RETURN SIGN AT 'A' |
| 8835 | 021532 | 122737 | 000120 | 031667 | 103\$: | CMPB #120,ETMPR1+1 | ;IS STRING TYPE LEADING SEPARATE? |
| 8836 | | | | | | | ;NOTE: SEPARATE TYPE ARE THE ONLY |
| 8837 | | | | | | | ; 0 LEN ZONED STRING THAT |
| 8838 | | | | | | | ; THAT OCCUPIES MEMORY. |
| 8839 | 021540 | 001002 | | | | BNE 102\$ | ;BRANCH IF NO |
| 8840 | 021542 | 000137 | 021736 | | | JMP 15\$ | ;RETURN SIGN AT 'A-1' |
| 8841 | 021546 | 005037 | 024722 | | 102\$: | CLR SGNBYT | ;CLEAR SIGN BYTE FOR POSSIBLE ERROR PRINT |
| 8842 | 021552 | 000137 | 021610 | | | JMP 100\$ | ;RETURN + SIGN |
| 8843 | | | | | | | |
| 8844 | 021556 | 122737 | 000000 | 031667 | 10\$: | CMPB #000,ETMPR1+1 | ;IS TYPE = SIGNED ZONED? |
| 8845 | 021564 | 001022 | | | | BNE 11\$ | ;BRANCH IF NO |
| 8846 | 021566 | 060100 | | | | ADD R1,R0 | ;FORM SIGN ADDRESS |
| 8847 | 021570 | 114002 | | | | MOV B -(R0),R2 | |
| 8848 | 021572 | 010237 | 024722 | | | MOV R2,SGNBYT | ;SAVE SIGN BYTE |
| 8849 | 021576 | 042702 | 177417 | | | BIC #177417,R2 | ;LOOK ONLY AT SIGN |
| 8850 | 021602 | 020227 | 000160 | | | CMP R2,#160 | ;IS IT (0111) NEGATIVE |
| 8851 | 021606 | 001404 | | | | BEQ 101\$ | ;BRANCH IF YES |
| 8852 | 021610 | 005037 | 024616 | | 100\$: | CLR ERSNEG | ;SET SIGN FLAG TO + |
| 8853 | 021614 | 005002 | | | | CLR R2 | |
| 8854 | 021616 | 000722 | | | | BR 2\$ | |
| 8855 | | | | | | | |
| 8856 | 021620 | 012737 | 177777 | 024616 | 101\$: | MOV #177777,ERSNEG | ;SET SIGN FLAG TO '-' |
| 8857 | 021626 | 005002 | | | | CLR R2 | |
| 8858 | 021630 | 000715 | | | | BR 2\$ | |
| 8859 | 021632 | 122737 | 000020 | 031667 | 11\$: | CMPB #020,ETMPR1+1 | ;IS TYPE = UNSIGNED ZONED? |
| 8860 | 021640 | 001004 | | | | BNE 12\$ | ;BRANCH IF NO |
| 8861 | 021642 | 060100 | | | | ADD R1,R0 | |
| 8862 | 021644 | 114037 | 024722 | | | MOV B -(R0),SGNBYT | ;SAVE SIGN BYTE |
| 8863 | 021650 | 000757 | | | | BR 100\$ | |
| 8864 | 021652 | 122737 | 000040 | 031667 | 12\$: | CMPB #040,ETMPR1+1 | ;IS TYPE = TRAILING OVERPUNCH? |
| 8865 | 021660 | 001005 | | | | BNE 13\$ | ;BRANCH IF NO |
| 8866 | 021662 | 005301 | | | | DEC R1 | |
| 8867 | 021664 | 004737 | 021764 | | 120\$: | JSR PC,DECZO | |
| 8868 | 021670 | 005002 | | | | CLR R2 | |
| 8869 | 021672 | 000674 | | | | BR 2\$ | |
| 8870 | | | | | | | |
| 8871 | 021674 | 122737 | 000060 | 031667 | 13\$: | CMPB #060,ETMPR1+1 | ;IS TYPE = LEADING OVERPUNCH? |
| 8872 | 021702 | 001002 | | | | BNE 14\$ | ;BRANCH IF NO |
| 8873 | 021704 | 005001 | | | | CLR R1 | |
| 8874 | 021706 | 000766 | | | | BR 120\$ | |
| 8875 | 021710 | 122737 | 000100 | 031667 | 14\$: | CMPB #100,ETMPR1+1 | ;IS TYPE = TRAILING SEPARATE |
| 8876 | 021716 | 001007 | | | | BNE 15\$ | ;BRANCH IF NO |
| 8877 | 021720 | 060100 | | | | ADD R1,R0 | ;FORM ADDRESS OF SIGN |
| 8878 | 021722 | 111037 | 024722 | | | MOV B (R0),SGNBYT | ;SAVE SIGN BYTE |
| 8879 | 021726 | 121027 | 000055 | | | CMPB (R0),#55 | ;IS SIGN = '-' |
| 8880 | 021732 | 001326 | | | | BNE 100\$ | ;BRANCH IF NO |
| 8881 | 021734 | 000731 | | | | BR 101\$ | |
| 8882 | 021736 | 122737 | 000120 | 031667 | 15\$: | CMPB #120,ETMPR1+1 | ;IS TYPE=LEADING SEPARATE |
| 8883 | 021744 | 001401 | | | | BEQ 115\$ | |
| 8884 | 021746 | 000000 | | | | HALT | ;ILLEGAL ZONED DATA TYPE |
| 8885 | 021750 | 114037 | 024722 | | 115\$: | MOV B -(R0),SGNBYT | |

```

8886 021754 121027 000055      CMPB (R0),#55      ;IS SIGN = '-' ?
8887 021760 001313              BNE 100$           ;BRANCH IF NO
8888 021762 000716              BR 101$
8889
8890
8891
8892      ;SUBROUTINE TO DECODE ZONED OVERPUNCH SIGN DIGIT BYTE
8893      ;
8894 021764 060100      DECZO:      ADD R1,R0      ;FIND DIGIT BY DECODING
8895      ;DIGIT RETURNED IN R2; SIGN IN ERSNEG
8896 021766 111002      MOV B (R0),R2    ;GET ENCODED BYTE
8897 021770 010237 024722      MOV R2,SGNBYT   ;SAVE SIGN BYTE FOR POSSIBLE ERROR PRINTOUT
8898 021774 042702 177417      BIC #177417,R2  ;LOOK AT HIGH NIBBLE
8899 022000 020227 000160      CMP R2,#160     ;IS HIGH NIBBLE A 7
8900 022004 001014      BNE 1$          ;BRANCH IF NO
8901 022006 111002      MOV B (R0),R2    ;DIGIT = 0
8902 022010 032702 000002      5$:      BIT #2,R2      ;IS SIGN + OR -
8903 022014 001403      BEQ 2$          ;BRANCH IF -
8904 022016 005037 024616      CLR ERSNEG
8905 022022 000403      BR 3$
8906 022024 012737 177777 024616      2$:      MOV #177777,ERSNEG
8907 022032 005002      3$:      CLR R2
8908 022034 000207      RTS PC
8909 022036 020227 000120      1$:      CMP R2,#120     ;IS HIGH NIBBLE A 5?
8910 022042 001014      BNE 4$          ;BRANCH IF NO
8911 022044 111002      MOV B (R0),R2
8912 022046 032702 000010      BIT #10,R2      ;IS DIGIT = 0?
8913 022052 001356      BNE 5$          ;BRANCH IF YES
8914 022054 012737 177777 024616      MOV #177777,ERSNEG ;DIGIT IS NEG
8915 022062 042702 177760      BIC #177760,R2
8916 022066 062702 000007      ADD #7,R2
8917 022072 000207      RTS PC
8918 022074 020227 000060      4$:      CMP R2,#060     ;IS HIGH NIBBLE A 3?
8919 022100 001006      BNE 6$          ;BRANCH IF NO
8920 022102 005037 024616      CLR ERSNEG      ;SIGN IS POSITIVE
8921 022106 111002      MOV B (R0),R2
8922 022110 042702 177760      BIC #177760,R2  ;DIGIT = LOW NIBBLE OF BYTE
8923 022114 000207      RTS PC
8924 022116 020227 000100      6$:      CMP R2,#100     ;IS HIGH NIBBLE A 4?
8925 022122 001404      BEQ 61$
8926 022124 005737 002300      TST PRTSGN     ;DECIMAL PRINTING IN PROGRESS?
8927 022130 001016      BNE 62$        ;BRANCH IF YES (DONT WANT TO HALT IN
8928      ; MIDDLE OF ERROR PRINTOUT)
8929      ;ILLEGAL ENCODING OF OVERPUNCH DIGIT
8929 022132 000000      HALT
8930 022134 111002      61$:      MOV B (R0),R2
8931 022136 042702 177760      BIC #177760,R2
8932 022142 020227 000012      CMP R2,#12     ;IS DIGIT POSITIVE?
8933 022146 103407      BLO 62$        ;BRANCH IF YES
8934 022150 012737 177777 024616      MOV #177777,ERSNEG
8935 022156 042702 000010      BIC #10,R2
8936 022162 005302      DEC R2
8937 022164 000207      RTS PC
8938 022166 005037 024616      62$:      CLR ERSNEG
8939 022172 000207      RTS PC

```

8941
8942
8943
8944
8945
8946
8947
8948
8949
8950
8951
8952
8953
8954
8955
8956
8957
8958
8959
8960
8961
8962
8963
8964 022174 005737 026056
8965 022200 100403
8966 022202 004737 022452
8967 022206 000402
8968 022210 004737 022216
8969 022214 000207
8970
8971
8972 022216 010037 031664
8973 022222 010137 031666
8974 022226 042701 070000
8975 022232 010237 031670
8976 022236 005737 024646
8977 022242 100401
8978 022244 005201
8979 022246 010137 024630
8980 022252 006237 024630
8981 022256 063700 024630
8982 022262 042702 177760
8983 022266 123737 024664 031666
8984 022274 001021
8985 022276 122737 000160 031667
8986 022304 001005
8987 022306 112702 000017
8988 022312 005037 024616
8989 022316 000410
8990 022320 005737 024616
8991 022324 001403
8992 022326 112702 000015
8993 022332 000402
8994 022334 112702 000014

```

.SBTTL                                DIGIT (SIGN) PUSHER
*****
ROUTINE TO PUSH A PACKED OR ZONED DIGIT (OR SIGN) ONTO A DECIMAL STRING

INPUTS: R0 = STRING ADDRESS
        R1 = IYPE & POSITION OF WHERE TO STORE DIGIT IN STRING
        ELSD = TYPE & STRING LEN
        EODD = ODD SIZE INDICATOR
        ERSNEG = SIGN IF SIGN IS TO BE STORED
                (EXCEPT 1111 IS ALWAYS STORED FOR UNSIGNED PACKED STRINGS)
        R2 = DIGIT TO PUSH INTO STRING

OUTPUT: ERSNEG = 0 FOR UNSIGNED STRINGS SIGN
        R0,R1,ELSD,EODD,R2 RETURNED UNDISTURBED

USAGE:  TO PUSH MSD  SET R1 = 0
        .. .. LSD   SET R1 = STRING LEN - 1
        .. .. SIGN  SET R1 = STRING LEN
        IF STRING LEN = 0 THEN STORED BYTE = 0,SIGN
*****

EPUSH:      TST EPAK                    ;ZONED OR PACKED STRING?
            BMI 1$
            JSR PC,EPUDTZ                ;ZONED
            BR 2$
            1$: JSR PC,EPUTDT             ;PACKED
            2$: RTS PC

;PACKED STRING NIBBLE PUSHER
EPUTDT:     MOV R0,ETMPRO                 ;SAVE REGISTERS
            MOV R1,ETMPR1
            BIC #070000,R1                ;CLEAR TYPE FIELD
            MOV R2,ETMPR2
            TST EODD                       ;POSITION CORRECT FOR ODD SIZE NUMBERS
            BMI 10$
            INC R1
            10$: MOV R1,EFINDA             ;FIND WORD TO NIBBLE
            ASR EFINDA
            ADD EFINDA,R0
            BIC #177760,R2                ;MASK JUNK FROM DATA
            CMPB ELSD,ETMPR1              ;REQUEST TO INSERT SIGN?
            BNE 5$                          ;BRANCH IF NO
            CMPB #160,ETMPR1+1            ;UNSIGNED INST?
            BNE 7$                          ;BRANCH IF NO
            MOVB #17,R2                    ;YES - STORE (1111)
            CLR ERSNEG                     ;SET SIGN FLAG TO POSITIVE
            BR 5$
            7$: TST ERSNEG                 ;STORE + SIGN?
            BEQ 6$                          ;BRANCH IF YES
            MOVB #15,R2                    ;STORE MINUS SIGN
            BR 5$
            6$: MOVB #14,R2                ;STORE POSITIVE SIGN

```

| | | | | | | | |
|------|--------|--------|--------|--------|--------|----------------------------|--|
| 8995 | 022340 | 010237 | 017136 | | 5\$: | MOV R2,TEMP | :HOLD DATA |
| 8996 | 022344 | 012737 | 000017 | 024642 | 4\$: | MOV #17,EMASK | :LOW NIBBLE MASK |
| 8997 | 022352 | 032701 | 000001 | | | BIT #1,R1 | :WHICH NIBBLE |
| 8998 | 022356 | 001013 | | | | BNE 2\$ | :LOW |
| 8999 | 022360 | 006337 | 017136 | | | ASL TEMP | :POSITION FOR HIGH NIBBLE |
| 9000 | 022364 | 006337 | 017136 | | | ASL TEMP | |
| 9001 | 022370 | 006337 | 017136 | | | ASL TEMP | |
| 9002 | 022374 | 006337 | 017136 | | | ASL TEMP | |
| 9003 | 022400 | 012737 | 000360 | 024642 | | MOV #360,EMASK | :NIBBLE MASK |
| 9004 | 022406 | 143710 | 024642 | | 2\$: | BICB EMASK,(R0) | :CLEAR NIBBLE BEFORE LOAD |
| 9005 | 022412 | 153710 | 017136 | | | BISB TEMP,(R0) | :LOAD NIBBLE |
| 9006 | 022416 | 005737 | 024646 | | | TST EODD | :IF NUMBER IS EVEN LENGTH |
| 9007 | 022422 | 100404 | | | | BMI 3\$ | |
| 9008 | 022424 | 005037 | 017136 | | | CLR TEMP | :AND THIS IS LAST DIGIT |
| 9009 | 022430 | 005301 | | | | DEC R1 | |
| 9010 | 022432 | 001744 | | | | BEQ 4\$ | :THEN CLEAR UNUSED NIBBLE |
| 9011 | 022434 | 013700 | 031664 | | 3\$: | MOV ETMPRO,R0 | :RESTORE REGISTERS |
| 9012 | 022440 | 013701 | 031666 | | | MOV ETMPR1,R1 | |
| 9013 | 022444 | 013702 | 031670 | | | MOV ETMPR2,R2 | |
| 9014 | 022450 | 000207 | | | | RTS PC | |
| 9015 | | | | | | | |
| 9016 | | | | | | | |
| 9017 | 022452 | 010037 | 031664 | | | :ZONED STRING DIGIT PUSHER | |
| 9018 | 022456 | 010137 | 031666 | | | EPUDTZ: MOV R0,ETMPRO | :SAVE REGISTER |
| 9019 | 022462 | 042701 | 070000 | | | MOV R1,ETMPR1 | |
| 9020 | 022466 | 010237 | 031670 | | | BIC #070000,R1 | |
| 9021 | 022472 | 042702 | 177760 | | | MOV R2,ETMPR2 | |
| 9022 | 022476 | 052702 | 000060 | | | BIC #177760,R2 | :MASK OFF JUNK |
| 9023 | 022502 | 123737 | 024664 | 031666 | | BIS #60,R2 | :ADD JUNK |
| 9024 | 022510 | 001411 | | | | CMPB ELSD,ETMPR1 | :REQUEST TO PUSH SIGN? |
| 9025 | 022512 | 060100 | | | | BEQ 1\$ | :BRANCH IF YES |
| 9026 | 022514 | 110210 | | | | ADD R1,R0 | :WORD ADDRESS |
| 9027 | 022516 | 013700 | 031664 | | 2\$: | MOVB R2,(R0) | :DEPOSIT DATA |
| 9028 | 022522 | 013701 | 031666 | | | MOV ETMPRO,R0 | :RESTORE REGISTER |
| 9029 | 022526 | 013702 | 031670 | | | MOV ETMPR1,R1 | |
| 9030 | 022532 | 000207 | | | | MOV ETMPR2,R2 | |
| 9031 | 022534 | 105737 | 024664 | | 1\$: | RTS PC | |
| 9032 | 022540 | 001015 | | | | TSTB ELSD | :IS STRING 0 IN LEN |
| 9033 | 022542 | 122737 | 000100 | 031667 | | BNE 3\$ | :BRANCH IF NO |
| 9034 | 022550 | 001001 | | | | CMPB #100,ETMPR1+1 | :IS SIGN = TRAILING SEPARATE |
| 9035 | 022552 | 000472 | | | | BNE 100\$ | |
| 9036 | 022554 | 122737 | 000120 | 031667 | 100\$: | BR 14\$ | :PUSH TRAILING SEP SIGN INTO 'A' |
| 9037 | 022562 | 001001 | | | | CMPB #120,ETMPR1+1 | :IS STRING TYPE = LEADING SEPARATE |
| 9038 | 022564 | 000503 | | | | BNE 4\$ | :BRANCH IF NO |
| 9039 | 022566 | 005037 | 024616 | | 4\$: | BR 16\$ | :PUSH LEADING SEPARATE SIGN INTO 'A-1' |
| 9040 | 022572 | 000751 | | | | CLR ERSNEG | |
| 9041 | 022574 | 122737 | 000000 | 031667 | 3\$: | BR 2\$ | |
| 9042 | 022602 | 001021 | | | | CMPB #000,ETMPR1+1 | :IS TYPE=SIGNED ZONED |
| 9043 | 022604 | 005737 | 024616 | | | BNE 5\$ | :BRANCH IF NO |
| 9044 | 022610 | 001003 | | | | TST ERSNEG | :WHAT SIGN IS TO BE STORED |
| 9045 | 022612 | 112702 | 000003 | | | BNE 6\$ | |
| 9046 | 022616 | 000402 | | | | MOVB #3,R2 | :STORE + SIGN |
| 9047 | 022620 | 112702 | 000007 | | 6\$: | BR 7\$ | :STORE - SIGN |
| 9048 | 022624 | 060100 | | | 7\$: | MOVB #7,R2 | :FORM SIGN BYTE ADDRESS |
| | | | | | | ADD R1,R0 | |

| | | | | | | |
|------|--------|--------|--------|--------|--------------------------|--|
| 9049 | 022626 | 142740 | 000360 | | BICB #360,-(R0) | :CLEAR POSITION FOR SIGN |
| 9050 | 022632 | 006302 | | | ASL R2 | :SHIFT SIGN INTO HIGH NIBBLE POSITION |
| 9051 | 022634 | 006302 | | | ASL R2 | |
| 9052 | 022636 | 006302 | | | ASL R2 | |
| 9053 | 022640 | 006302 | | | ASL R2 | |
| 9054 | 022642 | 150210 | | | BISB R2,(R0) | :INSERT SIGN |
| 9055 | 022644 | 000724 | | | BR 2\$ | |
| 9056 | 022646 | 122737 | 000020 | 031667 | 5\$: CMPB #020,ETMPR1+1 | :IS TYPE = UNSIGNED ZONED |
| 9057 | 022654 | 001001 | | | BNE 10\$ | :BRANCH IF NO |
| 9058 | 022656 | 000743 | | | BR 4\$ | :SIGN = 3 |
| 9059 | 022660 | 122737 | 000040 | 031667 | 10\$: CMPB #040,ETMPR1+1 | :IS TYPE = TRAILING OVERPUNCHED? |
| 9060 | 022666 | 001016 | | | BNE 11\$ | |
| 9061 | 022670 | 060100 | | | ADD R1,R0 | |
| 9062 | 022672 | 114002 | | | 15\$: MOVB -(R0),R2 | :PICKUP DIGIT TO BE ENCODED WITH SIGN |
| 9063 | 022674 | 042702 | 000060 | | BIC #60,R2 | :STRIP JUNK THAT WAS PREVIOUSLY ADDED. |
| 9064 | 022700 | 005737 | 024616 | | TST ERSNEG | :IS SIGN NEGATIVE? |
| 9065 | 022704 | 001004 | | | BNE 12\$ | :BRANCH IF YES |
| 9066 | 022706 | 062702 | 024676 | | ADD #OPEPTB,R2 | :ENCODE DIGIT WITH + SIGN |
| 9067 | 022712 | 111210 | | | 13\$: MOVB (R2),(R0) | :STORE ENCODED BYTE IN STRING. |
| 9068 | 022714 | 000700 | | | BR 2\$ | |
| 9069 | | | | | | |
| 9070 | 022716 | 062702 | 024710 | | 12\$: ADD #OPENTB,R2 | :ENCODE DIGIT WITH - SIGN |
| 9071 | 022722 | 000773 | | | BR 13\$ | :STORE ENCODED BYTE IN STRING. |
| 9072 | 022724 | 122737 | 000060 | 031667 | 11\$: CMPB #060,ETMPR1+1 | :IS TYPE = LEADING OVERPUNCH? |
| 9073 | 022732 | 001002 | | | BNE 14\$ | :BRANCH IF NO |
| 9074 | 022734 | 005200 | | | INC R0 | :ADJUST R0 TO ENABLE USE |
| 9075 | | | | | | : OF TRAILING OVERPUNCH ROUTINE |
| 9076 | 022736 | 000755 | | | BR 15\$ | |
| 9077 | | | | | | |
| 9078 | 022740 | 122737 | 000100 | 031667 | 14\$: CMPB #100,ETMPR1+1 | :IS TYPE=TRAILING SEPARATE? |
| 9079 | 022746 | 001012 | | | BNE 16\$ | :BRANCH IF NO |
| 9080 | 022750 | 060100 | | | ADD R1,R0 | |
| 9081 | 022752 | 005737 | 024616 | | 21\$: TST ERSNEG | :IS SIGN NEGATIVE? |
| 9082 | 022756 | 001003 | | | BNE 17\$ | :BRANCH IF YES |
| 9083 | 022760 | 112710 | 000053 | | MOVB #053,(R0) | :STORE + SIGN |
| 9084 | 022764 | 000654 | | | BR 2\$ | |
| 9085 | 022766 | 112710 | 000055 | | 17\$: MOVB #055,(R0) | :STORE - SIGN |
| 9086 | 022772 | 000651 | | | BR 2\$ | |
| 9087 | | | | | | |
| 9088 | 022774 | 122737 | 000120 | 031667 | 16\$: CMPB #120,ETMPR1+1 | :IS TYPE = LEADING SEPARATE? |
| 9089 | 023002 | 001401 | | | BEQ 20\$ | :BRANCH IF YES |
| 9090 | 023004 | 000000 | | | HALT | :ILLEGAL ZONED DATA TYPE |
| 9091 | 023006 | 005300 | | | 20\$: DEC R0 | |
| 9092 | 023010 | 000760 | | | BR 21\$ | |
| 9093 | | | | | | |

| | | | | | | | |
|------|--------|--------|--------|--------|--------|--------------------|--------------------------|
| 9095 | | | | | | .SBTTL | DECIMAL ADDER |
| 9096 | 023012 | 063737 | 024624 | 024626 | EDCAD: | ADD EDCOPA,EDCOPB | :ADD TWO SOURCE DIGITS |
| 9097 | 023020 | 005737 | 024622 | | | TST ECARRY | :ANY CARRY OR BORROW |
| 9098 | 023024 | 001410 | | | | BEQ 1\$ | :NO |
| 9099 | 023026 | 105737 | 024623 | | | TSTB ECARRY+1 | :WHICH ONE |
| 9100 | 023032 | 001403 | | | | BEQ 2\$ | |
| 9101 | 023034 | 005337 | 024626 | | | DEC EDCOPB | :BORROW |
| 9102 | 023040 | 000402 | | | | BR 1\$ | |
| 9103 | 023042 | 005237 | 024626 | | 2\$: | INC EDCOPB | :CARRY |
| 9104 | 023046 | 005037 | 024622 | | 1\$: | CLR ECARRY | :RESET CARRY/BORROW FLAG |
| 9105 | 023052 | 005737 | 024626 | | | TST EDCOPB | :IS RESULT NEGATIVE |
| 9106 | 023056 | 100006 | | | | BPL 3\$ | :NO |
| 9107 | 023060 | 062737 | 000012 | 024626 | | ADD #12,EDCOPB | :MAKE VALUE POSITIVE |
| 9108 | 023066 | 112737 | 000377 | 024623 | | MOVB #377,ECARRY+1 | :SET BORROW FLAG |
| 9109 | 023074 | 023727 | 024626 | 000011 | 3\$: | CMP EDCOPB,#11 | :IS RESULT > 9 |
| 9110 | 023102 | 101411 | | | | BLOS 4\$ | :NO |
| 9111 | 023104 | 062737 | 000006 | 024626 | | ADD #6,EDCOPB | :CONVERT TO DECIMAL |
| 9112 | 023112 | 042737 | 177760 | 024626 | | BIC #177760,EDCOPB | |
| 9113 | 023120 | 112737 | 000377 | 024622 | | MOVB #377,ECARRY | |
| 9114 | 023126 | 053737 | 024626 | 024640 | 4\$: | BIS EDCOPB,EADSUM | :RESULT = 0 INDICATOR |
| 9115 | 023134 | 000207 | | | | RTS PC | |
| 9116 | | | | | | | |

| | | | | | | | |
|------|--------|--------|--------|--------|---------|-------------------|-------------------------------------|
| 9118 | | | | | | .SBTTL | ADDP,ADDN,SUBP,SUBN INSTRUCTIONS |
| 9119 | 023136 | 012737 | 177777 | 026056 | EADDP: | MOV #177777,EPAK | :INDICATE PACKED MODE |
| 9120 | 023144 | 000430 | | | | BR EADSUB | |
| 9121 | 023146 | 005037 | 026056 | | EADDN: | CLR EPAK | :INDICATE ZONED MODE |
| 9122 | 023152 | 000425 | | | | BR EADSUB | |
| 9123 | 023154 | 012737 | 177777 | 026056 | ESUBP: | MOV #177777,EPAK | :INDICATE PACKED MODE |
| 9124 | 023162 | 012737 | 177777 | 024660 | | MOV #177777,ESUBF | :SET SUBTRACT FLAG |
| 9125 | 023170 | 005037 | 024646 | | | CLR EODD | |
| 9126 | 023174 | 032737 | 000001 | 017120 | | BIT #1,E0 | :IS NUMBER ODD LENGTH |
| 9127 | 023202 | 001411 | | | | BEQ EADSUB | :NO |
| 9128 | 023204 | 012737 | 177777 | 024646 | | MOV #177777,EODD | |
| 9129 | 023212 | 000405 | | | | BR EADSUB | |
| 9130 | 023214 | 005037 | 026056 | | ESUBN: | CLR EPAK | :INDICATE ZONED MODE |
| 9131 | 023220 | 012737 | 177777 | 024660 | | MOV #177777,ESUBF | :SET SUBTRACT FLAG |
| 9132 | 023226 | 005037 | 024650 | | EADSUB: | CLR EAODD | :ODD SIZE INDICATORS |
| 9133 | 023232 | 005037 | 024652 | | | CLR EBODD | |
| 9134 | 023236 | 005037 | 024654 | | | CLR ECODD | |
| 9135 | 023242 | 013737 | 017120 | 024670 | | MOV E0,TE0 | :SAVE STRING LEN WORDS |
| 9136 | 023250 | 013737 | 017124 | 024672 | | MOV E2,TE2 | |
| 9137 | 023256 | 013737 | 017130 | 024674 | | MOV E4,TE4 | |
| 9138 | 023264 | 032737 | 000001 | 017120 | | BIT #1,E0 | :IF ODD SIZE SET INDICATOR |
| 9139 | 023272 | 001403 | | | | BEQ 10\$ | :EVEN NO. OF DIGITS |
| 9140 | 023274 | 012737 | 177777 | 024650 | | MOV #177777,EAODD | :SHOW ITS ODD |
| 9141 | 023302 | 032737 | 000001 | 017124 | 10\$: | BIT #1,E2 | :IF ODD SIZE SET INDICATOR |
| 9142 | 023310 | 001403 | | | | BEQ 11\$ | :EVEN NO. OF DIGITS |
| 9143 | 023312 | 012737 | 177777 | 024652 | | MOV #177777,EBODD | :SHOW ITS ODD |
| 9144 | 023320 | 032737 | 000001 | 017130 | 11\$: | BIT #1,E4 | :IS RESULT ODD LENGTH |
| 9145 | 023326 | 001403 | | | | BEQ 12\$ | :NO |
| 9146 | 023330 | 012737 | 177777 | 024654 | | MOV #177777,ECODD | :SET ODD INDICATOR |
| 9147 | 023336 | 013700 | 017122 | | 12\$: | MOV E1,R0 | :FIND SIGN OF SRC1 |
| 9148 | 023342 | 013701 | 017120 | | | MOV E0,R1 | |
| 9149 | 023346 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 9150 | 023354 | 013737 | 024650 | 024646 | | MOV EAODD,EODD | |
| 9151 | 023362 | 004737 | 021126 | | | JSR PC,ESNK | |
| 9152 | 023366 | 005037 | 024606 | | | CLR ES1 | |
| 9153 | 023372 | 005737 | 024660 | | | TST ESUBF | :SUBTRACT INST? |
| 9154 | 023376 | 001007 | | | | BNE 1\$ | :BRANCH IF YES |
| 9155 | 023400 | 005737 | 024616 | | | TST ERSNEG | :IS SIGN NEGATIVE |
| 9156 | 023404 | 001407 | | | | BEQ 2\$ | :BRANCH IF NO |
| 9157 | 023406 | 012737 | 177777 | 024606 | 3\$: | MOV #177777,ES1 | :SET NEGATIVE INDICATOR |
| 9158 | 023414 | 000403 | | | | BR 2\$ | |
| 9159 | 023416 | 005737 | 024616 | | 1\$: | TST ERSNEG | :SUBT INST - IS SIGN NEGATIVE |
| 9160 | 023422 | 001771 | | | | BEQ 3\$ | :BRANCH IF NO TO SET NEG. INDICATOR |
| 9161 | | | | | | | :NOTE: FOR SUBT, THE SIGN OF SRC1 |
| 9162 | | | | | | | : IS INVERTED AND THE ADD IS USED. |
| 9163 | 023424 | 013700 | 017126 | | 2\$: | MOV E3,R0 | :FIND SIGN OF SRC2 |
| 9164 | 023430 | 013701 | 017124 | | | MOV E2,R1 | |
| 9165 | 023434 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 9166 | 023442 | 013737 | 024652 | 024646 | | MOV EBODD,EODD | |
| 9167 | 023450 | 004737 | 021126 | | | JSR PC,ESNK | |
| 9168 | 023454 | 005037 | 024610 | | | CLR ES2 | |
| 9169 | 023460 | 005737 | 024616 | | | TST ERSNEG | :IS SIGN NEGATIVE? |
| 9170 | 023464 | 001403 | | | | BEQ EADSB1 | :BRANCH IF NO |
| 9171 | 023466 | 012737 | 177777 | 024610 | | MOV #177777,ES2 | :SET NEGATIVE INDICATOR |

| | | | | | | | | | | |
|------|--------|--------|--------|--------|---------|-------------------|--|--|----------------------------------|--|
| 9172 | | | | | | | | | | |
| 9173 | 023474 | 005037 | 026066 | | EADSB1: | CLR EVTSSV | | | :RESET RESULT SIZE INDICATOR | |
| 9174 | 023500 | 005037 | 024612 | | | CLR EANEG | | | :RESET SUBTRACT FLAGS | |
| 9175 | 023504 | 005037 | 024614 | | | CLR EBNEG | | | | |
| 9176 | 023510 | 023737 | 024606 | 024610 | | CMP ES1,ES2 | | | :ADD OR SUBTRACT DISPATCH | |
| 9177 | 023516 | 001527 | | | | BEQ EADAD | | | :LIKE SIGNS ADD | |
| 9178 | 023520 | 113703 | 017124 | | | MOVB E2,R3 | | | :UNLIKE SIGNS SUBTRACT | |
| 9179 | 023524 | 123737 | 017120 | 017124 | | CMPB E0,E2 | | | :WHO IS LONGER | |
| 9180 | 023532 | 103402 | | | | BLO 1\$ | | | | |
| 9181 | 023534 | 113703 | 017120 | | | MOVB E0,R3 | | | :MAXIMUM LENGTH | |
| 9182 | 023540 | 113704 | 017120 | | 1\$: | MOVB E0,R4 | | | :POSITION OF SCR1 | |
| 9183 | 023544 | 113705 | 017124 | | | MOVB E2,R5 | | | :POSITION OF SCR2 | |
| 9184 | 023550 | 160304 | | | | SUB R3,R4 | | | :START POSITION | |
| 9185 | 023552 | 160305 | | | | SUB R3,R5 | | | | |
| 9186 | 023554 | 123704 | 017120 | | 6\$: | CMPB E0,R4 | | | :EXIT HERE INDICATES | |
| 9187 | 023560 | 001470 | | | | BEQ EAIS | | | :RESULT = 0 | |
| 9188 | 023562 | 005037 | 024602 | | | CLR EOPA | | | :RESET DATA REGS | |
| 9189 | 023566 | 005037 | 024604 | | | CLR EOPB | | | | |
| 9190 | 023572 | 005704 | | | | TST R4 | | | :IS POSITION OF SCR1 VALID | |
| 9191 | 023574 | 100424 | | | | BMI 2\$ | | | :NO | |
| 9192 | 023576 | 013700 | 017122 | | | MOV E1,R0 | | | :GIT A OPERAND | |
| 9193 | 023602 | 010401 | | | | MOV R4,R1 | | | | |
| 9194 | 023604 | 013737 | 017120 | 024664 | | MOV E0,ELSD | | | | |
| 9195 | 023612 | 013737 | 024650 | 024646 | | MOV EAODD,EODD | | | | |
| 9196 | 023620 | 013737 | 017120 | 024666 | | MOV E0,TEMPE | | | :SET TYPE FIELD | |
| 9197 | 023626 | 105037 | 024666 | | | CLRB TEMPE | | | | |
| 9198 | 023632 | 053701 | 024666 | | | BIS TEMPE,R1 | | | | |
| 9199 | 023636 | 004737 | 021126 | | | JSR PC,ESNK | | | | |
| 9200 | 023642 | 010237 | 024602 | | | MOV R2,EOPA | | | :DATA FOR COMPARE | |
| 9201 | 023646 | 005705 | | | 2\$: | TST R5 | | | :IS POSITION OF SCR2 VALID | |
| 9202 | 023650 | 100424 | | | | BMI 4\$ | | | :NO | |
| 9203 | 023652 | 013700 | 017126 | | | MOV E3,R0 | | | :GIT B OPERAND | |
| 9204 | 023656 | 010501 | | | | MOV R5,R1 | | | | |
| 9205 | 023660 | 013737 | 017124 | 024664 | | MOV E2,ELSD | | | | |
| 9206 | 023666 | 013737 | 024652 | 024646 | | MOV EBODD,EODD | | | | |
| 9207 | 023674 | 013737 | 017124 | 024666 | | MOV E2,TEMPE | | | :SET TYPE FIELD | |
| 9208 | 023702 | 105037 | 024666 | | | CLRB TEMPE | | | | |
| 9209 | 023706 | 053701 | 024666 | | | BIS TEMPE,R1 | | | | |
| 9210 | 023712 | 004737 | 021126 | | | JSR PC,ESNK | | | | |
| 9211 | 023716 | 010237 | 024604 | | | MOV R2,EOPB | | | :DATA FOR COMPARE | |
| 9212 | 023722 | 023737 | 024602 | 024604 | 4\$: | CMP EOPA,EOPB | | | :WHO IS LARGER | |
| 9213 | 023730 | 101004 | | | | BHI EAIS | | | :A IS | |
| 9214 | 023732 | 103412 | | | | BLO EBIS | | | :B IS | |
| 9215 | 023734 | 005204 | | | | INC R4 | | | :OH NO THEIR THE SAME | |
| 9216 | 023736 | 005205 | | | | INC R5 | | | | |
| 9217 | 023740 | 000705 | | | | BR 6\$ | | | | |
| 9218 | 023742 | 012737 | 177777 | 024614 | EAIS: | MOV #177777,EBNEG | | | :MAKE B NEGATIVE | |
| 9219 | 023750 | 013737 | 024606 | 024616 | | MOV ES1,ERSNEG | | | :IF A IS NEGATIVE THEN RESULT IS | |
| 9220 | 023756 | 000412 | | | | BR EADAD1 | | | | |
| 9221 | 023760 | 012737 | 177777 | 024612 | EBIS: | MOV #177777,EANEG | | | :MAKE A NEGATIVE | |
| 9222 | 023766 | 013737 | 024610 | 024616 | | MOV ES2,ERSNEG | | | :IF B IS NEGATIVE THEN RESULT IS | |
| 9223 | 023774 | 000403 | | | | BR EADAD1 | | | | |
| 9224 | 023776 | 013737 | 024606 | 024616 | EADAD: | MOV ES1,ERSNEG | | | :SIGN OF RESULT | |
| 9225 | 024004 | 005037 | 024640 | | EADAD1: | CLR EADSUM | | | :RESULT =0 INDICATOR | |

| | | | | | | |
|-----------------|-------------------|-------------------|-------------------|---------|---------------------|------------------------------------|
| 9226 | 024010 | 013737 | 024616 | 024662 | MOV ERSNEG,SAVSGN | :SAVE SIGN OF RESULT |
| 9227 | 024016 | 012700 | 025662 | | MOV #EVRTAB+^D40,R0 | :CLEAR DATA AREA |
| 9228 | 024022 | 005040 | | | 21\$: CLR -(R0) | |
| 9229 | 024024 | 020027 | 025612 | | CMP R0,#EVRTAB | |
| 9230 | 024030 | 001374 | | | BNE 21\$ | |
| 9231 | 024032 | 012703 | 025662 | | MOV #EVRTAB+^D40,R3 | :DEST. POINTER |
| 9232 | 024036 | 005037 | 024622 | | CLR ECARRY | :RESET CARRY |
| 9233 | 024042 | 005037 | 026062 | | CLR EVT PAS | :RESET PASS COUNTER |
| 9234 | 024046 | 005037 | 024620 | | 20\$: CLR ENOA | :NO A OPERAND FLAG |
| 9235 | 024052 | 105337 | 017120 | | DECB E0 | :A DIGIT POINTER |
| 9236 | 024056 | 100005 | | | BPL 1\$ | |
| 9237 | 024060 | 005237 | 024620 | | INC ENOA | :NO DIGITS LEFT |
| 9238 | 024064 | 005037 | 024624 | | CLR EDCOPA | :DIGIT = 0 |
| 9239 | 024070 | 000404 | | | BR 2\$ | |
| 9240 | 024072 | 004737 | 024176 | | 1\$: JSR PC,EGTOPA | :GIT A OPERAND |
| 9241 | 024076 | 010237 | 024624 | | MOV R2,EDCOPA | :SAVE VALUE |
| 9242 | 024102 | 105337 | 017124 | | 2\$: DECB E2 | :B DIGIT POINTER |
| 9243 | 024106 | 100006 | | | BPL 3\$ | |
| 9244 | 024110 | 005737 | 024620 | | TST ENOA | :NO DIGITS LEFT |
| 9245 | 024114 | 001077 | | | BNE EXT | |
| 9246 | 024116 | 005037 | 024626 | | CLR EDCOPB | |
| 9247 | 024122 | 000404 | | | BR 4\$ | |
| 9248 | 024124 | 004737 | 024230 | | 3\$: JSR PC,EGTOPB | :GIT B OPERAND |
| 9249 | 024130 | 010237 | 024626 | | MOV R2,EDCOPB | :SAVE VALUE |
| 9250 | 024134 | 005737 | 024612 | | 4\$: TST EANEG | :WANT A COMPLEMENTED |
| 9251 | 024140 | 100004 | | | BPL 5\$ | :NO |
| 9252 | 024142 | 005137 | 024624 | | COM EDCOPA | :YES |
| 9253 | 024146 | 005237 | 024624 | | INC EDCOPA | |
| 9254 | 024152 | 005737 | 024614 | | 5\$: TST EBNEG | :WANT B COMPLEMENTED |
| 9255 | 024156 | 100004 | | | BPL 6\$ | |
| 9256 | 024160 | 005137 | 024626 | | COM EDCOPB | :YES |
| 9257 | 024164 | 005237 | 024626 | | INC EDCOPB | |
| 9258 | 024170 | 004737 | 024262 | | 6\$: JSR PC,EADIT | :ADD DIGITS STORE RESULT |
| 9259 | 024174 | 000724 | | | BR 20\$ | |
| 9260 | | | | | | |
| 9261 | 024176 | | | EGTOPA: | | :SUBROUTINE TO GET 'A' OPERAND |
| 9262 | 024176 | 013700 | 017122 | | MOV E1,R0 | :START ADDRESS OF NUMBER |
| 9263 | 024202 | 013701 | 017120 | | MOV E0,R1 | :DIGIT OF NUMBER |
| 9264 | 024206 | 013737 | 024650 | 024646 | MOV EAODD,EODD | |
| 9265 | 024214 | 013737 | 024670 | 024664 | MOV TE0,ELSD | |
| 9266 | 024222 | 004737 | 021126 | | JSR PC,ESNK | :CALL ROUTINE TO RETRIEVE DIGIT |
| 9267 | 024226 | 000207 | | | RTS PC | |
| 9268 | | | | | | |
| 9269 | 024230 | | | EGTOPB: | | :SUBROUTINE TO GET 'B' OPERAND |
| 9270 | 024230 | 013737 | 024652 | 024646 | MOV EBODD,EODD | |
| 9271 | 024236 | 013700 | 017126 | | MOV E3,R0 | :START ADDRESS OF NUMBER |
| 9272 | 024242 | 013701 | 017124 | | MOV E2,R1 | :DIGIT OF NUMBER |
| 9273 | 024246 | 013737 | 024672 | 024664 | MOV TE2,ELSD | |
| 9274 | 024254 | 004737 | 021126 | | JSR PC,ESNK | :CALL ROUTINE TO RETRIEVE DIGIT |
| 9275 | 024260 | 000207 | | | RTS PC | |
| 9276 | | | | | | |
| 9277 | 024262 | 004737 | 023012 | EADIT: | JSR PC,EDCAD | :ADD TWO DECIMAL DIGITS |
| 9278 | 024266 | 005237 | 026062 | | INC EVT PAS | :BUMP PASS COUNTER |
| 9279 | 024272 | 005737 | 024626 | | TST EDCOPB | :SAVE POSITION OF LAST VALID DIGIT |

| | | | | | | | |
|------|--------|--------|--------|--------|-------|---------------------|--|
| 9280 | 024276 | 001403 | | | | BEQ 1\$ | |
| 9281 | 024300 | 013737 | 026062 | 026066 | | MOV EVTPAS,EVTSSV | :SAVE POSITION |
| 9282 | 024306 | 113743 | 024626 | | 1\$: | MOVB EDCOPB,-(R3) | :SAVE RESULT |
| 9283 | 024312 | 000207 | | | | RTS PC | |
| 9284 | 024314 | 005037 | 031702 | | EXT: | CLR ENZI | :INITIALIZE NONZERO DIGIT STORED INDICATOR TO ZE |
| 9285 | 024320 | 105737 | 024622 | | | TSTB ECARRY | :ANY CARRY FROM LAST ADD |
| 9286 | 024324 | 100007 | | | | BPL 1\$ | :NO |
| 9287 | 024326 | 112743 | 000001 | | | MOVB #1,-(R3) | :ADD CARRY TO WORD |
| 9288 | 024332 | 005237 | 026062 | | | INC EVTPAS | :BUMP PASS COUNTER |
| 9289 | 024336 | 013737 | 026062 | 026066 | | MOV EVTPAS,EVTSSV | :STORE POSITION |
| 9290 | 024344 | 012703 | 025662 | | 1\$: | MOV #EVRTAB+^D40,R3 | :ADDRESS OF DATA |
| 9291 | 024350 | 105337 | 017130 | | 2\$: | DECB E4 | :LAST TRANSFER COMPLETE |
| 9292 | 024354 | 100423 | | | | BMI 3\$ | :YES |
| 9293 | 024356 | 013700 | 017132 | | | MOV E5,R0 | :START ADDRESS OF DEST. |
| 9294 | 024362 | 013701 | 017130 | | | MOV E4,R1 | :DIGIT POSITION |
| 9295 | 024366 | 114302 | | | | MOVB -(R3),R2 | :DATA TO DEPOSIT |
| 9296 | 024370 | 005702 | | | | TST R2 | :CHECK DIGIT BEING STORED IN DST |
| 9297 | 024372 | 001403 | | | | BEQ 7\$ | |
| 9298 | 024374 | 012737 | 177777 | 031702 | | MOV #177777,ENZI | :DIGIT NOT = 0, SET INDICATOR |
| 9299 | 024402 | 013737 | 024654 | 024646 | 7\$: | MOV ECODD,EODD | |
| 9300 | 024410 | 013737 | 024674 | 024664 | | MOV TE4,ELSD | |
| 9301 | 024416 | 004737 | 022174 | | | JSR PC,EPUSH | :CALL ROUTINE TO PUSH DIGIT ONTO DST STRING |
| 9302 | 024422 | 000752 | | | | BR 2\$ | |
| 9303 | 024424 | 013701 | 024674 | | 3\$: | MOV TE4,R1 | :POSITION OF SIGN |
| 9304 | 024430 | 013737 | 024662 | 024616 | | MOV SAVSGN,ERSNEG | :SETUP ERSNEG WITH RESULT SIGN |
| 9305 | 024436 | 013737 | 024674 | 024664 | | MOV TE4,ELSD | |
| 9306 | 024444 | 005737 | 026066 | | | TST EVTSSV | :IF ZERO, SIGN = + |
| 9307 | 024450 | 001002 | | | | BNE 10\$ | |
| 9308 | 024452 | 005037 | 024616 | | | CLR ERSNEG | :SET POSITIVE |
| 9309 | 024456 | 013700 | 017132 | | 10\$: | MOV E5,R0 | :START OF DEST. |
| 9310 | 024462 | 013737 | 024654 | 024646 | | MOV ECODD,EODD | |
| 9311 | 024470 | 004737 | 022174 | | | JSR PC,EPUSH | :SAVE SIGN |
| 9312 | 024474 | 005077 | 172416 | | EXT1: | CLR @EOPSW | :RESET EMULATE PSW |
| 9313 | 024500 | 005737 | 031702 | | | TST ENZI | :IF = 0 SET Z BIT & SKIP OVER SETTING OF N BIT. |
| 9314 | 024504 | 001407 | | | | BEQ 1\$ | |
| 9315 | 024506 | 005737 | 024616 | | | TST ERSNEG | :IF (-) SET N BIT |
| 9316 | 024512 | 100007 | | | | BPL 2\$ | |
| 9317 | 024514 | 052777 | 000010 | 172374 | | BIS #10,@EOPSW | :SET N BIT |
| 9318 | 024522 | 000403 | | | | BR 2\$ | |
| 9319 | 024524 | 052777 | 000004 | 172364 | 1\$: | BIS #4,@EOPSW | :COMP. LENGTH FOR OVERFLOW |
| 9320 | 024532 | 123737 | 024674 | 026066 | 2\$: | CMPB TE4,EVTSSV | :FIND LENGTH OF RESULT |
| 9321 | 024540 | 103003 | | | | BHIS 4\$ | |
| 9322 | 024542 | 052777 | 000002 | 172346 | | BIS #2,@EOPSW | :SET OVERFLOW |
| 9323 | 024550 | 013702 | 017114 | | 4\$: | MOV EORSTK,R2 | :REGISTER UNLOAD |
| 9324 | 024554 | 005022 | | | | CLR (R2)+ | :R0 = 0 |
| 9325 | 024556 | 005022 | | | | CLR (R2)+ | :R1 = 0 |
| 9326 | 024560 | 005022 | | | | CLR (R2)+ | :R2 = 0 |
| 9327 | 024562 | 005022 | | | | CLR (R2)+ | :R3 = 0 |
| 9328 | 024564 | 013722 | 024674 | | | MOV TE4,(R2)+ | :R4 = R4 |
| 9329 | 024570 | 013722 | 017132 | | | MOV E5,(R2)+ | :R5 = R5 |
| 9330 | 024574 | 005037 | 024660 | | | CLR ESUBF | :CLEAR SUBTRACT FLAG |
| 9331 | 024600 | 000207 | | | | RTS PC | |

| | | | | | |
|------|--------|--------|---------|-------|-----|
| 9333 | 024602 | 000000 | EOPA: | .WORD | 0 |
| 9334 | 024604 | 000000 | EOPB: | .WORD | 0 |
| 9335 | 024606 | 000000 | ES1: | .WORD | 0 |
| 9336 | 024610 | 000000 | ES2: | .WORD | 0 |
| 9337 | 024612 | 000000 | EANEG: | .WORD | 0 |
| 9338 | 024614 | 000000 | EBNEG: | .WORD | 0 |
| 9339 | 024616 | 000000 | ERSNEG: | .WORD | 0 |
| 9340 | 024620 | 000000 | ENOA: | .WORD | 0 |
| 9341 | 024622 | 000000 | ECARRY: | .WORD | 0 |
| 9342 | 024624 | 000000 | EDCOPA: | .WORD | 0 |
| 9343 | 024626 | 000000 | EDCOPB: | .WORD | 0 |
| 9344 | 024630 | 000000 | EFINDA: | .WORD | 0 |
| 9345 | 024632 | 000000 | ESGNA: | .WORD | 0 |
| 9346 | 024634 | 000000 | ESGNB: | .WORD | 0 |
| 9347 | 024636 | 000000 | ESGNC: | .WORD | 0 |
| 9348 | 024640 | 000000 | EADSUM: | .WORD | 0 |
| 9349 | 024642 | 000000 | EMASK: | .WORD | 0 |
| 9350 | 024644 | 000000 | TEMP1: | .WORD | 0 |
| 9351 | 024646 | 000000 | EODD: | .WORD | 0 |
| 9352 | 024650 | 000000 | EAODD: | .WORD | 0 |
| 9353 | 024652 | 000000 | EBODD: | .WORD | 0 |
| 9354 | 024654 | 000000 | ECODD: | .WORD | 0 |
| 9355 | 024656 | 000000 | ESLSD: | .WORD | 0 |
| 9356 | 024660 | 000000 | ESUBF: | .WORD | 0 |
| 9357 | 024662 | 000000 | SAVSGN: | .WORD | 0 |
| 9358 | 024664 | 000000 | ELSD: | .WORD | 0 |
| 9359 | 024666 | 000000 | TEMPE: | .WORD | 0 |
| 9360 | 024670 | 000000 | TE0: | .WORD | 0 |
| 9361 | 024672 | 000000 | TE2: | .WORD | 0 |
| 9362 | 024674 | 000000 | TE4: | .WORD | 0 |
| 9363 | 024676 | | OPEPTB: | | |
| 9364 | 024676 | 173 | | .BYTE | 173 |
| 9365 | 024677 | 101 | | .BYTE | 101 |
| 9366 | 024700 | 102 | | .BYTE | 102 |
| 9367 | 024701 | 103 | | .BYTE | 103 |
| 9368 | 024702 | 104 | | .BYTE | 104 |
| 9369 | 024703 | 105 | | .BYTE | 105 |
| 9370 | 024704 | 106 | | .BYTE | 106 |
| 9371 | 024705 | 107 | | .BYTE | 107 |
| 9372 | 024706 | 110 | | .BYTE | 110 |
| 9373 | 024707 | 111 | | .BYTE | 111 |
| 9374 | 024710 | 175 | OPENTB: | .BYTE | 175 |
| 9375 | 024711 | 112 | | .BYTE | 112 |
| 9376 | 024712 | 113 | | .BYTE | 113 |
| 9377 | 024713 | 114 | | .BYTE | 114 |
| 9378 | 024714 | 115 | | .BYTE | 115 |
| 9379 | 024715 | 116 | | .BYTE | 116 |
| 9380 | 024716 | 117 | | .BYTE | 117 |
| 9381 | 024717 | 120 | | .BYTE | 120 |
| 9382 | 024720 | 121 | | .BYTE | 121 |
| 9383 | 024721 | 122 | | .BYTE | 122 |
| 9384 | 024722 | 000000 | SGNBYT: | .WORD | 0 |

:OVERPUNCH SIGN DIGIT (PREFERRED ENCODINGS)

:+0
:+1
:+2
:+3
:+4
:+5
:+6
:+7
:+8
:+9
:-0
:-1
:-2
:-3
:-4
:-5
:-6
:-7
:-8
:-9

```
9386          .SBTTL          CVTLP,Z INSTRUCTIONS
9387          :          CONVERT LONG FORMAT NUMBERS INTO DECIMAL
9388          :          IN EITHER PACKED OR ZONED FORMAT.
9389          -----
9390 024724 005037 026056  ECVTLN:          CLR EPAK          :ZONED FORMATED OUTPUT
9391 024730 000403          BR ECVT
9392 024732 012737 177777 026056  ECVTLP:          MOV #177777,EPAK          :PACKED FORMAT OUTPUT
9393 024740 005037 024646  ECVT:          CLR EODD          :RESET ODD INDICATORS
9394 024744 005037 026066          CLR EVTSSV
9395 024750 005037 024624          CLR EDCOPA
9396 024754 005037 024626          CLR EDCOPB
9397 024760 005037 026064          CLR ESCF
9398 024764 032737 000001 017120          BIT #1,E0          :IF ODD SET ODD FLAG
9399 024772 001403          BEQ 10$          :EVEN NO. OF DIGITS
9400 024774 012737 177777 024646          MOV #177777,EODD          :SET ODD FLAG
9401 025002 012700 025626          10$:          MOV #EVRTAB+14,R0          :CLEAR DATA TABLE
9402 025006 013702 017124          MOV E2,R2          :DATA TO BE CONVERTED
9403 025012 013703 017126          MOV E3,R3
9404 025016 005040          1$:          CLR -(R0)
9405 025020 022700 025612          CMP #EVRTAB,R0          :CLEAR COMPLETE
9406 025024 001374          BNE 1$          :NO
9407 025026 005037 026060          CLR EVTSGN          :CLEAR OLD SIGN INFO
9408 025032 005702          TST R2          :IS DATA NEGATIVE
9409 025034 100026          BPL 2$          :NO
9410 025036 112737 000377 026060          MOVB #377,EVTSGN          :YES , SAVE SIGN
9411 025044 005103          COM R3          :NEGATE DATA
9412 025046 005102          COM R2
9413 025050 022703 177777          CMP #177777,R3          :CHECK FOR SPECIAL CASE - MOST NEG #.
9414 025054 001007          BNE 6$          :BRANCH IF THIS IS NOT THE SPECIAL CASE
9415 025056 022702 077777          CMP #077777,R2          :FOR MOST NEG # SRC HIGH =100000, SRC LOW = 0.
9416 025062 001004          BNE 6$          :BRANCH IF NOT THE SPECIAL CASE
9417 025064 012737 177777 026064          MOV #177777,ESCF          :SET SPECIAL CASE FLAG
9418 025072 000405          BR 3$
9419 025074 000257          6$:          CCC
9420 025076 062703 000001          ADD #1,R3
9421 025102 103001          BCC 3$          :BRANCH IF NO CARRY FROM FIRST WORD
9422 025104 005202          INC R2          :YES
9423 025106 042702 100000          3$:          BIC #100000,R2          :CLEAR SIGN BIT
9424 025112 005037 026062          2$:          CLR EVTPAS          :RESET PASS COUNTER
9425 025116 023727 026062 000174          5$:          CMP EVTPAS,#^D124          :31 PASSES COMPLETE YET
9426 025124 001460          BEQ EVTWRP          :WRAP UP ROUTEEN
9427 025126 000257          CCC          :CLEAR CC BITS FOR 32 BIT SHIFT
9428 025130 006002          ROR R2          :SHIFT LOB INTO R3
9429 025132 006003          ROR R3          :SHIFT LOB FOR VALUE
9430 025134 103002          BCC 4$          :IF NOT SET BIT HAS NO VALUE
9431 025136 004737 025152          JSR PC,EVRTAD          :ADD BIT VALUE TO DECIMAL NUMBER
9432 025142 062737 000004 026062          4$:          ADD #4,EVTPAS          :BUMP PASS COUNTER
9433 025150 000762          BR 5$          :NEXT PASS
9434 025152 005037 024622          EVRTAD:          CLR ECARRY          :RESET CARRY/BORROW FLAGS
9435 025156 012700 025624          MOV #EVRTAB+12,R0          :A OPERAND POINTER
9436 025162 013704 026062          MOV EVTPAS,R4          :TABLE OFFSET
9437 025166 062704 026070          ADD #EVTABA,R4          :TABLE START ADDRESS
9438 025172 012401          MOV (R4)+,R1          :START ADDRESS OF DATA
9439 025174 061401          ADD (R4),R1          :END ADDRESS OF DATA
```


| | | | | | | |
|------|--------|--------|--------|---------|-------------------|--|
| 9440 | 025176 | 011404 | | | MOV (R4),R4 | :NO. OF DIGITS |
| 9441 | 025200 | 010437 | 026066 | | MOV R4,EVTSSV | |
| 9442 | 025204 | 114037 | 024624 | 1\$: | MOVB -(R0),EDCOPA | :FIND A OPERAND |
| 9443 | 025210 | 114137 | 024626 | | MOVB -(R1),EDCOPB | :FIND B OPERAND |
| 9444 | 025214 | 004737 | 023012 | | JSR PC,EDCAD | :ADD |
| 9445 | 025220 | 113710 | 024626 | | MOVB EDCOPB,(R0) | :SAVE RESULT |
| 9446 | 025224 | 005304 | | | DEC R4 | :IS THIS LAST DIGIT |
| 9447 | 025226 | 001366 | | | BNE 1\$ | :NO |
| 9448 | 025230 | 105737 | 024622 | 2\$: | TSTB ECARRY | :ANY CARRY |
| 9449 | 025234 | 100401 | | | BMI 3\$ | :YES |
| 9450 | 025236 | 000207 | | | RTS PC | |
| 9451 | 025240 | 005237 | 026066 | 3\$: | INC EVTSSV | |
| 9452 | 025244 | 114037 | 024624 | | MOVB -(R0),EDCOPA | :A OPERAND |
| 9453 | 025250 | 005037 | 024626 | | CLR EDCOPB | :NO B OPERAND |
| 9454 | 025254 | 004737 | 023012 | | JSR PC,EDCAD | :ADD THE CARRY |
| 9455 | 025260 | 113710 | 024626 | | MOVB EDCOPB,(R0) | :SAVE RESULT |
| 9456 | 025264 | 000761 | | | BR 2\$ | |
| 9457 | 025266 | 005037 | 026052 | EVTWRP: | CLR ENULL | :RESET RESULT = 0 INDICATOR |
| 9458 | 025272 | 013737 | 026060 | 024616 | MOV EVTSGN,ERSNEG | :SETUP SIGN INDICATOR |
| 9459 | 025300 | 012704 | 025624 | | MOV #EVRTAB+12,R4 | :SETUP POINTER TO CONVERT RESULT |
| 9460 | 025304 | 113701 | 017120 | | MOVB E0,R1 | |
| 9461 | 025310 | 160104 | | | SUB R1,R4 | |
| 9462 | 025312 | 013737 | 017120 | 024664 | MOV E0,ELSD | :SETUP POINTER TO LEAST SIGN. DIGIT |
| 9463 | 025320 | 113737 | 017120 | 026054 | MOVB E0,EPOPS | |
| 9464 | 025326 | 005337 | 026054 | | DEC EPOPS | |
| 9465 | 025332 | 013701 | 017120 | | MOV E0,R1 | :SETUP POSITION OF # TO INSERT = 0 |
| 9466 | 025336 | 042701 | 000377 | | BIC #377,R1 | |
| 9467 | 025342 | 013700 | 017122 | | MOV E1,R0 | :SETUP ADDRESS OF # TO LOAD |
| 9468 | 025346 | 005002 | | 2\$: | CLR R2 | :SETUP DATA TO LOAD |
| 9469 | 025350 | 020427 | 025612 | | CMP R4,#EVRTAB | |
| 9470 | 025354 | 103403 | | | BLO 1\$ | |
| 9471 | 025356 | 111402 | | | MOVB (R4),R2 | |
| 9472 | 025360 | 050237 | 026052 | | BIS R2,ENULL | |
| 9473 | 025364 | 005204 | | 1\$: | INC R4 | |
| 9474 | 025366 | 004737 | 022174 | | JSR PC,EPUSH | :CALL ROUTINE TO PUSH DIGIT INTO STRING |
| 9475 | 025372 | 120137 | 026054 | | CMPB R1,EPOPS | :READY TO PUSH SIGN? |
| 9476 | 025376 | 001004 | | | BNE 3\$ | :BRANCH IF NO |
| 9477 | 025400 | 005737 | 026064 | | TST ESCF | :SPECIAL CASE? |
| 9478 | 025404 | 001401 | | | BEQ 3\$ | :BRANCH IF NO |
| 9479 | 025406 | 000405 | | | BR EVTXT | :WORK WITH SPECIAL CASE BEFORE INSERTING SIGN. |
| 9480 | 025410 | 020137 | 017120 | 3\$: | CMP R1,E0 | :ALL DIGITS PLUS SIGN PUSHED? |
| 9481 | 025414 | 001402 | | | BEQ EVTXT | :BRANCH IF YES |
| 9482 | 025416 | 005201 | | | INC R1 | |
| 9483 | 025420 | 000752 | | | BR 2\$ | :RETURN TO PUSH NEXT DIGIT |
| 9484 | | | | | | |
| 9485 | 025422 | 105737 | 017120 | EVTXT: | TSTB E0 | :IF DST.LEN=0 ,DON'T BOTHER TESTING |
| 9486 | | | | | | : FOR SPECIAL CASE. |
| 9487 | 025426 | 001425 | | | BEQ 5\$ | |
| 9488 | 025430 | 005737 | 026064 | | TST ESCF | :SPECIAL CASE? (MOST NEG #). |
| 9489 | 025434 | 001422 | | | BEQ 5\$ | :BRANCH IF NO. |
| 9490 | 025436 | 013737 | 017120 | 024664 | MOV E0,ELSD | |
| 9491 | 025444 | 013701 | 017120 | | MOV E0,R1 | |
| 9492 | 025450 | 005301 | | | DEC R1 | :SET R1=DST.LEN-1. |
| 9493 | 025452 | 013700 | 017122 | | MOV E1,R0 | :INCREMENT LEAST SIGN DIGIT TO 8. |

| | | | | | | |
|------|--------|--------|--------|--------|---------------------|---|
| 9494 | 025456 | 012702 | 000010 | | MOV #10,R2 | |
| 9495 | 025462 | 004737 | 022174 | | JSR PC,EPUSH | ;CALL ROUTINE TO PUSH THE 8 INTO DEST. |
| 9496 | 025466 | 005201 | | | INC R1 | ;INSERT NEGATIVE SIGN |
| 9497 | 025470 | 012737 | 177777 | 024616 | MOV #177777,ERSNEG | |
| 9498 | 025476 | 004737 | 022174 | | 7\$: JSR PC,EPUSH | ;CALL ROUTINE TO PUSH NEG SIGN INTO DEST. |
| 9499 | 025502 | 005077 | 171410 | | 5\$: CLR @EOPSW | ;INIT EMULATE PSW |
| 9500 | 025506 | 005737 | 026052 | | TST ENULL | ;IF = 0 SET Z BIT; SKIP SETTING OF N BIT |
| 9501 | 025512 | 001407 | | | BEQ 1\$ | |
| 9502 | 025514 | 105737 | 024616 | | TSTB ERSNEG | ;IF (-) SET N BIT |
| 9503 | 025520 | 100007 | | | BPL 2\$ | |
| 9504 | 025522 | 052777 | 000010 | 171366 | BIS #10,@EOPSW | ;SET N BIT |
| 9505 | 025530 | 000403 | | | BR 2\$ | |
| 9506 | 025532 | 052777 | 000004 | 171356 | 1\$: BIS #4,@EOPSW | ;SET Z BIT |
| 9507 | 025540 | 123737 | 017120 | 026066 | 2\$: CMPB E0,EVTSSV | |
| 9508 | 025546 | 103003 | | | BHIS 4\$ | |
| 9509 | 025550 | 052777 | 000002 | 171340 | BIS #2,@EOPSW | ;SET OVERFLOW |
| 9510 | 025556 | 013702 | 017114 | | 4\$: MOV EORSTK,R2 | ;REGISTER UNLOAD |
| 9511 | 025562 | 013722 | 017120 | | MOV E0,(R2)+ | ;R0 = R0 |
| 9512 | 025566 | 013722 | 017122 | | MOV E1,(R2)+ | ;R1 = R1 |
| 9513 | 025572 | 005022 | | | CLR (R2)+ | ;R2 = 0 |
| 9514 | 025574 | 005022 | | | CLR (R2)+ | ;R3 = 0 |
| 9515 | 025576 | 013722 | 017130 | | MOV E4,(R2)+ | ;R4 = R4 |
| 9516 | 025602 | 013712 | 017132 | | MOV E5,(R2) | ;R5 = R5 |
| 9517 | 025606 | 000207 | | | RTS PC | |
| 9518 | 025610 | 000000 | | | .WORD 0 | ;MUST PRECEDE EVRTAB!! |
| 9519 | 025612 | 000240 | | | .BLKB ^D160 | |
| 9520 | 026052 | 000000 | | | EVRTAB: .WORD 0 | |
| 9521 | 026054 | 000000 | | | ENULL: .WORD 0 | |
| 9522 | 026056 | 000000 | | | EPOPS: .WORD 0 | |
| 9523 | 026060 | 000000 | | | EPAK: .WORD 0 | |
| 9524 | 026062 | 000000 | | | EVTSGN: .WORD 0 | |
| 9525 | 026064 | 000000 | | | EVTSPAS: .WORD 0 | |
| 9526 | 026066 | 000000 | | | ESCF: .WORD 0 | |
| 9527 | 026070 | 026264 | 000001 | | EVTSSV: .WORD 0 | |
| 9528 | 026074 | 026265 | 000001 | | EVTABA: .WORD E00,1 | |
| 9529 | 026100 | 026266 | 000001 | | .WORD E01,1 | |
| 9530 | 026104 | 026267 | 000001 | | .WORD E02,1 | |
| 9531 | 026110 | 026270 | 000002 | | .WORD E03,1 | |
| 9532 | 026114 | 026272 | 000002 | | .WORD E04,2 | |
| 9533 | 026120 | 026274 | 000002 | | .WORD E05,2 | |
| 9534 | 026124 | 026276 | 000003 | | .WORD E06,2 | |
| 9535 | 026130 | 026301 | 000003 | | .WORD E07,3 | |
| 9536 | 026134 | 026304 | 000003 | | .WORD E08,3 | |
| 9537 | 026140 | 026307 | 000004 | | .WORD E09,3 | |
| 9538 | 026144 | 026313 | 000004 | | .WORD E10,4 | |
| 9539 | 026150 | 026317 | 000004 | | .WORD E11,4 | |
| 9540 | 026154 | 026323 | 000004 | | .WORD E12,4 | |
| 9541 | 026160 | 026327 | 000005 | | .WORD E13,4 | |
| 9542 | 026164 | 026334 | 000005 | | .WORD E14,5 | |
| 9543 | 026170 | 026341 | 000005 | | .WORD E15,5 | |
| 9544 | 026174 | 026346 | 000006 | | .WORD E16,5 | |
| 9545 | 026200 | 026354 | 000006 | | .WORD E17,6 | |
| 9546 | 026204 | 026362 | 000006 | | .WORD E18,6 | |
| 9547 | 026210 | 026370 | 000007 | | .WORD E19,6 | |
| | | | | | .WORD E20,7 | |

| | | | | | | | |
|------|--------|--------|--------|-----|------|-------|---------------------------------|
| 9548 | 026214 | 026377 | 000007 | | | .WORD | E21,7 |
| 9549 | 026220 | 026406 | 000007 | | | .WORD | E22,7 |
| 9550 | 026224 | 026415 | 000007 | | | .WORD | E23,7 |
| 9551 | 026230 | 026424 | 000010 | | | .WORD | E24,10 |
| 9552 | 026234 | 026434 | 000010 | | | .WORD | E25,10 |
| 9553 | 026240 | 026444 | 000010 | | | .WORD | E26,10 |
| 9554 | 026244 | 026454 | 000011 | | | .WORD | E27,11 |
| 9555 | 026250 | 026465 | 000011 | | | .WORD | E28,11 |
| 9556 | 026254 | 026476 | 000011 | | | .WORD | E29,11 |
| 9557 | 026260 | 026507 | 000012 | | | .WORD | E30,12 |
| 9558 | 026264 | 001 | | | E00: | .BYTE | ^D1 |
| 9559 | 026265 | 002 | | | E01: | .BYTE | ^D2 |
| 9560 | 026266 | 004 | | | E02: | .BYTE | ^D4 |
| 9561 | 026267 | 010 | | | E03: | .BYTE | ^D8 |
| 9562 | 026270 | 001 | 006 | | E04: | .BYTE | ^D1,^D6 |
| 9563 | 026272 | 003 | 002 | | E05: | .BYTE | ^D3,^D2 |
| 9564 | 026274 | 006 | 004 | | E06: | .BYTE | ^D6,^D4 |
| 9565 | 026276 | 001 | 002 | 010 | E07: | .BYTE | ^D1,^D2,^D8 |
| 9566 | 026301 | 002 | 005 | 006 | E08: | .BYTE | ^D2,^D5,^D6 |
| 9567 | 026304 | 005 | 001 | 002 | E09: | .BYTE | ^D5,^D1,^D2 |
| 9568 | 026307 | 001 | 000 | 002 | E10: | .BYTE | ^D1,^D0,^D2,^D4 |
| | 026312 | 004 | | | | | |
| 9569 | 026313 | 002 | 000 | 004 | E11: | .BYTE | ^D2,^D0,^D4,^D8 |
| | 026316 | 010 | | | | | |
| 9570 | 026317 | 004 | 000 | 011 | E12: | .BYTE | ^D4,^D0,^D9,^D6 |
| | 026322 | 006 | | | | | |
| 9571 | 026323 | 010 | 001 | 011 | E13: | .BYTE | ^D8,^D1,^D9,^D2 |
| | 026326 | 002 | | | | | |
| 9572 | 026327 | 001 | 006 | 003 | E14: | .BYTE | ^D1,^D6,^D3,^D8,^D4 |
| | 026332 | 010 | 004 | | | | |
| 9573 | 026334 | 003 | 002 | 007 | E15: | .BYTE | ^D3,^D2,^D7,^D6,^D8 |
| | 026337 | 006 | 010 | | | | |
| 9574 | 026341 | 006 | 005 | 005 | E16: | .BYTE | ^D6,^D5,^D5,^D3,^D6 |
| | 026344 | 003 | 006 | | | | |
| 9575 | 026346 | 001 | 003 | 001 | E17: | .BYTE | ^D1,^D3,^D1,^D0,^D7,^D2 |
| | 026351 | 000 | 007 | 002 | | | |
| 9576 | 026354 | 002 | 006 | 002 | E18: | .BYTE | ^D2,^D6,^D2,^D1,^D4,^D4 |
| | 026357 | 001 | 004 | 004 | | | |
| 9577 | 026362 | 005 | 002 | 004 | E19: | .BYTE | ^D5,^D2,^D4,^D2,^D8,^D8 |
| | 026365 | 002 | 010 | 010 | | | |
| 9578 | 026370 | 001 | 000 | 004 | E20: | .BYTE | ^D1,^D0,^D4,^D8,^D5,^D7,^D6 |
| | 026373 | 010 | 005 | 007 | | | |
| | 026376 | 006 | | | | | |
| 9579 | 026377 | 002 | 000 | 011 | E21: | .BYTE | ^D2,^D0,^D9,^D7,^D1,^D5,^D2 |
| | 026402 | 007 | 001 | 005 | | | |
| | 026405 | 002 | | | | | |
| 9580 | 026406 | 004 | 001 | 011 | E22: | .BYTE | ^D4,^D1,^D9,^D4,^D3,^D0,^D4 |
| | 026411 | 004 | 003 | 000 | | | |
| | 026414 | 004 | | | | | |
| 9581 | 026415 | 010 | 003 | 010 | E23: | .BYTE | ^D8,^D3,^D8,^D8,^D6,^D0,^D8 |
| | 026420 | 010 | 006 | 000 | | | |
| | 026423 | 010 | | | | | |
| 9582 | 026424 | 001 | 006 | 007 | E24: | .BYTE | ^D1,^D6,^D7,^D7,^D7,^D2,^D1,^D6 |
| | 026427 | 007 | 007 | 002 | | | |

| | | | | | | | | | |
|------|--------|-----|-----|-----|------|-------|---|--|--|
| 9583 | 026432 | 001 | 006 | | | | | | |
| | 026434 | 003 | 003 | 005 | E25: | .BYTE | ^D3,^D3,^D5,^D5,^D4,^D4,^D3,^D2 | | |
| | 026437 | 005 | 004 | 004 | | | | | |
| | 026442 | 003 | 002 | | | | | | |
| 9584 | 026444 | 006 | 007 | 001 | E26: | .BYTE | ^D6,^D7,^D1,^D0,^D8,^D8,^D6,^D4 | | |
| | 026447 | 000 | 010 | 010 | | | | | |
| | 026452 | 006 | 004 | | | | | | |
| 9585 | 026454 | 001 | 003 | 004 | E27: | .BYTE | ^D1,^D3,^D4,^D2,^D1,^D7,^D7,^D2,^D8 | | |
| | 026457 | 002 | 001 | 007 | | | | | |
| | 026462 | 007 | 002 | 010 | | | | | |
| 9586 | 026465 | 002 | 006 | 010 | E28: | .BYTE | ^D2,^D6,^D8,^D4,^D3,^D5,^D4,^D5,^D6 | | |
| | 026470 | 004 | 003 | 005 | | | | | |
| | 026473 | 004 | 005 | 006 | | | | | |
| 9587 | 026476 | 005 | 003 | 006 | E29: | .BYTE | ^D5,^D3,^D6,^D8,^D7,^D0,^D9,^D1,^D2 | | |
| | 026501 | 010 | 007 | 000 | | | | | |
| | 026504 | 011 | 001 | 002 | | | | | |
| 9588 | 026507 | 001 | 000 | 007 | E30: | .BYTE | ^D1,^D0,^D7,^D3,^D7,^D4,^D1,^D8,^D2,^D4 | | |
| | 026512 | 003 | 007 | 004 | | | | | |
| | 026515 | 001 | 010 | 002 | | | | | |
| | 026520 | 004 | | | | | | | |
| 9589 | 026522 | | | | | .EVEN | | | |

```
9591          .SBTTL          CONVERT PACKED OR ZONED TO LONG
9592          :          CONVERT A NUMBER THAT IS IN EITHER PACKED OR
9593          :          ZONED FORMAT INTO LONG FORMAT.          (CVTPL,CVTNL)
9594          -----
9595 026522 005037 026056  ECVTNL:  CLR EPAK          ;ZONED FORMAT INPUT
9596 026526 000403          BR EZL1
9597 026530 012737 177777 026056  ECVTPL:  MOV #177777,EPAK      ;PACKED FORMAT INPUT
9598 026536 005037 027262          EZL1:   CLR EFLO          ;RESET OVERFLOW
9599 026542 005037 024646          CLR EODD         ;RESET ODD LENGTH INDICATOR
9600 026546 005003          CLR R3           ;CLEAR RESULT DATA AREA
9601 026550 005004          CLR R4
9602 026552 005037 026052          CLR ENULL        ;RESET NULL INDICATOR
9603 026556 032737 000001 017120  BIT #1,E0        ;IS SOURCE ODD OR EVEN NO. OF DIGITS
9604 026564 001403          BEQ 1$           ;ITS EVEN
9605 026566 012737 177777 024646  MOV #177777,EODD ;ODD
9606          :
9607          :
9608 026574 013701 017120          1$:   FIND SIGN OF NUMBER
9609 026600 013737 017120 024664  MOV E0,R1        ;SETUP POSITION OF SIGN
9610 026606 013700 017122          MOV E0,ELSD      ;SETUP POSITION OF LEAST SIGN DIGIT.
9611 026612 004737 021126          MOV E1,R0        ;SETUP ADDRESS OF STRING
9612 026616 005037 024606          JSR PC,ESNK      ;CALL ROUTINE TO FIND SIGN
9613 026622 005737 024616          CLR ES1
9614 026626 001403          TST ERSNEG       ;IS SIGN NEGATIVE?
9615 026630 012737 177777 024606  BEQ EZL2         ;BRANCH IF NO
9616          :
9617          :
9618          :
9619          :
9620          :
9621          :
9622 026636 013737 017120 027264  EZL2:  CLEAR RESULT
          :          FOR COUNT = 1 TO LENGTH
          :          RESULT = (RESULT*10)+DIGIT<MSD+COUNT>
          :          NEXT COUNT
9623          :
9624 026644 042737 000377 027264  MOV E0,ECOUN     ;RESET COUNT(UPPER BYTE OF ECOUN MUST
9625 026652 013737 017120 024664  BIC #377,ECOUN   ; CONTAIN TYPE)
9626 026660 123737 027264 017120  10$:   CMPB ECOUN,E0    ;CONVERSION COMPLETE YET
9627 026666 001474          BEQ EZLE         ;YES
9628 026670 013701 027264          MOV ECOUN,R1     ;POSITION OF SOURCE
9629 026674 013700 017122          MOV E1,R0        ;START ADDRESS OF SOURCE
9630 026700 004737 021126          JSR PC,ESNK      ;CALL ROUTINE TO FIND DIGIT
9631 026704 005237 027264          INC ECOUN        ;BUMP COUNTER
9632 026710 050237 026052          BIS R2,ENULL    ;DIGIT SUM FOR NULL TEST
9633 026714 000257          CCC           ;MULTIPLY RESULT BY 10
9634 026716 006104          ROL R4
9635 026720 006103          ROL R3
9636 026722 103003          BCC 3$          ;ANY BIT SHIFTED OUT IS OVERFLOW
9637 026724 012737 177777 027262  MOV #177777,EFLO
9638 026732 010337 017136          3$:   MOV R3,TEMP      ;TEMP DATA HOLD FOR MULTIPLY
9639 026736 010437 024644          MOV R4,TEMP1
9640 026742 000257          CCC
9641 026744 006104          ROL R4
9642 026746 006103          ROL R3
9643 026750 103003          BCC 4$
9644 026752 012737 177777 027262  MOV #177777,EFLO
```

| | | | | | | |
|------|--------|--------|---------------|-------|------------------------------------|--|
| 9645 | 026760 | 000257 | | 4\$: | CCC | |
| 9646 | 026762 | 006104 | | | ROL R4 | |
| 9647 | 026764 | 006103 | | | ROL R3 | |
| 9648 | 026766 | 103003 | | | BCC 5\$ | |
| 9649 | 026770 | 012737 | 177777 027262 | | MOV #177777,EFLO | |
| 9650 | 026776 | 063704 | 024644 | 5\$: | ADD TEMP1,R4 | ;COMPLETE MULTIPLY |
| 9651 | 027002 | 103007 | | | BCC 6\$ | |
| 9652 | 027004 | 000257 | | | CCC | |
| 9653 | 027006 | 062703 | 000001 | | ADD #1,R3 | |
| 9654 | 027012 | 103003 | | | BCC 6\$ | |
| 9655 | 027014 | 012737 | 177777 027262 | | MOV #177777,EFLO | |
| 9656 | 027022 | 063703 | 017136 | 6\$: | ADD TEMP,R3 | |
| 9657 | 027026 | 103003 | | | BCC 8\$ | |
| 9658 | 027030 | 012737 | 177777 027262 | | MOV #177777,EFLO | |
| 9659 | 027036 | 060204 | | 8\$: | ADD R2,R4 | ;ADD NEW DIGIT |
| 9660 | 027040 | 103307 | | | BCC 10\$ | |
| 9661 | 027042 | 062703 | 000001 | | ADD #1,R3 | |
| 9662 | 027046 | 103304 | | | BCC 10\$ | |
| 9663 | 027050 | 012737 | 177777 027262 | | MOV #177777,EFLO | |
| 9664 | 027056 | 000700 | | | BR 10\$ | |
| 9665 | | | | | | |
| 9666 | | | | | | |
| 9667 | 027060 | 005703 | | | SET CC BITS , SET RESULT REGISTERS | |
| 9668 | | | | EZLE: | TST R3 | ;BIT 32= 1 IS OVERFLOW EXCEPT IF |
| 9669 | 027062 | 100013 | | | BPL 6\$ | ; ALL OTHER 31 BITS = 0 & SRC WAS NEGATIVE |
| 9670 | 027064 | 005704 | | | TST R4 | ;BRANCH IF OK |
| 9671 | 027066 | 001006 | | | BNE 7\$ | ;ALL OTHER 31 BITS = 0? |
| 9672 | 027070 | 032703 | 077777 | | BIT #77777,R3 | ;BRANCH IF NO |
| 9673 | 027074 | 001003 | | | BNE 7\$ | ;BRANCH IF NO |
| 9674 | 027076 | 005737 | 024606 | | TST ES1 | ;WAS SRC NEGATIVE? |
| 9675 | 027102 | 100403 | | | BMI 6\$ | ;BRANCH IF YES |
| 9676 | 027104 | 012737 | 177777 027262 | 7\$: | MOV #177777,EFLO | ;ELSE OVERFLOW |
| 9677 | 027112 | 005077 | 170000 | 6\$: | CLR @EOPSW | ;RESET PSW |
| 9678 | 027116 | 005737 | 027262 | | TST EFLO | ;ANY OVERFLOW |
| 9679 | 027122 | 001403 | | | BEQ 1\$ | ;NO |
| 9680 | 027124 | 052777 | 000002 167764 | | BIS #2,@EOPSW | ;SET V BIT |
| 9681 | 027132 | 005704 | | 1\$: | TST R4 | ;WAS RESULT = 0 |
| 9682 | 027134 | 001006 | | | BNE 5\$ | ;NO |
| 9683 | 027136 | 005703 | | | TST R3 | |
| 9684 | 027140 | 001004 | | | BNE 5\$ | ;NO |
| 9685 | 027142 | 052777 | 000004 167746 | | BIS #4,@EOPSW | ;SET Z BIT |
| 9686 | 027150 | 000412 | | | BR 2\$ | |
| 9687 | 027152 | 005737 | 024606 | 5\$: | TST ES1 | ;WHAT SIGN |
| 9688 | 027156 | 100007 | | | BPL 2\$ | ;POSITIVE |
| 9689 | 027160 | 005103 | | | COM R3 | ;COMPLEMENT VALUE |
| 9690 | 027162 | 005104 | | | COM R4 | |
| 9691 | 027164 | 000257 | | | CCC | |
| 9692 | 027166 | 062704 | 000001 | | ADD #1,R4 | |
| 9693 | 027172 | 103001 | | | BCC 2\$ | |
| 9694 | 027174 | 005203 | | | INC R3 | |
| 9695 | 027176 | 005703 | | 2\$: | TST R3 | ;SET N BIT BASED ON SIGN OF RESULT |
| 9696 | 027200 | 100003 | | | BPL 3\$ | ;IF RESULT SIGN =+ BRANCH |
| 9697 | 027202 | 052777 | 000010 167706 | | BIS #10,@EOPSW | ;SET N BIT |
| 9698 | 027210 | 032777 | 000004 167700 | 3\$: | BIT #4,@EOPSW | ;WAS DST = 0 |

| | | | | | | |
|------|--------|--------|--------|--------|---------------|--------------------------|
| 9699 | 027216 | 001006 | | | BNE 4\$ | :YES |
| 9700 | 027220 | 005737 | 024606 | | TST ES1 | :AND WAS SOURCE NEGATIVE |
| 9701 | 027224 | 100003 | | | BPL 4\$ | :NO |
| 9702 | 027226 | 052777 | 000001 | 167662 | BIS #1,@EOPSW | :YES , THEN SET C BIT |
| 9703 | 027234 | 013700 | 017114 | | MOV EORSTK,R0 | :REGISTER SAVE |
| 9704 | 027240 | 005020 | | | CLR (R0)+ | :R0 =0 |
| 9705 | 027242 | 005020 | | | CLR (R0)+ | :R1 = 0 |
| 9706 | 027244 | 010320 | | | MOV R3,(R0)+ | :R2 = DST HIGH |
| 9707 | 027246 | 010420 | | | MOV R4,(R0)+ | :R3 = DST LOW |
| 9708 | 027250 | 013720 | 017130 | | MOV E4,(R0)+ | :R4 = R4 |
| 9709 | 027254 | 013720 | 017132 | | MOV E5,(R0)+ | :R5 = R5 |
| 9710 | 027260 | 000207 | | | RTS PC | |
| 9711 | 027262 | 000000 | | EFLO: | .WORD 0 | |
| 9712 | 027264 | 000000 | | ECOUN: | .WORD 0 | |

| | | | | | | | |
|------|--------|--------|--------|--------|---------|------------------|--|
| 9714 | | | | | | .SBTTL | CONVERT PACKED TO ZONED |
| 9715 | 027266 | 012737 | 177777 | 026056 | ECVTPN: | MOV #177777,EPAK | :SET PACKED MODE INDICATOR |
| 9716 | 027274 | 005037 | 026066 | | | CLR EVTSSV | :RESET PASS COUNTER |
| 9717 | 027300 | 005037 | 026052 | | | CLR ENULL | :RESET NULL INDICATOR |
| 9718 | 027304 | 005077 | 167606 | | | CLR @EOPSW | :RESET EMULATE PSW |
| 9719 | 027310 | 013737 | 017120 | 024624 | | MOV E0,EDCOPA | :START POSITION OF SOURCE |
| 9720 | 027316 | 013737 | 017124 | 024626 | | MOV E2,EDCOPB | :START POSITION OF DEST. |
| 9721 | 027324 | 005037 | 024646 | | | CLR EODD | :IS SOURCE ODD OR EVEN NUMBER |
| 9722 | 027330 | 032737 | 000001 | 017120 | | BIT #1,E0 | :OF DIGITS |
| 9723 | 027336 | 001403 | | | | BEQ EPZ1 | :ITS EVEN |
| 9724 | 027340 | 012737 | 177777 | 024646 | | MOV #177777,EODD | :ITS ODD |
| 9725 | 027346 | 004737 | 030310 | | EPZ1: | JSR PC,EFMSD | :DETERMINE POSITION OF MOST SIGN SRC DIGIT |
| 9726 | 027352 | 005002 | | | | CLR R2 | |
| 9727 | 027354 | 105337 | 024624 | | | DECB EDCOPA | :END OF SOURCE YET |
| 9728 | 027360 | 100414 | | | | BMI 1\$ | :YES |
| 9729 | 027362 | 042777 | 000001 | 167526 | | BIC #1,@EOPSW | :RESET END OF SOURCE FLAG |
| 9730 | 027370 | 013700 | 017122 | | | MOV E1,R0 | :START ADDRESS OF SOURCE |
| 9731 | 027374 | 013701 | 024624 | | | MOV EDCOPA,R1 | :POSITION OF DIGIT |
| 9732 | 027400 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 9733 | 027406 | 004737 | 021150 | | | JSR PC,EFINDT | :GRAB DIGIT |
| 9734 | 027412 | 105337 | 024626 | | 1\$: | DECB EDCOPB | :END OF DEST. YET |
| 9735 | 027416 | 100421 | | | | BMI EPZE | :YES |
| 9736 | 027420 | 005237 | 026066 | | | INC EVTSSV | :DIGIT POSITION |
| 9737 | 027424 | 005702 | | | | TST R2 | :IS DIGIT ZERO |
| 9738 | 027426 | 001403 | | | | BEQ 2\$ | |
| 9739 | 027430 | 013737 | 026066 | 026052 | | MOV EVTSSV,ENULL | :POSITION OF LAST NON ZERO DIGIT STORED. |
| 9740 | 027436 | 013700 | 017126 | | 2\$: | MOV E3,R0 | :START ADDRESS OF DEST |
| 9741 | 027442 | 013701 | 024626 | | | MOV EDCOPB,R1 | :POSITION OF DIGIT |
| 9742 | 027446 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 9743 | 027454 | 004737 | 022452 | | | JSR PC,EPUDTZ | :SAVE DIGIT |
| 9744 | 027460 | 000732 | | | | BR EPZ1 | :LOOP TILL COMPLETE |
| 9745 | 027462 | 013700 | 017122 | | EPZE: | MOV E1,R0 | :START ADDRESS OF SOURCE |
| 9746 | 027466 | 013701 | 017120 | | | MOV E0,R1 | :POSITION OF SIGN |
| 9747 | 027472 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 9748 | 027500 | 004737 | 021150 | | | JSR PC,EFINDT | :GRAB SIGN |
| 9749 | 027504 | 005037 | 024606 | | | CLR ES1 | :RESET SIGN FLAG |
| 9750 | 027510 | 005737 | 024616 | | | TST ERSNEG | :IS SOURCE NEG? |
| 9751 | 027514 | 001406 | | | | BEQ 2\$ | :BRANCH IF NO |
| 9752 | 027516 | 005737 | 030372 | | | TST EMSDP | :IF SRC IS ZERO AND NEG TREAT IT AS POSITIVE |
| 9753 | 027522 | 001403 | | | | BEQ 2\$ | |
| 9754 | 027524 | 012737 | 177777 | 024606 | | MOV #177777,ES1 | :SET NEGATIVE FLAG |
| 9755 | 027532 | 005737 | 026052 | | 2\$: | TST ENULL | :WAS RESULT STORED ZERO |
| 9756 | 027536 | 001004 | | | | BNE 3\$ | :NO |
| 9757 | 027540 | 052777 | 000004 | 167350 | | BIS #4,@EOPSW | :SET Z BIT |
| 9758 | 027546 | 000406 | | | | BR 6\$ | :SKIP SETTING OF N BIT |
| 9759 | 027550 | 005737 | 024606 | | 3\$: | TST ES1 | :SIGN OF RESULT |
| 9760 | 027554 | 001403 | | | | BEQ 6\$ | :POSITIVE |
| 9761 | 027556 | 052777 | 000010 | 167332 | | BIS #10,@EOPSW | :SET N BIT |
| 9762 | 027564 | 013737 | 024606 | 024616 | 6\$: | MOV ES1,ERSNEG | :SETUP SIGN OF RESULT |
| 9763 | 027572 | 013701 | 017124 | | | MOV E2,R1 | :POSITION OF SIGN |
| 9764 | 027576 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 9765 | 027604 | 013700 | 017126 | | | MOV E3,R0 | :START ADDRESS OF DEST. |
| 9766 | 027610 | 004737 | 022452 | | | JSR PC,EPUDTZ | :INSERT SIGN IN DST STRING |
| 9767 | 027614 | 005737 | 024616 | | | TST ERSNEG | :WAS SIGN STORED POSITIVE?(UNSIGNED) |

| | | | | | | | |
|------|--------|--------|--------|--------|------|-----------------|-------------------------------|
| 9768 | 027620 | 001003 | | | | BNE 4\$ | :BRANCH IF NO |
| 9769 | 027622 | 042777 | 000010 | 167266 | | BIC #10,@EOPSW | :CLEAR PSW N BIT |
| 9770 | 027630 | 023737 | 030372 | 026052 | 4\$: | CMP EMSDP,ENULL | :CAN DEST. CONTAIN ALL DIGITS |
| 9771 | 027636 | 101403 | | | | BLOS 5\$ | :YES |
| 9772 | 027640 | 052777 | 000002 | 167250 | | BIS #2,@EOPSW | :SET V BIT |
| 9773 | 027646 | 013702 | 017114 | | 5\$: | MOV EORSTK,R2 | :SAVE REGISTERS |
| 9774 | 027652 | 005022 | | | | CLR (R2)+ | :R0 = 0 |
| 9775 | 027654 | 005022 | | | | CLR (R2)+ | :R1 = 0 |
| 9776 | 027656 | 013722 | 017124 | | | MOV E2,(R2)+ | :R2 = R2 |
| 9777 | 027662 | 013722 | 017126 | | | MOV E3,(R2)+ | :R3 = R3 |
| 9778 | 027666 | 013722 | 017130 | | | MOV E4,(R2)+ | :R4 = R4 |
| 9779 | 027672 | 013722 | 017132 | | | MOV E5,(R2)+ | :R5 = R5 |
| 9780 | 027676 | 000207 | | | | RTS PC | |

| | | | | | | |
|------|--------|--------|--------|--------|------------------|---|
| 9782 | | | | | .SBTTL | CONVERT ZONED TO PACKED |
| 9783 | 027700 | 005037 | 026066 | | CLR EVTSSV | :RESET PASS COUNTER |
| 9784 | 027704 | 005037 | 026056 | | CLR EPACK | :MODE = ZONED |
| 9785 | 027710 | 005037 | 026052 | | CLR ENULL | :RESET NULL INDICATOR |
| 9786 | 027714 | 005077 | 167176 | | CLR @EOPSW | :RESET EMULATE PSW |
| 9787 | 027720 | 013737 | 017120 | 024624 | MOV E0,EDCOPA | :START POSITION OF SOURCE |
| 9788 | 027726 | 013737 | 017124 | 024626 | MOV E2,EDCOPB | :START POSITION OF DEST. |
| 9789 | 027734 | 005037 | 024646 | | CLR EODD | :IS DEST. ODD OR EVEN # OF DIGITS. |
| 9790 | 027740 | 032737 | 000001 | 017124 | BIT #1,E2 | |
| 9791 | 027746 | 001403 | | | BEQ EZP1 | :ITS EVEN |
| 9792 | 027750 | 012737 | 177777 | 024646 | MOV #177777,EODD | :ITS ODD |
| 9793 | 027756 | 004737 | 030310 | | JSR PC,EFMSD | :FIND POSITION OF MOST SIGNIFICANT SOURCE DIGIT |
| 9794 | 027762 | 005002 | | | CLR R2 | |
| 9795 | 027764 | 105337 | 024624 | | DECB EDCOPA | :END OF SOURCE YET |
| 9796 | 027770 | 100414 | | | BMI 1\$ | |
| 9797 | 027772 | 042777 | 000001 | 167116 | BIC #1,@EOPSW | :RESET END OF SOURCE FLAG |
| 9798 | 030000 | 013700 | 017122 | | MOV E1,R0 | :START ADDRESS OF SOURCE |
| 9799 | 030004 | 013701 | 024624 | | MOV EDCOPA,R1 | :POSITION OF DIGIT |
| 9800 | 030010 | 013737 | 017120 | 024664 | MOV E0,ELSD | |
| 9801 | 030016 | 004737 | 021352 | | JSR PC,EFNDTZ | :GRAB DIGIT |
| 9802 | 030022 | 105337 | 024626 | | DECB EDCOPB | :END OF DEST. YET |
| 9803 | 030026 | 100421 | | | BMI EZPE | :YES |
| 9804 | 030030 | 005237 | 026066 | | INC EVTSSV | :PASS COUNTER |
| 9805 | 030034 | 005702 | | | TST R2 | :IS DIGIT ZERO |
| 9806 | 030036 | 001403 | | | BEQ 2\$ | |
| 9807 | 030040 | 013737 | 026066 | 026052 | MOV EVTSSV,ENULL | :SAVE POSITION OF LAST NON ZERO DIGIT STORED. |
| 9808 | 030046 | 013700 | 017126 | | MOV E3,R0 | :START ADDRESS OF DEST. |
| 9809 | 030052 | 013701 | 024626 | | MOV EDCOPB,R1 | :POSITION OF DIGIT |
| 9810 | 030056 | 013737 | 017124 | 024664 | MOV E2,ELSD | |
| 9811 | 030064 | 004737 | 022216 | | JSR PC,EPUTDT | :SAVE DIGIT |
| 9812 | 030070 | 000732 | | | BR EZP1 | :LOOP TILL COMPLETE |
| 9813 | 030072 | 013700 | 017122 | | MOV E1,R0 | :START ADDRESS OF SOURCE |
| 9814 | 030076 | 013701 | 017120 | | MOV E0,R1 | :POSITION OF SIGN |
| 9815 | 030102 | 013737 | 017120 | 024664 | MOV E0,ELSD | |
| 9816 | 030110 | 004737 | 021352 | | JSR PC,EFNDTZ | |
| 9817 | 030114 | 005037 | 024606 | | CLR ES1 | :RESET SIGN FLAG |
| 9818 | 030120 | 005737 | 024616 | | TST ERSNEG | :IS SOURCE NEG? |
| 9819 | 030124 | 001406 | | | BEQ 2\$ | :NO |
| 9820 | 030126 | 005737 | 030372 | | TST EMSDP | :IF SOURCE IS ZERO AND NEG TREAT IT AS POSITIVE |
| 9821 | 030132 | 001403 | | | BEQ 2\$ | |
| 9822 | 030134 | 012737 | 177777 | 024606 | MOV #177777,ES1 | :SET NEGATIVE FLAG |
| 9823 | 030142 | 005737 | 026052 | | TST ENULL | :WAS RESULT STORED ZERO |
| 9824 | 030146 | 001004 | | | BNE 3\$ | :NO |
| 9825 | 030150 | 052777 | 000004 | 166740 | BIS #4,@EOPSW | :SET Z BIT |
| 9826 | 030156 | 000406 | | | BR 4\$ | |
| 9827 | 030160 | 005737 | 024606 | | TST ES1 | :SET SIGN OF RESULT |
| 9828 | 030164 | 001403 | | | BEQ 4\$ | :POSITIVE |
| 9829 | 030166 | 052777 | 000010 | 166722 | BIS #10,@EOPSW | :SET N BIT |
| 9830 | 030174 | 023737 | 030372 | 026052 | CMP EMSDP,ENULL | :CAN DEST. CONTAIN ALL DIGITS |
| 9831 | 030202 | 101403 | | | BLOS 5\$ | :YES |
| 9832 | 030204 | 052777 | 000002 | 166704 | BIS #2,@EOPSW | :SET V BIT |
| 9833 | 030212 | 013701 | 017124 | | MOV E2,R1 | :POSITION OF SIGN |
| 9834 | 030216 | 013700 | 017126 | | MOV E3,R0 | :START ADDRESS OF NUMBER |
| 9835 | 030222 | 013737 | 024606 | 024616 | MOV ES1,ERSNEG | |

| | | | | | | | |
|------|--------|--------|--------|--------|------|----------------|---------------------------------------|
| 9836 | 030230 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 9837 | 030236 | 004737 | 022216 | | | JSR PC,EPUTDT | ;SAVE SIGN |
| 9838 | 030242 | 005737 | 024616 | | | TST ERSNEG | ;WAS SIGN STORED POSITIVE? (UNSIGNED) |
| 9839 | 030246 | 001003 | | | | BNE 1\$ | ;BRANCH IF NO |
| 9840 | 030250 | 042777 | 000010 | 166640 | | BIC #10,@EOPSW | ;CLEAR PSW N BIT |
| 9841 | 030256 | 013702 | 017114 | | 1\$: | MOV EORSTK,R2 | ;SAVE REGISTERS |
| 9842 | 030262 | 005022 | | | | CLR (R2)+ | ;R0 = 0 |
| 9843 | 030264 | 005022 | | | | CLR (R2)+ | ;R1 = 0 |
| 9844 | 030266 | 013722 | 017124 | | | MOV E2,(R2)+ | ;R2 = R2 |
| 9845 | 030272 | 013722 | 017126 | | | MOV E3,(R2)+ | ;R3 = R3 |
| 9846 | 030276 | 013722 | 017130 | | | MOV E4,(R2)+ | ;R4 = R4 |
| 9847 | 030302 | 013722 | 017132 | | | MOV E5,(R2)+ | ;R5 = R5 |
| 9848 | 030306 | 000207 | | | | RTS PC | |

| | | | | | | | |
|------|--------|--------|--------|--------|--------|----------------|----------------------------------|
| 9850 | 030310 | 005037 | 030374 | | EFMSD: | CLR ESDC | |
| 9851 | 030314 | 005037 | 030372 | | | CLR EMSDP | |
| 9852 | 030320 | 013700 | 017122 | | | MOV E1,R0 | ;SET R0=SRC.ADR |
| 9853 | 030324 | 013701 | 017120 | | | MOV E0,R1 | ;SET R1=SRC.PTR |
| 9854 | 030330 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 9855 | 030336 | 005301 | | | 1\$: | DEC R1 | |
| 9856 | 030340 | 105701 | | | | TSTB R1 | ;LOOKED AT ALL SRC DIGITS YET? |
| 9857 | 030342 | 100412 | | | | BMI 2\$ | ;BRANCH IF YES |
| 9858 | 030344 | 004737 | 021126 | | | JSR PC,ESNK | ;GET NEXT SRC DIGIT |
| 9859 | 030350 | 005237 | 030374 | | | INC ESDC | ;INCREMENT DIGIT COUNTER |
| 9860 | 030354 | 005702 | | | | TST R2 | ;IS DIGIT ZERO (NON-SIGNIFICANT) |
| 9861 | 030356 | 001767 | | | | BEQ 1\$ | ;BRANCH IF YES |
| 9862 | 030360 | 013737 | 030374 | 030372 | | MOV ESDC,EMSDP | ;SAVE POSITION OF DIGIT |
| 9863 | 030366 | 000763 | | | | BR 1\$ | |
| 9864 | 030370 | 000207 | | | 2\$: | RTS PC | |
| 9865 | | | | | | | |
| 9866 | 030372 | 000000 | | | EMSDP: | .WORD 0 | |
| 9867 | 030374 | 000000 | | | ESDC: | .WORD 0 | |

| | | | | | | |
|------|--------|--------|--------|--------|---------------------|--|
| 9923 | 030654 | 001515 | | | BEQ EFILZ | ;BRANCH IF THIS IS THE SPECIAL CASE. |
| 9924 | 030656 | 105703 | | | TSTB R3 | |
| 9925 | 030660 | 100003 | | | BPL 2\$ | ;BRANCH IF SRC PTR IS POSITIVE |
| 9926 | 030662 | 122703 | 177777 | | CMPB #-1,R3 | |
| 9927 | 030666 | 001110 | | | BNE EFILZ | ;IF SRC PTR IS <-1 FILL DST WITH ZFRO |
| 9928 | 030670 | 013700 | 017122 | | 2\$: MOV E1,R0 | ;FIND MOST SIGNIFICANT DIGIT TO BE |
| 9929 | 030674 | 010301 | | | MOV R3,R1 | ; SHIFTED OUT |
| 9930 | 030676 | 013737 | 017120 | 024664 | MOV E0,ELSD | |
| 9931 | 030704 | 105201 | | | INCB R1 | ;SET R0=SRC.ADR,R1=SRC.PTR+1 |
| 9932 | 030706 | 013737 | 031672 | 024646 | MOV EODDS,EODD | |
| 9933 | 030714 | 004737 | 021126 | | JSR PC,ESNK | ;CALL ROUTINE TO FIND DIGIT |
| 9934 | 030720 | 113700 | 017131 | | MOVB E4+1,R0 | |
| 9935 | 030724 | 060002 | | | ADD R0,R2 | ;ADD RND.DGT TO DIGIT FOUND IN R2 |
| 9936 | 030726 | 022702 | 000012 | | CMP #12,R2 | ;IS RESULT LESS THAN 10 |
| 9937 | 030732 | 101004 | | | BHI 1\$ | |
| 9938 | 030734 | 012737 | 000001 | 024622 | MOV #1,ECARRY | ;NO-SET CARRY |
| 9939 | 030742 | 000402 | | | BR ESISRC | |
| 9940 | 030744 | 005037 | 024622 | | 1\$: CLR ECARRY | |
| 9941 | | | | | | |
| 9942 | 030750 | | | | ESISRC: | ;SHIFT SRC DIGITS INTO DST. |
| 9943 | 030750 | 122704 | 000377 | | CMPB #377,R4 | ;IS RESULT.PTR <0? |
| 9944 | 030754 | 001457 | | | BEQ EDETSN | ;YES - BRANCH TO COPY RESULT INTO DEST |
| 9945 | 030756 | 105703 | | | TSTB R3 | ;NO - IS SRC.PTR <0? |
| 9946 | 030760 | 100453 | | | BMI EFILZ | |
| 9947 | 030762 | 013700 | 017122 | | MOV E1,R0 | ;NO - FIND SRC DIGIT TO SHIFT INTO RESULT |
| 9948 | 030766 | 010301 | | | MOV R3,R1 | ;SET R0=SRC.ADR, R1=SRC.PTR |
| 9949 | 030770 | 013737 | 017120 | 024664 | MOV E0,ELSD | |
| 9950 | 030776 | 013737 | 031672 | 024646 | MOV EODDS,EODD | |
| 9951 | 031004 | 004737 | 021126 | | JSR PC,ESNK | ;CALL ROUTINE TO FIND SRC DIGIT |
| 9952 | 031010 | 063702 | 024622 | | ADD ECARRY,R2 | ;ADD CARRY TO DIGIT FOUND |
| 9953 | 031014 | 022702 | 000011 | | CMP #11,R2 | ;DIGIT OVERFLOW? |
| 9954 | 031020 | 103005 | | | BHIS 2\$ | ;IF NO BRANCH |
| 9955 | 031022 | 005002 | | | CLR R2 | ;OVERFLOW - SET CARRY & SET DIGIT=0 |
| 9956 | 031024 | 012737 | 000001 | 024622 | MOV #1,ECARRY | |
| 9957 | 031032 | 000402 | | | BR 3\$ | |
| 9958 | 031034 | 005037 | 024622 | | 2\$: CLR ECARRY | |
| 9959 | 031040 | 012700 | 025612 | | 3\$: MOV #EVRTAB,R0 | ;PUSH DIGIT FOUND INTO RESULT. |
| 9960 | 031044 | 010401 | | | MOV R4,R1 | ;SET R0=RESULT.ADR, R1=RESULT.PTR, R2 CONTAINS D |
| 9961 | 031046 | 013737 | 032632 | 024664 | MOV ETLSD,ELSD | |
| 9962 | 031054 | 012737 | 177777 | 024646 | MOV #177777,EODD | |
| 9963 | 031062 | 004737 | 022174 | | JSR PC,EPUSH | ;CALL ROUTINE TO PUSH DIGIT INTO RESULT |
| 9964 | 031066 | 042702 | 177760 | | BIC #177760,R2 | ;MASK OFF ALL BUT DIGIT PUSHED |
| 9965 | 031072 | 001403 | | | BEQ 1\$ | ;IS DIGIT PUSHED=0? |
| 9966 | 031074 | 012737 | 177777 | 031704 | MOV #177777,ETNZI | ;NO-SET NON ZERO INDICATOR |
| 9967 | 031102 | 105303 | | | 1\$: DECB R3 | ;DECREMENT SRC.PTR |
| 9968 | 031104 | 105304 | | | DECB R4 | ;DECREMENT RESULT.PTR |
| 9969 | 031106 | 000720 | | | BR ESISRC | |
| 9970 | | | | | | |
| 9971 | 031110 | 000137 | 031706 | | EFILZ: JMP EFILLZ | |
| 9972 | 031114 | | | | EDETSN: | ;DETERMINE SIGN & STORE WITH RESULT |
| 9973 | | | | | | ;NOTE: THERE EXIST TWO CASES IN WHICH THE DST SIGN |
| 9974 | | | | | | ; WILL DIFFER FROM THE SRC SIGN. THESE CASES ARE |
| 9975 | | | | | | ; WHEN: |
| 9976 | | | | | | ; 1)SRC SIGN = -,SHIFT=RIGHT,AND RESULT(ETNZI)=0 |

| Line | Source | Target | Address | Value | Instruction | Comment |
|-------|--------|--------|---------|--------|--------------------|---|
| 9977 | | | | | | |
| 9978 | 031114 | 013700 | 017122 | | MOV E1,R0 | : 2)SRC SIGN = -,SHIFT=LEFT,AND SRC MAGNITUDE=0 |
| 9979 | 031120 | 013701 | 017120 | | MOV E0,R1 | :SETUP SRC ADR |
| 9980 | 031124 | 013737 | 017120 | 024664 | MOV E0,ELSD | :SETUP PTR TO SIGN |
| 9981 | 031132 | 013737 | 031672 | 024646 | MOV EODDS,EODD | :SETUP PTR TO LEAST SIGN DIGIT |
| 9982 | 031140 | 004737 | 021126 | | JSR PC,ESNK | :CALL ROUTINE TO FIND SRC SIGN |
| 9983 | | | | | | :SIGN RETURNED IN ERSNEG |
| 9984 | 031144 | 005737 | 031704 | | TST ETNZI | :NON ZERO INDICATOR SET? |
| 9985 | 031150 | 001035 | | | BNE 1\$ | :BRANCH IF YES |
| 9986 | 031152 | 032737 | 000200 | 017130 | BIT #200,E4 | :SHIFT RIGHT? |
| 9987 | 031160 | 001403 | | | BEQ 2\$ | :NO |
| 9988 | 031162 | 005037 | 024616 | | CLR ERSNEG | :MAKE SIGN POSITIVE (CASE 1 NOTED ABOVE) |
| 9989 | 031166 | 000426 | | | BR 1\$ | |
| 9990 | 031170 | 105737 | 017120 | | TSTB E0 | :SRC MAGNITUDE = 0 |
| 9991 | 031174 | 001421 | | | BEQ 4\$ | :BRANCH IF YES (SRC.LEN = 0) |
| 9992 | 031176 | 013700 | 017122 | | MOV E1,R0 | :SETUP SRC.ADR |
| 9993 | 031202 | 013701 | 017120 | | MOV E0,R1 | :SETUP PTR TO SIGN |
| 9994 | 031206 | 013737 | 017120 | 024664 | MOV E0,ELSD | :SETUP PTR TO LEAST SIGN DIGIT |
| 9995 | 031214 | 105301 | | | DEC B R1 | |
| 9996 | 031216 | 013737 | 031672 | 024646 | MOV EODDS,EODD | :FIND SRC DIGIT |
| 9997 | 031224 | 004737 | 021126 | | JSR PC,ESNK | |
| 9998 | 031230 | 005702 | | | TST R2 | :IS DIGIT = 0? |
| 9999 | 031232 | 001004 | | | BNE 1\$ | :BRANCH IF NO |
| 10000 | 031234 | 105701 | | | TSTB R1 | :ALL DIGITS IN SRC TESTED? |
| 10001 | 031236 | 001366 | | | BNE 3\$ | :BRANCH IF NO |
| 10002 | 031240 | 005037 | 024616 | | CLR ERSNEG | :CASE 2 NOTED ABOVE |
| 10003 | 031244 | | | | 1\$: | :STORE SIGN WITH RESULT |
| 10004 | 031244 | 012700 | 025612 | | MOV #EVRTAB,R0 | |
| 10005 | 031250 | 013701 | 032632 | | MOV ETLSD,R1 | |
| 10006 | 031254 | 013737 | 032632 | 024664 | MOV ETLSD,ELSD | |
| 10007 | 031262 | 012737 | 177777 | 024646 | MOV #177777,EODD | |
| 10008 | 031270 | 004737 | 022174 | | JSR PC,EPUSH | |
| 10009 | | | | | | |
| 10010 | 031274 | | | | ECRID: | :COPY RESULT INTO DESTINATION |
| 10011 | 031274 | 005037 | 031702 | | CLR ENZI | :CLEAR NON ZERO INDICATOR |
| 10012 | 031300 | 113701 | 017124 | | MOV B E2,R1 | : OF RESULT TO COPY. |
| 10013 | 031304 | 005737 | 026056 | | TST EPAK | :PACKED INST? |
| 10014 | 031310 | 001405 | | | BEQ 1\$ | :BRANCH IF NO |
| 10015 | 031312 | 012704 | 025732 | | MOV #EVRTAB+120,R4 | |
| 10016 | 031316 | 006201 | | | ASR R1 | :CALCULATE # OF BYTES OCCUPIED BY DST |
| 10017 | 031320 | 005201 | | | INC R1 | : # OF BYTES=# OF DIGITS/2 +1 |
| 10018 | 031322 | 000402 | | | BR 4\$ | |
| 10019 | 031324 | 012704 | 026051 | | 1\$: | MOV #EVRTAB+237,R4 |
| 10020 | 031330 | 160104 | | | 4\$: | SUB R1,R4 |
| 10021 | 031332 | 013701 | 017124 | | MOV E2,R1 | :SETUP POSITION POINTER TO ZERO |
| 10022 | 031336 | 042701 | 000377 | | BIC #377,R1 | |
| 10023 | 031342 | 010103 | | | MOV R1,R3 | :SAVE DATA TYPE FIELD |
| 10024 | 031344 | 010400 | | | 2\$: | MOV R4,R0 |
| 10025 | 031346 | 013737 | 017124 | 024664 | MOV E2,ELSD | :SETUP 'FROM' ADDRESS |
| 10026 | 031354 | 013737 | 031674 | 024646 | MOV EODDD,EODD | :SETUP POINTER TO LEAST SIGN DIGIT |
| 10027 | 031362 | 005737 | 026056 | | TST EPAK | :ZONED RESULT? |
| 10028 | 031366 | 001007 | | | BNE 5\$ | :BRANCH IN NO |
| 10029 | 031370 | 042701 | 177400 | | BIC #177400,R1 | :ZONED RESULT DATA TYPE = TRAILING SEPARATE |
| 10030 | 031374 | 052701 | 040000 | | BIS #40000,R1 | |

| | | | | | | | |
|-------|--------|--------|--------|--------|-------|-------------------|---|
| 10031 | 031400 | 112737 | 000100 | 024665 | | MOV B #100,ELSD+1 | |
| 10032 | 031406 | 004737 | 021126 | | 5\$: | JSR PC,ESNK | :CALL ROUTINE TO GET RESULT DIGIT |
| 10033 | 031412 | 013700 | 017126 | | | MOV E3,R0 | :SETUP 'TO' ADDRESS (I.E. DST) |
| 10034 | 031416 | 105037 | 024665 | | | CLRB ELSD+1 | |
| 10035 | 031422 | 050337 | 024664 | | | BIS R3,ELSD | |
| 10036 | 031426 | 042701 | 177400 | | | BIC #177400,R1 | |
| 10037 | 031432 | 050301 | | | | BIS R3,R1 | |
| 10038 | 031434 | 004737 | 022174 | | | JSR PC,EPUSH | :CALL ROUTINE TO PUSH RESULT DIGIT INTO DST |
| 10039 | 031440 | 005702 | | | | TST R2 | :WAS DIGIT PUSHED NON ZERO? |
| 10040 | 031442 | 001403 | | | | BEQ 3\$ | :BRANCH IF NO |
| 10041 | 031444 | 012737 | 177777 | 031702 | | MOV #177777,ENZI | :SET NON ZERO INDICATOR |
| 10042 | 031452 | 020137 | 017124 | | 3\$: | CMP R1,E2 | :COPY DONE? |
| 10043 | 031456 | 001402 | | | | BEQ ESCC | :BRANCH IF YES |
| 10044 | 031460 | 005201 | | | | INC R1 | :UPDATE PTR AND RETURN TO COPY NXT DIGIT |
| 10045 | 031462 | 000730 | | | | BR 2\$ | |
| 10046 | | | | | | | |
| 10047 | 031464 | | | | | | |
| 10048 | 031464 | 005077 | 165426 | | ESCC: | CLR @EOPSW | :SET CONDITION CODES |
| 10049 | 031470 | 005737 | 031702 | | | TST ENZI | :RESET EMULATION PSW |
| 10050 | 031474 | 001004 | | | | BNE 1\$ | :SET Z BIT IF NON ZERO INDICATOR=0. |
| 10051 | 031476 | 052777 | 000004 | 165412 | | BIS #4,@EOPSW | |
| 10052 | 031504 | 000406 | | | | BR 2\$ | |
| 10053 | 031506 | 005737 | 024616 | | 1\$: | TST ERSNEG | :IS SIGN NEGATIVE? |
| 10054 | 031512 | 001403 | | | | BEQ 2\$ | :BRANCH IF NO |
| 10055 | 031514 | 052777 | 000010 | 165374 | | BIS #10,@EOPSW | :SET N BIT |
| 10056 | 031522 | | | | 2\$: | | :DETERMINE V BIT |
| 10057 | 031522 | 113700 | 017124 | | | MOV B E2,R0 | :CALCULATE # OF DIGITS OF RESULT |
| 10058 | 031526 | 012704 | 000237 | | | MOV #237,R4 | : THAT WOULD NOT FIT IN DEST. |
| 10059 | 031532 | 160004 | | | | SUB R0,R4 | |
| 10060 | 031534 | 001437 | | | | BEQ 6\$ | |
| 10061 | 031536 | 005304 | | | | DEC R4 | |
| 10062 | 031540 | 012700 | 025612 | | | MOV #EVRTAB,R0 | :YES - WERE ANY SIGNIFICANT DIGITS |
| 10063 | 031544 | 013701 | 017124 | | | MOV E2,R1 | : NOT STORED? |
| 10064 | 031550 | 042701 | 000377 | | | BIC #377,R1 | |
| 10065 | 031554 | 005737 | 026056 | | | TST EPAK | :ZONED RESULT? |
| 10066 | 031560 | 001004 | | | | BNE 3\$ | :BRANCH IF NO |
| 10067 | 031562 | 042701 | 177400 | | | BIC #177400,R1 | :ZONED RESULT DATA TYPE =TRAILING SEPARATE |
| 10068 | 031566 | 052701 | 040000 | | | BIS #40000,R1 | |
| 10069 | 031572 | 012737 | 177777 | 024646 | 3\$: | MOV #177777,EODD | |
| 10070 | 031600 | 013737 | 032632 | 024664 | | MOV ETLSD,ELSD | |
| 10071 | 031606 | 004737 | 021126 | | 5\$: | JSR PC,ESNK | : CALL ROUTINE TO FIND RESULT DIGIT |
| 10072 | 031612 | 005702 | | | | TST R2 | : NOT STORED. |
| 10073 | 031614 | 001004 | | | | BNE 4\$ | :GO SET OVERFLOW - V BIT |
| 10074 | 031616 | 120104 | | | | CMPB R1,R4 | |
| 10075 | 031620 | 001405 | | | | BEQ 6\$ | |
| 10076 | 031622 | 005201 | | | | INC R1 | |
| 10077 | 031624 | 000770 | | | | BR 5\$ | |
| 10078 | 031626 | 052777 | 000002 | 165262 | 4\$: | BIS #2,@EOPSW | :SET V BIT |
| 10079 | 031634 | 013702 | 017114 | | 6\$: | MOV EORSTK,R2 | :REGISTER UNLOAD |
| 10080 | 031640 | 005022 | | | | CLR (R2)+ | :R0=0 |
| 10081 | 031642 | 005022 | | | | CLR (R2)+ | :R1=0 |
| 10082 | 031644 | 013722 | 017124 | | | MOV E2,(R2)+ | :R2=R2 |
| 10083 | 031650 | 013722 | 017126 | | | MOV E3,(R2)+ | :R3=R3 |
| 10084 | 031654 | 005022 | | | | CLR (R2)+ | :R4=0 |

| | | | | | | | |
|-------|--------|--------|--------|---------|-------------------|--|---|
| 10085 | 031656 | 013722 | 017132 | | MOV E5,(R2)+ | | :R5=R5 |
| 10086 | 031662 | 000207 | | | RTS PC | | |
| 10087 | | | | | | | |
| 10088 | 031664 | 000000 | | ETMPRO: | .WORD 0 | | |
| 10089 | 031666 | 000000 | | ETMPR1: | .WORD 0 | | |
| 10090 | 031670 | 000000 | | ETMPR2: | .WORD 0 | | |
| 10091 | 031672 | 000000 | | EODDS: | .WORD 0 | | |
| 10092 | 031674 | 000000 | | EODDD: | .WORD 0 | | |
| 10093 | 031676 | 000000 | | ESGN: | .WORD 0 | | |
| 10094 | 031700 | 000000 | | ENCC: | .WORD 0 | | |
| 10095 | 031702 | 000000 | | ENZI: | .WORD 0 | | |
| 10096 | 031704 | 000000 | | ETNZI: | .WORD 0 | | |
| 10097 | | | | | | | |
| 10098 | 031706 | | | EFILLZ: | | | :PAD REMAINING RESULT WITH ZERO DIGITS |
| 10099 | 031706 | 122704 | 000377 | | CMPB #377,R4 | | :IS RESULT.PTR <0? |
| 10100 | 031712 | 001426 | | | BEQ 1\$ | | :IF YES BRANCH |
| 10101 | 031714 | 012700 | 025612 | | MOV #EVRTAB,R0 | | :PUSH A ZERO DIGIT INTO RESULT. |
| 10102 | 031720 | 010401 | | | MOV R4,R1 | | :SET R0=RESULT.ADR, R1=RESULT.PTR, R2=0 |
| 10103 | 031722 | 013702 | 024622 | | MOV ECARRY,R2 | | |
| 10104 | 031726 | 013737 | 032632 | 024664 | MOV ETLSD,ELSD | | |
| 10105 | 031734 | 012737 | 177777 | 024646 | MOV #177777,EODD | | |
| 10106 | 031742 | 004737 | 022174 | | JSR PC,EPUSH | | :CALL ROUTINE TO PUSH ZERO + CARRY |
| 10107 | | | | | | | : DIGIT INTO RESULT. |
| 10108 | 031746 | 005702 | | | TST R2 | | :WAS DIGIT PUSHED = 0? |
| 10109 | 031750 | 001403 | | | BEQ 2\$ | | :BRANCH IF YES |
| 10110 | 031752 | 012737 | 177777 | 031704 | MOV #177777,ETNZI | | :DIGIT PUSHED NOT = 0;SET NONZERO INDICATOR |
| 10111 | 031760 | 105304 | | 2\$: | DECB R4 | | |
| 10112 | 031762 | 005037 | 024622 | | CLR ECARRY | | |
| 10113 | 031766 | 000747 | | | BR EFILLZ | | |
| 10114 | 031770 | 000137 | 031114 | 1\$: | JMP EDETSN | | |

| | | | | | | | |
|-------|--------|--------|--------|--------|--------|--------------------|--------------------------------------|
| 10116 | | | | | | .SBTTL | CMPP,CMPN INSTRUCTIONS |
| 10117 | 031774 | 012737 | 177777 | 026056 | ECMPP: | MOV #177777,EPAK | :INDICATE PACKED MODE |
| 10118 | 032002 | 005037 | 032622 | | | CLR EODDS1 | |
| 10119 | 032006 | 005037 | 032624 | | | CLR EODDS2 | |
| 10120 | 032012 | 032737 | 000001 | 017120 | | BIT #1,E0 | :IS SRC NUMBER ODD LENGTH? |
| 10121 | 032020 | 001403 | | | | BEQ 1\$ | |
| 10122 | 032022 | 012737 | 177777 | 032622 | | MOV #177777,EODDS1 | :YES - SET ODD INDICATOR |
| 10123 | 032030 | 032737 | 000001 | 017124 | 1\$: | BIT #1,E2 | :IS SRC2 ODD LENGTH? |
| 10124 | 032036 | 001406 | | | | BEQ ECMP | |
| 10125 | 032040 | 012737 | 177777 | 032624 | | MOV #177777,EODDS2 | :YES SET ODD INDICATOR |
| 10126 | 032046 | 000402 | | | | BR ECMP | |
| 10127 | 032050 | 005037 | 026056 | | ECMPN: | CLR EPAK | :INDICATE ZONED MODE |
| 10128 | | | | | | | |
| 10129 | 032054 | 013703 | 017120 | | ECMP: | MOV E0,R3 | :INITIALIZE SRC.PTY TO 0 |
| 10130 | 032060 | 042703 | 000377 | | | BIC #377,R3 | |
| 10131 | 032064 | 013704 | 017124 | | | MOV E2,R4 | :INITIALIZE DST.PTR TO 0 |
| 10132 | 032070 | 042704 | 000377 | | | BIC #377,R4 | |
| 10133 | 032074 | 010337 | 032626 | | | MOV R3,ELS1M | |
| 10134 | 032100 | 010437 | 032630 | | | MOV R4,ELS2M | |
| 10135 | 032104 | 013700 | 017122 | | 1\$: | MOV E1,R0 | :FIND MOST SIGNIFICANT DIGIT IN SRC1 |
| 10136 | 032110 | 010301 | | | | MOV R3,R1 | :SET R0=SRC1.ADR, R1=SRC1.PTR |
| 10137 | 032112 | 120337 | 017120 | | | CMPB R3,E0 | :REACH END OF SRC1 STRING? |
| 10138 | 032116 | 001416 | | | | BEQ 2\$ | :IF YES BRANCH |
| 10139 | 032120 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 10140 | 032126 | 013737 | 032622 | 024646 | | MOV EODDS1,EODD | |
| 10141 | 032134 | 004737 | 021126 | | | JSR PC,ESNK | :CALL ROUTINE TO FIND SRC DIGIT |
| 10142 | 032140 | 005702 | | | | TST R2 | :IS SRC1 DIGIT=0? |
| 10143 | 032142 | 001004 | | | | BNE 2\$ | :IF NO BRANCH |
| 10144 | 032144 | 005203 | | | | INC R3 | :UPDATE SRC1.PTR TO NEXT DIGIT |
| 10145 | 032146 | 010337 | 032626 | | | MOV R3,ELS1M | :SAVE SRC1.PTR |
| 10146 | 032152 | 000754 | | | | BR 1\$ | |
| 10147 | 032154 | 013705 | 017120 | | 2\$: | MOV E0,R5 | :CALCULATE # OF SIGN DIGITS IN SRC1 |
| 10148 | 032160 | 160305 | | | | SUB R3,R5 | :SAVE # OF SIGN DIGITS IN ES1NSD |
| 10149 | 032162 | 010537 | 032616 | | | MOV R5,ES1NSD | :FIND MOST SIGNIFICANT DIGIT IN SRC2 |
| 10150 | 032166 | 013700 | 017126 | | 3\$: | MOV E3,R0 | :SET R0=SRC2.ADR, R1=SRC2.PTR |
| 10151 | 032172 | 010401 | | | | MOV R4,R1 | :REACH END OF SRC2 STRING? |
| 10152 | 032174 | 120437 | 017124 | | | CMPB R4,E2 | :IF YES BRANCH |
| 10153 | 032200 | 001416 | | | | BEQ 4\$ | |
| 10154 | 032202 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 10155 | 032210 | 013737 | 032624 | 024646 | | MOV EODDS2,EODD | |
| 10156 | 032216 | 004737 | 021126 | | | JSR PC,ESNK | :CALL ROUTINE TO FIND SRC DIGIT |
| 10157 | 032222 | 005702 | | | | TST R2 | :IS SRC2 DIGIT=0? |
| 10158 | 032224 | 001004 | | | | BNE 4\$ | :IF NO BRANCH |
| 10159 | 032226 | 005204 | | | | INC R4 | :UPDATE SRC2.PTR TO NEXT DIGIT |
| 10160 | 032230 | 010437 | 032630 | | | MOV R4,ELS2M | :SAVE SRC2.PTR |
| 10161 | 032234 | 000754 | | | | BR 3\$ | |
| 10162 | 032236 | 013705 | 017124 | | 4\$: | MOV E2,R5 | :CALCULATE # OF SIGN DIGITS IN SRC2 |
| 10163 | 032242 | 160405 | | | | SUB R4,R5 | :SAVE # OF SIGN DIGITS IN ES2NSD |
| 10164 | 032244 | 010537 | 032620 | | | MOV R5,ES2NSD | |
| 10165 | | | | | | | |
| 10166 | 032250 | | | | ECNSD: | | :COMPARE # OF SIGN DIGITS IN SRC1 |
| 10167 | 032250 | 005077 | 164642 | | | CLR @EOPSW | : VERSUS SRC2 |
| 10168 | 032254 | 023737 | 032616 | 032620 | | CMP ES1NSD,ES2NSD | |
| 10169 | 032262 | 101052 | | | | BHI 5\$ | :BRANCH IF SRC1 HAS MORE SIGN DIGITS |

| | | | | | | | |
|-------|--------|--------|--------|--------|--|-----------------|---|
| 10170 | 032264 | 103457 | | | | BLO 6\$ | :BRANCH IF SRC2 HAS MORE SIGN DIGITS |
| 10171 | 032266 | 005737 | 032616 | | | TST ES1NSD | :SRC1 & SRC2 CONTAIN THE SAME # |
| 10172 | | | | | | | : OF SIGNIFICANT DIGITS |
| 10173 | 032272 | 001500 | | | | BEQ 3\$ | :BOTH SRC'S CONTAIN NO SIGNIFICANT DIGITS |
| 10174 | 032274 | 013703 | 032626 | | | MOV ELS1M,R3 | :SETUP SRC1.PTR |
| 10175 | 032300 | 013704 | 032630 | | | MOV ELS2M,R4 | :SETUP SRC2.PTR |
| 10176 | 032304 | 013700 | 017122 | | | MOV E1,R0 | :GET A SRC1 DIGIT |
| 10177 | 032310 | 010301 | | | | MOV R3,R1 | |
| 10178 | 032312 | 013737 | 017120 | 024664 | | MOV E0,ELSD | |
| 10179 | 032320 | 013737 | 032622 | 024646 | | MOV EODDS1,EODD | |
| 10180 | 032326 | 004737 | 021126 | | | JSR PC,ESNK | |
| 10181 | 032332 | 010237 | 032614 | | | MOV R2,ES1D | :SAVE SRC1 DIGIT |
| 10182 | 032336 | 013700 | 017126 | | | MOV E3,R0 | :GET A SRC2 DIGIT |
| 10183 | 032342 | 010401 | | | | MOV R4,R1 | |
| 10184 | 032344 | 013737 | 017124 | 024664 | | MOV E2,ELSD | |
| 10185 | 032352 | 013737 | 032624 | 024646 | | MOV EODDS2,EODD | |
| 10186 | 032360 | 004737 | 021126 | | | JSR PC,ESNK | |
| 10187 | 032364 | 023702 | 032614 | | | CMP ES1D,R2 | :COMPARE DIGITS |
| 10188 | 032370 | 101007 | | | | BHI 5\$ | :BRANCH IF SRC1 DIGIT IS BIGGER |
| 10189 | 032372 | 103414 | | | | BLO 6\$ | :BRANCH IF SRC2 DIGIT IS BIGGER |
| 10190 | 032374 | 005203 | | | | INC R3 | :DIGITS EQUAL - ALL DIGITS CHECKED? |
| 10191 | 032376 | 120337 | 032616 | | | CMPB R3,ES1NSD | |
| 10192 | 032402 | 001416 | | | | BEQ 7\$ | :BRANCH IF ALL CHECKED - I.E. ALL |
| 10193 | | | | | | | : DIGITS ARE EQUAL |
| 10194 | 032404 | 005204 | | | | INC R4 | |
| 10195 | 032406 | 000736 | | | | BR 4\$ | |
| 10196 | 032410 | 004737 | 032530 | | | JSR PC,EGS1S | :CHECK SIGN OF SRC1 |
| 10197 | 032414 | 005737 | 024616 | | | TST ERSNEG | :IS SRC1 NEG? |
| 10198 | 032420 | 001021 | | | | BNE 2\$ | :BRANCH IF YES |
| 10199 | 032422 | 000427 | | | | BR 1\$ | :SRC1 IS POSITIVE |
| 10200 | | | | | | | |
| 10201 | 032424 | 004737 | 032562 | | | JSR PC,EGS2S | :CHECK SIGN OF SRC2 |
| 10202 | 032430 | 005737 | 024616 | | | TST ERSNEG | :IS SRC2 NEGATIVE? |
| 10203 | 032434 | 001022 | | | | BNE 1\$ | :BRANCH IF YES |
| 10204 | 032436 | 000412 | | | | BR 2\$ | :SRC2 IS POSITIVE |
| 10205 | | | | | | | |
| 10206 | 032440 | | | | | | :COMPARE SIGNS |
| 10207 | 032440 | 004737 | 032530 | | | JSR PC,EGS1S | :GET SRC1 SIGN |
| 10208 | 032444 | 013705 | 024616 | | | MOV ERSNEG,R5 | :SAVE IT IN R5 |
| 10209 | 032450 | 004737 | 032562 | | | JSR PC,EGS2S | :GET SRC2 SIGN |
| 10210 | 032454 | 023705 | 024616 | | | CMP ERSNEG,R5 | :SIGNS =? |
| 10211 | 032460 | 001405 | | | | BEQ 3\$ | :BRANCH IF YES |
| 10212 | 032462 | 000752 | | | | BR 5\$ | :SIGNS NOT EQUAL |
| 10213 | | | | | | | |
| 10214 | 032464 | 052777 | 000010 | 164424 | | BIS #10,@EOPSW | :SRC2>SRC1 SET N BIT. |
| 10215 | 032472 | 000403 | | | | BR 1\$ | |
| 10216 | 032474 | 052777 | 000004 | 164414 | | BIS #4,@EOPSW | :SRC2=SRC1 SET Z BIT |
| 10217 | 032502 | 013702 | 017114 | | | MOV EORSTK,R2 | :REGISTER UNLOAD |
| 10218 | 032506 | 005022 | | | | CLR (R2)+ | :R0=0 |
| 10219 | 032510 | 005022 | | | | CLR (R2)+ | :R1=0 |
| 10220 | 032512 | 005022 | | | | CLR (R2)+ | :R2=0 |
| 10221 | 032514 | 005022 | | | | CLR (R2)+ | :R3=0 |
| 10222 | 032516 | 013722 | 017130 | | | MOV E4,(R2)+ | :R4=R4 |
| 10223 | 032522 | 013722 | 017132 | | | MOV E5,(R2)+ | :R5=R5 |

```
10224 032526 000207          RTS PC
10225
10226 032530 013700 017122          EGS1S:      MOV E1,R0          ;ROUTINE TO GET SRC1 SIGN
10227 032534 013701 017120          MOV E0,R1          ;SET R0=SRC1.ADR,R1=SRC1.PTR
10228 032540 013737 017120 024664      MOV E0,ELSD
10229 032546 013737 032622 024646      MOV EODDS1,EODD
10230 032554 004737 021126          JSR PC,ESNK        ;CALL ROUTINE TO FIND SRC1 SIGN
10231 032560 000207          RTS PC
10232
10233 032562 013700 017126          EGS2S:      MOV E3,R0          ;ROUTINE TO GET SRC2 SIGN
10234 032566 013701 017124          MOV E2,R1          ;SET R0=SRC2.ADR,R1=SRC2.PTR
10235 032572 013737 017124 024664      MOV E2,ELSD
10236 032600 013737 032624 024646      MOV EODDS2,EODD
10237 032606 004737 021126          JSR PC,ESNK        ;CALL ROUTINE TO FIND SRC2 SIGN
10238 032612 000207          RTS PC
10239
10240 032614 000000          ES1D:        .WORD 0
10241 032616 000000          ES1NSD:     .WORD 0
10242 032620 000000          ES2NSD:     .WORD 0
10243 032622 000000          EODDS1:     .WORD 0
10244 032624 000000          EODDS2:     .WORD 0
10245 032626 000000          ELS1M:      .WORD 0
10246 032630 000000          ELS2M:      .WORD 0
10247 032632 000000          ETLSD:      .WORD 0
```

| | | | | | | |
|-------|--------|--------|--------|--------|--------------------|--|
| 10249 | | | | | .SBTTL | MULP INSTRUCTION |
| 10250 | | | | | | |
| 10251 | 032634 | 004737 | 033676 | | EMULP: | :SAVE MULP CALL PARAMETERS |
| 10252 | 032640 | 005037 | 034664 | | JSR PC,ERSAV | :INITIALIZE MULP V-BIT RESULT |
| 10253 | 032644 | 004737 | 033760 | | CLR EMVBR | :INITIALIZE TEMPORARY RESULT BUFFER (ERT2) TO 0+ |
| 10254 | 032650 | 004737 | 034034 | | JSR PC,EIRT2 | :IS MULP SRC1=0? |
| 10255 | 032654 | 032737 | 000004 | 034344 | JSR PC,ETSTS1 | |
| 10256 | 032662 | 001147 | | | BIT #4,ETOPSW | |
| 10257 | 032664 | 004737 | 034104 | | BNE EPMID | |
| 10258 | 032670 | 032737 | 000004 | 034344 | JSR PC,ETSTS2 | :IS MULP SRC2=0? |
| 10259 | 032676 | 001141 | | | BIT #4,ETOPSW | |
| 10260 | | | | | BNE EPMID | |
| 10261 | 032700 | 005037 | 017124 | | CLR E2 | :FORM 1X,2X,3X, ETC TABLE |
| 10262 | 032704 | 012737 | 034306 | 017126 | MOV #EZDSC,E3 | : USE ADDP - SRC1=MULT.SRC1 |
| 10263 | 032712 | 012737 | 034614 | 034662 | MOV #EXTBP,EVXTBP | : SRC2=PREVIOUS ADDP DST |
| 10264 | 032720 | 012737 | 034636 | 034660 | MOV #EXTVB,EVXTVB | : DST=E(N)XT TABLE |
| 10265 | 032726 | 013737 | 034330 | 017120 | MOV ESR0,E0 | |
| 10266 | 032734 | 013737 | 034332 | 017122 | MOV ESR1,E1 | |
| 10267 | 032742 | 012737 | 000040 | 017130 | MOV #40,E4 | |
| 10268 | 032750 | 017737 | 001706 | 017132 | MOV @EVXTBP,E5 | |
| 10269 | 032756 | 012737 | 034310 | 017114 | MOV #ETRSTK,EORSTK | |
| 10270 | 032764 | 012737 | 034344 | 017116 | MOV #ETOPSW,EOPSW | |
| 10271 | 032772 | 004737 | 023136 | | JSR PC,EADDP | |
| 10272 | 032776 | 042737 | 177775 | 034344 | BIC #177775,ETOPSW | :CLEAR ALL BUT V BIT FROM ADDP |
| 10273 | 033004 | 013777 | 034344 | 001646 | MOV ETOPSW,@EVXTVB | : RESULT PSW; SAVE V BITS IN TABLE. |
| 10274 | 033012 | 027727 | 001644 | 034346 | CMP @EVXTBP,#E1XT | :FIRST TABLE ENTRY FORMATION? |
| 10275 | 033020 | 001041 | | | BNE 2\$ | :BRANCH IF NO |
| 10276 | 033022 | 032737 | 000001 | 034334 | BIT #1,ESR2 | :WORK ON TABLE ENTRY SIGNS |
| 10277 | 033030 | 001404 | | | BEQ 31\$ | |
| 10278 | 033032 | 012737 | 177777 | 024646 | MOV #177777,EODD | |
| 10279 | 033040 | 000402 | | | BR 3\$ | |
| 10280 | 033042 | 005037 | 024646 | | CLR EODD | |
| 10281 | 033046 | 013700 | 034336 | | MOV ESR3,R0 | |
| 10282 | 033052 | 013701 | 034334 | | MOV ESR2,R1 | |
| 10283 | 033056 | 013737 | 034334 | 024664 | MOV ESR2,ELSD | |
| 10284 | 033064 | 004737 | 021126 | | JSR PC,ESNK | :CALL ROUTINE TO GET MULP.SRC2 SIGN |
| 10285 | 033070 | 005737 | 024616 | | TST ERSNEG | :IF THIS SIGN IS POSITIVE THEN |
| 10286 | 033074 | 001413 | | | BEQ 2\$ | : LEAVE TABLE SIGNS = MULP.SRC1 SIGN |
| 10287 | 033076 | 132737 | 000001 | 034366 | BITB #1,E1XT+20 | :MULP.SRC2 SIGN = NEG; MAKE TABLE |
| 10288 | 033104 | 001404 | | | BEQ 4\$ | : SIGNS = COMPLEMENT OF MULP.SRC1 SIGN |
| 10289 | 033106 | 142737 | 000001 | 034366 | BICB #1,E1XT+20 | |
| 10290 | 033114 | 000403 | | | BR 2\$ | |
| 10291 | 033116 | 152737 | 000001 | 034366 | BISB #1,E1XT+20 | |
| 10292 | 033124 | 027727 | 001532 | 034566 | CMP @EVXTBP,#E9XT | :ALL TABLE ENTRIES FORMED? |
| 10293 | 033132 | 001424 | | | BEQ EISP | :BRANCH IF YES |
| 10294 | 033134 | 012737 | 000040 | 017120 | MOV #40,E0 | |
| 10295 | 033142 | 012737 | 034346 | 017122 | MOV #E1XT,E1 | |
| 10296 | 033150 | 012737 | 000040 | 017124 | MOV #40,E2 | :UPDATE ADDP SRC2 TO CURRENT |
| 10297 | 033156 | 017737 | 001500 | 017126 | MOV @EVXTBP,E3 | : DST POINTER |
| 10298 | 033164 | 062737 | 000002 | 034662 | ADD #2,EVXTBP | :UPDATE TABLE POINTERS |
| 10299 | 033172 | 062737 | 000002 | 034660 | ADD #2,EVXTVB | |
| 10300 | 033200 | 000660 | | | BR 1\$ | :RETURN TO FORM NEXT ENTRY. |
| 10301 | | | | | | |
| 10302 | 033202 | 000565 | | | EPMID: | BR EMID |

| | | | | | | | | | |
|-------|--------|--------|--------|--------|-------|----------------------|--|--|--|
| 10303 | | | | | | | | | |
| 10304 | 033204 | 005037 | 034610 | | EISP: | CLR ESPOS | | :INITIALIZE SHIFT POSITION TO ZERO | |
| 10305 | 033210 | 032737 | 000001 | 034334 | | BIT #1,ESR2 | | | |
| 10306 | 033216 | 001404 | | | | BEQ 10\$ | | | |
| 10307 | 033220 | 012737 | 177777 | 024646 | | MOV #177777,EODD | | | |
| 10308 | 033226 | 000402 | | | | BR 11\$ | | | |
| 10309 | 033230 | 005037 | 024646 | | | 10\$: CLR EODD | | | |
| 10310 | 033234 | 013737 | 024646 | 034666 | | 11\$: MOV EODD,SEODD | | :SAVE EODD | |
| 10311 | 033242 | 013737 | 034334 | 034612 | | MOV ESR2,EMS2D | | :INITIALIZE MULP.SRC2 POINTER | |
| 10312 | 033250 | 013737 | 034666 | 024646 | | 1\$: MOV SEODD,EODD | | :RESTORE EODD | |
| 10313 | 033256 | 105337 | 034612 | | | DECB EMS2D | | | |
| 10314 | 033262 | 100535 | | | | BMI EMID | | :BRANCH IF NO MORE MULP.SRC2 DIGITS TO WORK ON | |
| 10315 | 033264 | 013700 | 034336 | | | MOV ESR3,R0 | | | |
| 10316 | 033270 | 013737 | 034334 | 024664 | | MOV ESR2,ELSD | | | |
| 10317 | 033276 | 013701 | 034612 | | | MOV EMS2D,R1 | | | |
| 10318 | 033302 | 004737 | 021126 | | | JSR PC,ESNK | | :CALL ROUTINE TO GET NEXT SRC2 DIGIT | |
| 10319 | 033306 | 005702 | | | | TST R2 | | | |
| 10320 | 033310 | 001517 | | | | BEQ 2\$ | | :BRANCH IF DIGIT = 0 | |
| 10321 | 033312 | 005302 | | | | DEC R2 | | | |
| 10322 | 033314 | 010237 | 034660 | | | MOV R2,EVXTVB | | | |
| 10323 | 033320 | 006337 | 034660 | | | ASL EVXTVB | | | |
| 10324 | 033324 | 062737 | 034636 | 034660 | | ADD #EXTVB,EVXTVB | | :SETUP POINTER INTO V-BIT TABLE | |
| 10325 | 033332 | 010237 | 034662 | | | MOV R2,EVXTBP | | :INDEX INTO 1X,2X,ETC TABLE USING | |
| 10326 | 033336 | 006337 | 034662 | | | ASL EVXTBP | | : NEXT SRC2 DIGIT | |
| 10327 | 033342 | 062737 | 034614 | 034662 | | ADD #EXTBP,EVXTBP | | | |
| 10328 | 033350 | 012737 | 000040 | 017120 | | MOV #40,E0 | | :MULTIPLY TABLE VALUE BY PROPER | |
| 10329 | 033356 | 017737 | 001300 | 017122 | | MOV @EVXTBP,E1 | | : POWER OF TEN INDICATOR BY SHIFT | |
| 10330 | 033364 | 013737 | 034610 | 017130 | | MOV ESPOS,E4 | | : POSITION | |
| 10331 | 033372 | 012737 | 000037 | 017124 | | MOV #37,E2 | | : USE ASHP - SRC=TABLE VALUE | |
| 10332 | 033400 | 012737 | 034154 | 017126 | | MOV #ERT1,E3 | | : SHFT.CT=SHFT.POSITION | |
| 10333 | 033406 | 012737 | 034310 | 017114 | | MOV #ETRSTK,EORSTK | | : DST=ERT1 | |
| 10334 | 033414 | 012737 | 034344 | 017116 | | MOV #ETOPSW,EOPSW | | | |
| 10335 | 033422 | 004737 | 030376 | | | JSR PC,EASHP | | | |
| 10336 | 033426 | 042737 | 177775 | 034344 | | BIC #177775,ETOPSW | | :WORK ON V BIT | |
| 10337 | 033434 | 053737 | 034344 | 034664 | | BIS ETOPSW,EMVBR | | : 'OR' ASHP V BIT WITH RESULT V BIT. | |
| 10338 | 033442 | 057737 | 001212 | 034664 | | BIS @EVXTVB,EMVBR | | : 'OR' TABLE V BIT WITH RESULT V BIT | |
| 10339 | 033450 | 012737 | 000037 | 017120 | | MOV #37,E0 | | :ADD SHIFTED VALUE TO RESULT | |
| 10340 | 033456 | 012737 | 034176 | 017122 | | MOV #ERT2,E1 | | : USE ADDP - SRC1=ERT2 | |
| 10341 | 033464 | 012737 | 000037 | 017124 | | MOV #37,E2 | | : SRC2=ERT1 | |
| 10342 | 033472 | 012737 | 034154 | 017126 | | MOV #ERT1,E3 | | : DST= ERT2 | |
| 10343 | 033500 | 012737 | 000037 | 017130 | | MOV #37,E4 | | | |
| 10344 | 033506 | 012737 | 034176 | 017132 | | MOV #ERT2,E5 | | | |
| 10345 | 033514 | 012737 | 034310 | 017114 | | MOV #ETRSTK,EORSTK | | | |
| 10346 | 033522 | 012737 | 034344 | 017116 | | MOV #ETOPSW,EOPSW | | | |
| 10347 | 033530 | 004737 | 023136 | | | JSR PC,EADDP | | | |
| 10348 | 033534 | 042737 | 177775 | 034344 | | BIC #177775,ETOPSW | | :WORK ON V BIT | |
| 10349 | 033542 | 053737 | 034344 | 034664 | | BIS ETOPSW,EMVBR | | : 'OR' ADDP V BIT WITH RESULT | |
| 10350 | | | | | | | | | |
| 10351 | 033550 | 005237 | 034610 | | | 2\$: INC ESPOS | | :INCREMENT SHIFT POSITION FOR NEXT | |
| 10352 | 033554 | 000635 | | | | BR 1\$ | | : MULP SRC2 DIGIT. | |
| 10353 | | | | | | | | | |
| 10354 | 033556 | 012737 | 000037 | 017120 | EMID: | MOV #37,E0 | | :MOVE RESULT INTO MULP DST | |
| 10355 | 033564 | 012737 | 034176 | 017122 | | MOV #ERT2,E1 | | : USE ASHP - SRC=ERT2 | |
| 10356 | 033572 | 005037 | 017130 | | | CLR E4 | | : DST=MULP.DST | |

| | | | | |
|-------|--------|--------|--------|--------|
| 10357 | 033576 | 005037 | 017132 | |
| 10358 | 033602 | 013737 | 034340 | 017124 |
| 10359 | 033610 | 013737 | 034342 | 017126 |
| 10360 | 033616 | 013737 | 034324 | 017114 |
| 10361 | 033624 | 013737 | 034326 | 017116 |
| 10362 | 033632 | 004737 | 030376 | |
| 10363 | 033636 | 053777 | 034664 | 163252 |
| 10364 | | | | |
| 10365 | | | | |
| 10366 | 033644 | 013702 | 017114 | |
| 10367 | | | | |
| 10368 | | | | |
| 10369 | 033650 | 016262 | 000004 | 000010 |
| 10370 | 033656 | 016262 | 000006 | 000012 |
| 10371 | 033664 | 005062 | 000004 | |
| 10372 | 033670 | 005062 | 000006 | |
| 10373 | 033674 | 000207 | | |

EXMD:

CLR E5
MOV ESR4,E2
MOV ESR5,E3
MOV ESOSTK,EORSTK
MOV ESOPSW,EOPSW
JSR PC,EASHP
BIS EMVBR,@EOPSW

MOV EORSTK,R2

MOV 4(R2),10(R2)
MOV 6(R2),12(R2)
CLR 4(R2)
CLR 6(R2)
RTS PC

: SHFT.CT=0

:MULP CONDITION CODE RESULTS:
: N,Z, AND C FROM LAST ASHP
: V = 'OR' OF ALL PREVIOUS V'S
:REGISTER UNLOAD - NEEDED BECAUSE REGISTER
:OUTPUTS FOR ASHP ARE R2 AND R3
:WHEREAS THOSE FOR MULP AND DIVP ARE R4 AND R5.
:R4=R2
:R5=R3
:R2=0
:R3=0
:EXIT MULP EMULATION ROUTINE

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|---------------------------------|
| 10375 | | | | | | .SBTTL | MULP/DIVP SUBROUTINES |
| 10376 | 033676 | 013737 | 017114 | 034324 | ERSAV: | MOV EORSTK,ESOSTK | :SAVE MULP/DIVP CALL PARAMETERS |
| 10377 | 033704 | 013737 | 017116 | 034326 | | MOV EOPSW,ESOPSW | :THIS PERMITS CALLING OTHER |
| 10378 | 033712 | 013737 | 017120 | 034330 | | MOV E0,ESR0 | : EMULATION ROUTINES WHILE |
| 10379 | 033720 | 013737 | 017122 | 034332 | | MOV E1,ESR1 | : IN THE MULP/DIVP EMULATION. |
| 10380 | 033726 | 013737 | 017124 | 034334 | | MOV E2,ESR2 | |
| 10381 | 033734 | 013737 | 017126 | 034336 | | MOV E3,ESR3 | |
| 10382 | 033742 | 013737 | 017130 | 034340 | | MOV E4,ESR4 | |
| 10383 | 033750 | 013737 | 017132 | 034342 | | MOV E5,ESR5 | |
| 10384 | 033756 | 000207 | | | | RTS PC | |
| 10385 | | | | | | | |
| 10386 | 033760 | 005037 | 017120 | | EIRT2: | CLR E0 | :INITIALIZE ERT2 BUFFER TO 0+ |
| 10387 | 033764 | 012737 | 034306 | 017122 | | MOV #EZDSC,E1 | : USE ASHP - SRC.LEN=0 |
| 10388 | 033772 | 005037 | 017130 | | | CLR E4 | : SHF.CT=0 |
| 10389 | 033776 | 012737 | 000037 | 017124 | | MOV #37,E2 | : DST.LEN=31 |
| 10390 | 034004 | 012737 | 034176 | 017126 | | MOV #ERT2,E3 | : DST.ADR=ERT2 |
| 10391 | 034012 | 012737 | 034310 | 017114 | | MOV #ETRSTK,EORSTK | |
| 10392 | 034020 | 012737 | 034344 | 017116 | | MOV #ETOPSW,EOPSW | |
| 10393 | 034026 | 004737 | 030376 | | | JSR PC,EASHP | |
| 10394 | 034032 | 000207 | | | | RTS PC | |
| 10395 | | | | | | | |
| 10396 | 034034 | 005037 | 017124 | | ETSTS1: | CLR E2 | :COMPARE MULP/DIVP SRC1 WITH 0. |
| 10397 | 034040 | 012737 | 034306 | 017126 | | MOV #EZDSC,E3 | : USE CMPP - SRC2.LEN=0 |
| 10398 | 034046 | 013737 | 034330 | 017120 | | MOV ESR0,E0 | : SRC1.LEN=MULP.SRC1.LEN |
| 10399 | 034054 | 013737 | 034332 | 017122 | | MOV ESR1,E1 | : SRC1.ADR=MULP.SRC1.ADR |
| 10400 | 034062 | 012737 | 034310 | 017114 | | MOV #ETRSTK,EORSTK | |
| 10401 | 034070 | 012737 | 034344 | 017116 | | MOV #ETOPSW,EOPSW | |
| 10402 | 034076 | 004737 | 031774 | | | JSR PC,ECMPP | |
| 10403 | 034102 | 000207 | | | | RTS PC | |
| 10404 | | | | | | | |
| 10405 | 034104 | 005037 | 017124 | | ETSTS2: | CLR E2 | :COMPARE MULP/DIVP SRC2 WITH 0. |
| 10406 | 034110 | 012737 | 034306 | 017126 | | MOV #EZDSC,E3 | : USE CMPP - SRC2.LEN=0 |
| 10407 | 034116 | 013737 | 034334 | 017120 | | MOV ESR2,E0 | : SRC1.LEN=MULP.SRC2.LEN |
| 10408 | 034124 | 013737 | 034336 | 017122 | | MOV ESR3,E1 | : SRC1.ADR=MULP.SRC2.ADR |
| 10409 | 034132 | 012737 | 034310 | 017114 | | MOV #ETRSTK,EORSTK | |
| 10410 | 034140 | 012737 | 034344 | 017116 | | MOV #ETOPSW,EOPSW | |
| 10411 | 034146 | 004737 | 031774 | | | JSR PC,ECMPP | |
| 10412 | 034152 | 000207 | | | | RTS PC | |
| 10413 | | | | | | | |

| | | | .SBTTL | MULP/DIVP VARIABLES AND BUFFERS |
|-------|--------|--------|---------|---------------------------------|
| 10415 | | | | |
| 10416 | | | | |
| 10417 | 034154 | 000011 | ERT1: | .BLKW 11 |
| 10418 | 034176 | 000011 | ERT2: | .BLKW 11 |
| 10419 | 034220 | 000011 | ERT3: | .BLKW 11 |
| 10420 | 034242 | 000011 | ERT4: | .BLKW 11 |
| 10421 | 034264 | 000011 | ERT5: | .BLKW 11 |
| 10422 | 034306 | 000000 | EZDSC: | .WORD 0 |
| 10423 | 034310 | 000006 | ETRSTK: | .BLKW 6 |
| 10424 | 034324 | 000000 | ESOSTK: | .WORD 0 |
| 10425 | 034326 | 000000 | ESOPSW: | .WORD 0 |
| 10426 | 034330 | 000000 | ESR0: | .WORD 0 |
| 10427 | 034332 | 000000 | ESR1: | .WORD 0 |
| 10428 | 034334 | 000000 | ESR2: | .WORD 0 |
| 10429 | 034336 | 000000 | ESR3: | .WORD 0 |
| 10430 | 034340 | 000000 | ESR4: | .WORD 0 |
| 10431 | 034342 | 000000 | ESR5: | .WORD 0 |
| 10432 | 034344 | 000000 | ETOPSW: | .WORD 0 |
| 10433 | 034346 | 000011 | E1XT: | .BLKW 11 |
| 10434 | 034370 | 000011 | E2XT: | .BLKW 11 |
| 10435 | 034412 | 000011 | E3XT: | .BLKW 11 |
| 10436 | 034434 | 000011 | E4XT: | .BLKW 11 |
| 10437 | 034456 | 000011 | E5XT: | .BLKW 11 |
| 10438 | 034500 | 000011 | E6XT: | .BLKW 11 |
| 10439 | 034522 | 000011 | E7XT: | .BLKW 11 |
| 10440 | 034544 | 000011 | E8XT: | .BLKW 11 |
| 10441 | 034566 | 000011 | E9XT: | .BLKW 11 |
| 10442 | 034610 | 000000 | ESPOS: | .WORD 0 |
| 10443 | 034612 | 000000 | EMS2D: | .WORD 0 |
| 10444 | 034614 | 034346 | EXTBP: | .WORD E1XT |
| 10445 | 034616 | 034370 | | .WORD E2XT |
| 10446 | 034620 | 034412 | | .WORD E3XT |
| 10447 | 034622 | 034434 | | .WORD E4XT |
| 10448 | 034624 | 034456 | | .WORD E5XT |
| 10449 | 034626 | 034500 | | .WORD E6XT |
| 10450 | 034630 | 034522 | | .WORD E7XT |
| 10451 | 034632 | 034544 | | .WORD E8XT |
| 10452 | 034634 | 034566 | | .WORD E9XT |
| 10453 | 034636 | 000011 | EXTVB: | .BLKW 11 |
| 10454 | 034660 | 000000 | EVXTVB: | .WORD 0 |
| 10455 | 034662 | 000000 | EVXTBP: | .WORD 0 |
| 10456 | 034664 | 000000 | EMVBR: | .WORD 0 |
| 10457 | 034666 | 000000 | SEODD: | .WORD 0 |
| 10458 | 034670 | 000000 | ESS2SN: | .WORD 0 |
| 10459 | 034672 | 000000 | ESS1SN: | .WORD 0 |
| 10460 | 034674 | 000000 | ESUBCT: | .WORD 0 |

| | | | | |
|--|--|--|--|---------------------|
| | | | | :MULP/DIVP SRC1.LEN |
| | | | | :MULP/DIVP SRC1.ADR |
| | | | | :MULP/DIVP SRC2.LEN |
| | | | | :MULP/DIVP SRC2.ADR |
| | | | | :MULP/DIVP DST.LEN |
| | | | | :MULP/DIVP DST.ADR |
| | | | | : 1 X SRC1 |
| | | | | : 2 X SRC1 |
| | | | | : 3 X SRC1 |
| | | | | : 4 X SRC1 |
| | | | | : 5 X SRC1 |
| | | | | : 6 X SRC1 |
| | | | | : 7 X SRC1 |
| | | | | : 8 X SRC1 |
| | | | | : 9 X SRC1 |

| | | | | |
|--|--|--|--|--------------|
| | | | | :V-BIT TABLE |
|--|--|--|--|--------------|

| | | | | | .SBTTL | DIVP INSTRUCTION |
|-------|--------|--------|--------|--------|---------------------|--|
| 10462 | | | | | | |
| 10463 | | | | | | |
| 10464 | 034676 | 004737 | 033676 | | EDIVP: JSR PC,ERSAV | :SAVE DIVP CALL PARAMETERS |
| 10465 | 034702 | 005037 | 035744 | | CLR EZDF | :CLEAR ZERO DIVIDE FLAG |
| 10466 | 034706 | 005037 | 034610 | | CLR ESPOS | :CLEAR SHIFT POSITION |
| 10467 | 034712 | 004737 | 033760 | | JSR PC,EIRT2 | :INITIALIZE TEMPORARY RESULT BUFFER (ERT2) TO 0+ |
| 10468 | 034716 | 004737 | 034034 | | JSR PC,ETSTS1 | :IS DIVP SRC1 = 0? |
| 10469 | 034722 | 032737 | 000004 | 034344 | BIT #4,ETOPSW | : |
| 10470 | 034730 | 001152 | | | BNE E1XZD | :BRANCH IF YES |
| 10471 | 034732 | 042737 | 177767 | 034344 | BIC #177767,ETOPSW | |
| 10472 | 034740 | 013737 | 034344 | 034672 | MOV ETOPSW,ESS1SN | :SAVE SRC1 SIGN |
| 10473 | 034746 | 004737 | 034104 | | JSR PC,ETSTS2 | :IS DIVP SRC2=0? |
| 10474 | 034752 | 032737 | 000004 | 034344 | BIT #4,ETOPSW | |
| 10475 | 034760 | 001134 | | | BNE E1DID | :BRANCH IF YES |
| 10476 | 034762 | 042737 | 177767 | 034344 | BIC #177767,ETOPSW | |
| 10477 | 034770 | 013737 | 034344 | 034670 | MOV ETOPSW,ESS2SN | :SAVE SRC2 SIGN |
| 10478 | | | | | | |
| 10479 | 034776 | 013737 | 034330 | 017120 | MOV ESRO,E0 | :MOVE DIVP SRC1 INTO ERT3 |
| 10480 | 035004 | 013737 | 034332 | 017122 | MOV ESR1,E1 | : USE ASHP - SRC = DIVP.SRC1 |
| 10481 | 035012 | 005037 | 017130 | | CLR E4 | : DST = ERT3 |
| 10482 | 035016 | 005037 | 017132 | | CLR E5 | : SHFT.CT=0 |
| 10483 | 035022 | 012737 | 000040 | 017124 | MOV #40,E2 | |
| 10484 | 035030 | 012737 | 034220 | 017126 | MOV #ERT3,E3 | |
| 10485 | 035036 | 004737 | 030376 | | JSR PC,EASHP | |
| 10486 | 035042 | 142737 | 000001 | 034240 | BICB #1,@#ERT3+20 | :MAKE ERT3 SIGN = + |
| 10487 | | | | | | |
| 10488 | 035050 | 013737 | 034334 | 017120 | MOV ESR2,E0 | :MOVE DIVP SRC2 INTO ERT4 |
| 10489 | 035056 | 013737 | 034336 | 017122 | MOV ESR3,E1 | : USE ASHP - SRC = DIVP SRC2 |
| 10490 | 035064 | 005037 | 017130 | | CLR E4 | : DST = ERT4 |
| 10491 | 035070 | 005037 | 017132 | | CLR E5 | : SHFT.CT = 0 |
| 10492 | 035074 | 012737 | 000037 | 017124 | MOV #37,E2 | |
| 10493 | 035102 | 012737 | 034242 | 017126 | MOV #ERT4,E3 | |
| 10494 | 035110 | 004737 | 030376 | | JSR PC,EASHP | |
| 10495 | 035114 | 142737 | 000001 | 034261 | BICB #1,@#ERT4+17 | :MAKE ERT4 SIGN + |
| 10496 | | | | | | |
| 10497 | 035122 | 012737 | 000040 | 017120 | EPOSS1: MOV #40,E0 | :SHIFT DIVP SRC1 LEFT UNTIL SRC2 - SRC1(SHIFTED) |
| 10498 | 035130 | 012737 | 034220 | 017122 | MOV #ERT3,E1 | : IS NEGATIVE. NOTE: LEN=40 IS LEGAL FOR EMUL. |
| 10499 | 035136 | 012737 | 000037 | 017124 | MOV #37,E2 | :SUBTRACT ERT3 FROM ERT4 |
| 10500 | 035144 | 012737 | 034242 | 017126 | MOV #ERT4,E3 | : USE SUBP - SRC1 = ERT3 |
| 10501 | 035152 | 012737 | 000037 | 017130 | MOV #37,E4 | : SRC2 = ERT4 |
| 10502 | 035160 | 012737 | 034154 | 017132 | MOV #ERT1,E5 | : DST = ERT1 |
| 10503 | 035166 | 004737 | 023154 | | JSR PC,ESUBP | |
| 10504 | 035172 | 032777 | 000010 | 161716 | BIT #10,@EOPSW | :IS RESULT NEGATIVE? |
| 10505 | 035200 | 001030 | | | BNE EPOSD | :BRANCH IF YES |
| 10506 | 035202 | 012737 | 000040 | 017120 | MOV #40,E0 | :SHIFT DIVP SRC1 LEFT 1 PLACE |
| 10507 | 035210 | 012737 | 034220 | 017122 | MOV #ERT3,E1 | : USE ASHP - SRC = ERT3 |
| 10508 | 035216 | 012737 | 000001 | 017130 | MOV #1,E4 | : DST = ERT3 |
| 10509 | 035224 | 012737 | 000040 | 017124 | MOV #40,E2 | : SHFT.CT = 1 |
| 10510 | 035232 | 012737 | 034220 | 017126 | MOV #ERT3,E3 | :NOTE - LEGAL FOR EMULATOR. |
| 10511 | 035240 | 004737 | 030376 | | JSR PC,EASHP | |
| 10512 | 035244 | 005237 | 034610 | | INC ESPOS | :INCREMENT SHIFT POSITION |
| 10513 | 035250 | 000724 | | | BR EPOSS1 | |
| 10514 | 035252 | 000137 | 035610 | | E1DID: JMP EDID | |
| 10515 | 035256 | 000137 | 035674 | | E1XZD: JMP EXZD | |

| | | | | | | | | | |
|-------|--------|--------|--------|--------|--------|-----------------------|--|---|--|
| 10516 | | | | | | | | | |
| 10517 | 035262 | 005737 | 034610 | | EPCSD: | TST ESPOS | | ;SHIFT POSITION = 0? | |
| 10518 | 035266 | 001550 | | | | BEQ EDID | | ;BRANCH IF YES | |
| 10519 | | | | | | | | | |
| 10520 | 035270 | 012737 | 000040 | 017120 | EDIVL: | MOV #40,E0 | | ;REPOSITION ERT3 BACK 1 PLACE(RIGHT) | |
| 10521 | 035276 | 012737 | 034220 | 017122 | | MOV #ERT3,E1 | | ; USE ASHP - SRC = ERT3 | |
| 10522 | 035304 | 012737 | 000377 | 017130 | | MOV #377,E4 | | ; DST = ERT3 | |
| 10523 | 035312 | 012737 | 000040 | 017124 | | MOV #40,E2 | | ; SHFT.CT=-1 | |
| 10524 | 035320 | 012737 | 034220 | 017126 | | MOV #ERT3,E3 | | ;NOTE - THIS IS LEGAL FOR EMULATOR | |
| 10525 | 035326 | 004737 | 030376 | | | JSR PC,EASHP | | | |
| 10526 | | | | | | | | | |
| 10527 | 035332 | 005037 | 034674 | | | CLR ESUBCT | | ;CLEAR SUBTRACT COUNTER | |
| 10528 | | | | | | | | | |
| 10529 | 035336 | 012737 | 000040 | 017120 | 1\$: | MOV #40,E0 | | ;SUBTRACT DIVP SRC1(SHIFTED) FROM DIVP SRC2 | |
| 10530 | 035344 | 012737 | 034220 | 017122 | | MOV #ERT3,E1 | | ; USE SUBP - SRC1 = ERT3 | |
| 10531 | 035352 | 012737 | 000037 | 017124 | | MOV #37,E2 | | ; SRC2 = ERT4 | |
| 10532 | 035360 | 012737 | 034242 | 017126 | | MOV #ERT4,E3 | | ; DST = ERT4 | |
| 10533 | 035366 | 012737 | 000037 | 017130 | | MOV #37,E4 | | | |
| 10534 | 035374 | 012737 | 034242 | 017132 | | MOV #ERT4,E5 | | | |
| 10535 | 035402 | 004737 | 023154 | | | JSR PC,ESUBP | | | |
| 10536 | 035406 | 032777 | 000010 | 161502 | | BIT #10,@EOPSW | | ;IS RESULT OF SUBP POSITIVE? | |
| 10537 | 035414 | 001003 | | | | BNE ESBTD | | ;BRANCH IF NO | |
| 10538 | 035416 | 005237 | 034674 | | | INC ESUBCT | | ;INCREMENT SUBTRACT COUNTER | |
| 10539 | 035422 | 000745 | | | | BR 1\$ | | | |
| 10540 | | | | | | | | | |
| 10541 | 035424 | 012737 | 000040 | 017120 | ESBTD: | MOV #40,EJ | | ;BACKUP TO LAST POSITIVE RESULT FROM SUBP | |
| 10542 | 035432 | 012737 | 034220 | 017122 | | MOV #ERT3,E1 | | ; USE ADDP - SRC1 = ERT3 | |
| 10543 | 035440 | 012737 | 000037 | 017124 | | MOV #37,E2 | | ; SRC2 = ERT4 | |
| 10544 | 035446 | 012737 | 034242 | 017126 | | MOV #ERT4,E3 | | ; DST = ERT4 | |
| 10545 | 035454 | 012737 | 000037 | 017130 | | MOV #37,E4 | | | |
| 10546 | 035462 | 012737 | 034242 | 017132 | | MOV #ERT4,E5 | | | |
| 10547 | 035470 | 004737 | 023136 | | | JSR PC,EADDP | | | |
| 10548 | | | | | | | | | |
| 10549 | 035474 | 012737 | 000037 | 017120 | | MOV #37,E0 | | ;STORE SUBTRACT COUNTER IN RESULT(ERT2) | |
| 10550 | 035502 | 012737 | 034176 | 017122 | | MOV #ERT2,E1 | | ;SHIFT RESULT, THEN ENTER DIGIT | |
| 10551 | 035510 | 012737 | 000001 | 017130 | | MOV #1,E4 | | ; USE ASHP - SRC = ERT2 | |
| 10552 | 035516 | 012737 | 000037 | 017124 | | MOV #37,E2 | | ; DST=ERT2 | |
| 10553 | 035524 | 012737 | 034176 | 017126 | | MOV #ERT2,E3 | | ; SHFT.CT=1 | |
| 10554 | 035532 | 004737 | 030376 | | | JSR PC,EASHP | | | |
| 10555 | 035536 | 006337 | 034674 | | | ASL ESUBCT | | ;INSERT SUBTRACT COUNTER IN RESULT | |
| 10556 | 035542 | 006337 | 034674 | | | ASL ESUBCT | | | |
| 10557 | 035546 | 006337 | 034674 | | | ASL ESUBCT | | | |
| 10558 | 035552 | 006337 | 034674 | | | ASL ESUBCT | | | |
| 10559 | 035556 | 153737 | 034674 | 034215 | | BISB ESUBCT,@#ERT2+17 | | | |
| 10560 | 035564 | 005337 | 034610 | | | DEC ESPOS | | ;DECREMENT SHIFT POSITION | |
| 10561 | 035570 | 001237 | | | | BNE EDIVL | | ;IS SHIFT POSITION=0? BRANCH IF NO | |
| 10562 | 035572 | 023737 | 034670 | 034672 | | CMP ESS2SN,ESS1SN | | ;DIVP SRC1 SIGN = DIVP SRC2 SIGN? | |
| 10563 | 035600 | 001403 | | | | BEQ EDID | | ;BRANCH IF YES | |
| 10564 | 035602 | 152737 | 000001 | 034215 | | BISB #1,@#ERT2+17 | | ;NO - MAKE SIGN IN ERT2 NEGATIVE | |
| 10565 | | | | | | | | | |
| 10566 | 035610 | 012737 | 000037 | 017120 | EDID: | MOV #37,E0 | | ;MOVE RESULT INTO DST | |
| 10567 | 035616 | 012737 | 034176 | 017122 | | MOV #ERT2,E1 | | ; USE ASHP - SRC = ERT2 | |
| 10568 | 035624 | 005037 | 017130 | | | CLR E4 | | ; DST = DIVP.DST | |
| 10569 | 035630 | 005037 | 017132 | | | CLR E5 | | ; SHFT.CT = 0 | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|-------------------|---------------------------------------|
| 10570 | 035634 | 013737 | 034340 | 017124 | | MOV ESR4,E2 | |
| 10571 | 035642 | 013737 | 034342 | 017126 | | MOV ESR5,E3 | |
| 10572 | 035650 | 013737 | 034324 | 017114 | | MOV ESOSTK,EORSTK | |
| 10573 | 035656 | 013737 | 034326 | 017116 | | MOV ESOPSW,EOPSW | |
| 10574 | 035664 | 004737 | 030376 | | | JSR PC,EASHP | |
| 10575 | 035670 | 000137 | 033644 | | | JMP EXMD | :EXIT DIVP |
| 10576 | | | | | | | |
| 10577 | 035674 | 013737 | 034326 | 017116 | EXZD: | MOV ESOPSW,EOPSW | :EXIT FROM DIVIDE BY ZERO |
| 10578 | 035702 | 052777 | 000003 | 161206 | | BIS #3,@EOPSW | :SET V & C COND. CODES |
| 10579 | | | | | | | :SET ZERO DIVIDE FLAG TO SIGNAL |
| 10580 | 035710 | 012737 | 177777 | 035744 | | MOV #177777,EZDF | :TABLE DRIVER NOT TO COMPARE |
| 10581 | | | | | | | :ANYTHING EXCEPT V & C COND. CODE |
| 10582 | | | | | | | :SAVE POINTER TO START & # OF BYTES |
| 10583 | 035716 | 013737 | 034332 | 035746 | | MOV ESR1,EZDBEG | :OF DST STRING |
| 10584 | | | | | | | :CONTENTS OF DST STRING UNPREDICTABLE |
| 10585 | 035724 | 013737 | 034330 | 035750 | | MOV ESRO,EZDEND | :AFTER ZERO DIVP |
| 10586 | 035732 | 006237 | 035750 | | | ASR EZDEND | |
| 10587 | 035736 | 005237 | 035750 | | | INC EZDEND | |
| 10588 | 035742 | 000207 | | | | RTS PC | :RESULTS |
| 10589 | | | | | | | |
| 10590 | 035744 | 000000 | | | EZDF: | .WORD 0 | |
| 10591 | 035746 | 000000 | | | EZDBEG: | .WORD 0 | |
| 10592 | 035750 | 000000 | | | EZDEND: | .WORD 0 | |
| 10593 | | | | | | | |

| | | | | | | |
|-------|--------|--------|--------|--------|----------------|--------------------------------|
| 10595 | | | | | .SBTTL | LOAD DESCRIPTORS |
| 10596 | 035752 | 013701 | 017120 | | MOV E0,R1 | :GET REGISTER POINTER |
| 10597 | 035756 | 012102 | | EL2D0: | MOV (R1)+,R2 | :GET ADDRESS OF DESCRIPTOR |
| 10598 | 035760 | 011237 | 017120 | | MOV (R2),E0 | :LOAD 1ST WORD OF DESC INTO E0 |
| 10599 | 035764 | 016237 | 000002 | 017122 | MOV 2(R2),E1 | :LOAD 2ND WORD OF DESC INTO E1 |
| 10600 | 035772 | 011102 | | | MOV (R1),R2 | :GET ADDRESS OF NEXT DESC |
| 10601 | 035774 | 011237 | 017124 | | MOV (R2),E2 | :LOAD 1ST WORD OF DESC INTO E2 |
| 10602 | 036000 | 016237 | 000002 | 017126 | MOV 2(R2),E3 | :LOAD 2ND WORD OF DESC INTO E3 |
| 10603 | 036006 | 013700 | 017114 | | MOV EORSTK,R0 | :RETURN CLEAN UP |
| 10604 | 036012 | 013720 | 017120 | | MOV E0,(R0)+ | :R0=R0 |
| 10605 | 036016 | 013720 | 017122 | | MOV E1,(R0)+ | :R1=R1 |
| 10606 | 036022 | 013720 | 017124 | | MOV E2,(R0)+ | :R2=R2 |
| 10607 | 036026 | 013720 | 017126 | | MOV E3,(R0)+ | :R3=R3 |
| 10608 | 036032 | 013720 | 017130 | | MOV E4,(R0)+ | :R4=R4 |
| 10609 | 036036 | 013720 | 017132 | | MOV E5,(R0)+ | :R5=R5 |
| 10610 | 036042 | 013720 | 017134 | | MOV E6,(R0)+ | :R6=R6 |
| 10611 | 036046 | 012777 | 000017 | 161042 | MOV #17,@EOPSW | :SET ALL COND. CODE BITS. |
| 10612 | 036054 | 000207 | | | RTS PC | |
| 10613 | | | | | | |
| 10614 | 036056 | | | | EL2D1: | |
| 10615 | 036056 | 013701 | 017122 | | MOV E1,R1 | |
| 10616 | 036062 | 000735 | | | BR EL2 | |
| 10617 | | | | | | |
| 10618 | 036064 | | | | EL2D2: | |
| 10619 | 036064 | 013701 | 017124 | | MOV E2,R1 | |
| 10620 | 036070 | 000732 | | | BR EL2 | |
| 10621 | | | | | | |
| 10622 | 036072 | | | | EL2D3: | |
| 10623 | 036072 | 013701 | 017126 | | MOV E3,R1 | |
| 10624 | 036076 | 000727 | | | BR EL2 | |
| 10625 | | | | | | |
| 10626 | 036100 | | | | EL2D4: | |
| 10627 | 036100 | 013701 | 017130 | | MOV E4,R1 | |
| 10628 | 036104 | 062737 | 000004 | 017130 | ADD #4,E4 | |
| 10629 | 036112 | 000721 | | | BR EL2 | |
| 10630 | | | | | | |
| 10631 | 036114 | | | | EL2D5: | |
| 10632 | 036114 | 013701 | 017132 | | MOV E5,R1 | |
| 10633 | 036120 | 062737 | 000004 | 017132 | ADD #4,E5 | |
| 10634 | 036126 | 000713 | | | BR EL2 | |
| 10635 | | | | | | |
| 10636 | 036130 | | | | EL2D6: | |
| 10637 | 036130 | 013701 | 017134 | | MOV E6,R1 | |
| 10638 | 036134 | 012102 | | | MOV (R1)+,R2 | |
| 10639 | 036136 | 011237 | 017120 | | MOV (R2),E0 | |
| 10640 | 036142 | 016237 | 000002 | 017122 | MOV 2(R2),E1 | |
| 10641 | 036150 | 012102 | | | MOV (R1)+,R2 | |
| 10642 | 036152 | 011237 | 017124 | | MOV (R2),E2 | |
| 10643 | 036156 | 016237 | 000002 | 017126 | MOV 2(R2),E3 | |
| 10644 | 036164 | 010137 | 017134 | | MOV R1,E6 | |
| 10645 | 036170 | 000137 | 036006 | | JMP EXL2 | |
| 10646 | | | | | | |
| 10647 | 036174 | | | | EL2D7: | |
| 10648 | 036174 | 013701 | 047466 | | MOV TINST+2,R1 | |

:NOTE:L2D6 UPDATES R6 (POPS STACK)

| | | | | |
|-------|--------|--------|--------|--------|
| 10649 | 036200 | 011137 | 017120 | |
| 10650 | 036204 | 016137 | 000002 | 017122 |
| 10651 | 036212 | 013701 | 047470 | |
| 10652 | 036216 | 011137 | 017124 | |
| 10653 | 036222 | 016137 | 000002 | 017126 |
| 10654 | 036230 | 000137 | 036006 | |
| 10655 | | | | |
| 10656 | | | | |

MOV (R1),E0
MOV 2(R1),E1
MOV TINST+4,R1
MOV (R1),E2
MOV 2(R1),E3
JMP EXL2

| Address | Op Code | Op 1 | Op 2 | Op 3 | Op 4 | Instruction | Comment |
|---------|---------|--------|--------|--------|------|-----------------------|------------------------------------|
| 10658 | 036234 | 013701 | 017120 | | | EL3D0: MOV E0,R1 | :GET REGISTER POINTER |
| 10659 | 036240 | 012102 | | | | EL3: MOV (R1)+,R2 | :GET ADDRESS OF DESCRIPTOR |
| 10660 | 036242 | 011237 | 017120 | | | MOV (R2),E0 | :LOAD 1ST WORD OF DESC INTO E0 |
| 10661 | 036246 | 016237 | 000002 | 017122 | | MOV 2(R2),E1 | :LOAD 2ND WORD OF DESC INTO E1 |
| 10662 | 036254 | 012102 | | | | MOV (R1)+,R2 | :GET ADDRESS OF NEXT DESC |
| 10663 | 036256 | 011237 | 017124 | | | MOV (R2),E2 | :LOAD 1ST WORD OF DESC INTO E2 |
| 10664 | 036262 | 016237 | 000002 | 017126 | | MOV 2(R2),E3 | :LOAD 2ND WORD OF DESC INTO E3 |
| 10665 | 036270 | 011102 | | | | MOV (R1),R2 | :GET ADDRESS OF NEXT DESC |
| 10666 | 036272 | 011237 | 017130 | | | MOV (R2),E4 | :LOAD 1ST WORD OF DESC INTO E4 |
| 10667 | 036276 | 016237 | 000002 | 017132 | | MOV 2(R2),E5 | :LOAD 2ND WORD OF DESC INTO E5 |
| 10668 | 036304 | 000137 | 036006 | | | JMP EXL2 | |
| 10669 | | | | | | | |
| 10670 | 036310 | | | | | EL3D1: MOV E1,R1 | |
| 10671 | 036310 | 013701 | 017122 | | | BR EL3 | |
| 10672 | 036314 | 000751 | | | | | |
| 10673 | | | | | | | |
| 10674 | 036316 | | | | | EL3D2: MOV E2,R1 | |
| 10675 | 036316 | 013701 | 017124 | | | BR EL3 | |
| 10676 | 036322 | 000746 | | | | | |
| 10677 | | | | | | | |
| 10678 | 036324 | | | | | EL3D3: MOV E3,R1 | |
| 10679 | 036324 | 013701 | 017126 | | | BR EL3 | |
| 10680 | 036330 | 000743 | | | | | |
| 10681 | | | | | | | |
| 10682 | 036332 | | | | | EL3D4: MOV E4,R1 | |
| 10683 | 036332 | 013701 | 017130 | | | BR EL3 | |
| 10684 | 036336 | 000740 | | | | | |
| 10685 | | | | | | | |
| 10686 | 036340 | | | | | EL3D5: MOV E5,R1 | |
| 10687 | 036340 | 013701 | 017132 | | | BR EL3 | |
| 10688 | 036344 | 000735 | | | | | |
| 10689 | | | | | | | |
| 10690 | 036346 | | | | | EL3D6: MOV E6,R1 | |
| 10691 | 036346 | 013701 | 017134 | | | MOV (R1)+,R2 | |
| 10692 | 036352 | 012102 | | | | MOV (R2),E0 | |
| 10693 | 036354 | 011237 | 017120 | | | MOV 2(R2),E1 | |
| 10694 | 036360 | 016237 | 000002 | 017122 | | MOV (R1)+,R2 | |
| 10695 | 036366 | 012102 | | | | MOV (R2),E2 | |
| 10696 | 036370 | 011237 | 017124 | | | MOV 2(R2),E3 | |
| 10697 | 036374 | 016237 | 000002 | 017126 | | MOV (R1)+,R2 | |
| 10698 | 036402 | 012102 | | | | MOV (R2),E4 | |
| 10699 | 036404 | 011237 | 017130 | | | MOV 2(R2),E5 | |
| 10700 | 036410 | 016237 | 000002 | 017132 | | MOV R1,E6 | |
| 10701 | 036416 | 010137 | 017134 | | | JMP EXL2 | :NOTE:L2D6 UPDATES R6 (POPS STACK) |
| 10702 | 036422 | 000137 | 036006 | | | | |
| 10703 | | | | | | | |
| 10704 | 036426 | | | | | EL3D7: MOV TINST+2,R1 | |
| 10705 | 036426 | 013701 | 047466 | | | MOV (R1),E0 | |
| 10706 | 036432 | 011137 | 017120 | | | MOV 2(R1),E1 | |
| 10707 | 036436 | 016137 | 000002 | 017122 | | MOV TINST+4,R1 | |
| 10708 | 036444 | 013701 | 047470 | | | MOV (R1),E2 | |
| 10709 | 036450 | 011137 | 017124 | | | MOV 2(R1),E3 | |
| 10710 | 036454 | 016137 | 000002 | 017126 | | MOV TINST+6,R1 | |
| 10711 | 036462 | 013701 | 047472 | | | | |

| | | | | |
|-------|--------|--------|--------|--------|
| 10712 | 036466 | 011137 | 017130 | |
| 10713 | 036472 | 016137 | 000002 | 017132 |
| 10714 | 036500 | 000137 | 036006 | |
| 10715 | | | | |

MOV (R1),E4
MOV 2(R1),E5
JMP EXL2


```

10717          .SBTTL  CIS INSTRUCTION TEST LOOP
10718
10719 036504 012737 177777 001744 SEEDST: MOV #177777,ESEED          ;SET ENTER RNG SEED FLAG
10720 036512 000476          BR DVTST
10722 036514 012737 177777 001750 START:  MOV #177777,N200M          ;SET FLAG TO INDICATE THAT PROG WAS
10723          ;STARTED AT LOC 200
10725 036522 012737 177777 002206 QVST:   MOV #177777,QVMODE          ;SET QVMODE FLAG
10731 036530 005037 001746 NST:     CLR DENS          ;CLEAR DON'T ENTER NORMAL RNG SEED FLAG.
10732 036534 005037 001120          CLR $TESTN          ;CLEAR TEST COUNT
10733 036540 012706 001100          MOV #STACK,SP          ;SETUP THE STACK POINTER
(1) 036544 012737 110536 000034          MOV #$TRAP,@#TRAPVEC  ;TRAP VECTOR FOR TRAP CALLS
(1) 036552 012737 000340 000036          MOV #340,@#TRAPVEC+2;LEVEL 7
(1) 036560 012737 110562 000024          MOV #$PWRDN,@#PWRVEC ;POWER FAILURE VECTOR
(1) 036566 012737 000340 000026          MOV #340,@#PWRVEC+2  ;LEVEL 7
(2) 036574 013746 000004          MOV @#4,-(SP)        ;SAVE ERROR VECTOR
(2) 036600 013746 000006          MOV @#6,-(SP)
(2) 036604 012737 036620 000004          MOV #64$,4          ;SET UP TIME OUT VECTOR
(2) 036612 005777 143120          TST @SWR            ;TRY TO REFERENCE HARDWARE SWR
(2) 036616 000407          BR 65$             ;BRANCH IF NO TIMEOUT TRAP OCCURS
(2) 036620 012737 000176 001736 64$:   MOV #SWREG,SWR      ;POINT TO SOFTWARE SWR
(2) 036626 012737 000174 001740          MOV #DISPREG,DISPLAY;POINT TO SOFTWARE DISPLAY REG
(2) 036634 022626          CMP (SP)+,(SP)+    ;RESTORE STACK
(2) 036636 012637 000006          MOV (SP)+,@#6      ;RESTORE ERROR VECTOR
(2) 036642 012637 000004          MOV (SP)+,@#4
(1) 036646 005037 001122          CLR $PASS          ;CLEAR PASS COUNT
(1) 036652 132737 000200 001135          BITB #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT
(1) 036660 001403          BEQ 3$             ;YES,USE NON-APT SWITCH
(1) 036662 012737 001136 001736          MOV #SWREG,SWR     ;NO,USE APT SWITCH REGISTER
(1) 036670          3$:
10734          ; NO QUESTIONS ASKED - EXERCISES FIXED TABLE TEST
10735          ; CONDITIONS FIRST THEN ENTERS RANDOM MODE TESTING
10736 036670 012737 177777 002074          MOV #177777,FSRUN
10737 036676 005037 001660          CLR INCSQ1          ;PRIOR TO EACH TEST, BUFFERS WILL BE INITIALIZED TO ZERO
10738 036702 005037 001662          CLR INCSQ2
10739 036706 000462          BR COMST
10740 036710 005037 002206 DVTST: CLR QVMODE
10741 036714 005037 001746          CLR DENS          ;CLEAR DON'T ENTER NORMAL RNG SEED FLAG
10742 036720 012706 001100          MOV #STACK,SP          ;SETUP THE STACK POINTER
(1) 036724 012737 110536 000034          MOV #$TRAP,@#TRAPVEC  ;TRAP VECTOR FOR TRAP CALLS
(1) 036732 012737 000340 000036          MOV #340,@#TRAPVEC+2;LEVEL 7
(1) 036740 012737 110562 000024          MOV #$PWRDN,@#PWRVEC ;POWER FAILURE VECTOR
(1) 036746 012737 000340 000026          MOV #340,@#PWRVEC+2  ;LEVEL 7
(2) 036754 013746 000004          MOV @#4,-(SP)        ;SAVE ERROR VECTOR
(2) 036760 013746 000006          MOV @#6,-(SP)
(2) 036764 012737 037000 000004          MOV #64$,4          ;SET UP TIME OUT VECTOR
(2) 036772 005777 142740          TST @SWR            ;TRY TO REFERENCE HARDWARE SWR
(2) 036776 000407          BR 65$             ;BRANCH IF NO TIMEOUT TRAP OCCURS
(2) 037000 012737 000176 001736 64$:   MOV #SWREG,SWR      ;POINT TO SOFTWARE SWR
(2) 037006 012737 000174 001740          MOV #DISPREG,DISPLAY;POINT TO SOFTWARE DISPLAY REG
(2) 037014 022626          CMP (SP)+,(SP)+    ;RESTORE STACK
(2) 037016 012637 000006          MOV (SP)+,@#6      ;RESTORE ERROR VECTOR
(2) 037022 012637 000004          MOV (SP)+,@#4
(1) 037026 005037 001122          CLR $PASS          ;CLEAR PASS COUNT
(1) 037032 132737 000200 001135          BITB #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT

```

```

(1) 037040 001403          BEQ 3$          ;;YES,USE NON-APT SWITCH
(1) 037042 012737 001136 001736  MOV  #$$SWREG,SWR ;;NO,USE APT SWITCH REGISTER
(1) 037050          3$:          ; RESULTS IN DIALOG WITH USER TO DETERMINE
10743          ; EXACT RUN MODE DESIRED.
10744
10749 037050 005037 002074          CLR FSRUN
10750 037054          COMST:
10751 037054 005077 142604          CLR @TPSW          ;SET PROCESSOR PRIORITY TO ZERO
10752 037060 012737 037112 000010  MOV #22$,@#RESVEC ;CHECK FOR SWITCH ON CIS MODULE TO BE IN CORRECT POSITIO
10753 037066 005037 000012          CLR @#RESVEC+2
10754 037072 076001          76001          ;THIS INST SHOULD TRAP TO LOC 10 IF
10755          ; SWITCH POSITION IS OK; OTHERWISE
10756          ; IT WILL ACT LIKE A 'NOP'.
10757 037074          PRINTB #SWNG
(6) 037074 012746 013400          MOV #SWNG,-(SP)
(3) 037100 010600          MOV SP,R0
(4) 037102 004737 065304          JSR PC,FPRINT
10758 037106 000000          HALT
10759 037110 000761          BR COMST
10764 037112 005737 001744          22$: TST ESEED          ;GET NEW SEED CONSTANTS?
10765 037116 001437          BEQ 2$          ;BRANCH IF NO
10766 037120          111$: PRINTB #ACCSEED ;PRINT MESSAGE: ENTER RNG SEED CONSTANTS
(6) 037120 012746 016033          MOV #ACCSEED,-(SP)
(3) 037124 010600          MOV SP,R0
(4) 037126 004737 065304          JSR PC,FPRINT
10767 037132 004737 064776          JSR PC,ACCOCT          ;GET OCTAL SEED
10768 037136 000770          BR 111$          ;<CR> RETURN
10769 037140 000240          NOP          ;XXXXXX<CR> RETURN
10770 037142 000240          NOP          ;XXXXXX<-> RETURN
10771 037144 012637 063544          MOV (SP)+,RNCON ;INSERT FIRST SEED CONSTANT
10772 037150 004737 064776          JSR PC,ACCOCT          ;GET SECOND SEED
10773 037154 000761          BR 111$
10774 037156 000240          NOP
10775 037160 000240          NOP
10776 037162 012637 063546          MOV (SP)+,RP1          ;INSERT SECOND SEED CONSTANT
10777 037166 004737 064776          JSR PC,ACCOCT          ;GET THIRD SEED
10778 037172 000752          BR 111$
10779 037174 000240          NOP
10780 037176 000240          NOP
10781 037200 012637 063550          MOV (SP)+,RP2          ;INSERT THIRD SEED CONSTANT
10782 037204 012737 177777 001746          MOV #177777,DENS ;SET DON'T ENTER NORMAL SEED FLAG
10783 037212 005037 001744          CLR ESEED
10784 037216 012737 056566 000010  2$: MOV #ILLSER,@#RESVEC ;SETUP ILLEGAL INST TRAP CATCHER
10786 037224 005037 000012          CLR @#RESVEC+2
10787 037230 012737 056750 000250          MOV #MMVIOL,@#MMVEC ;SETUP MEMORY MANAGEMENT TRAP CATCHER
10788 037236 005037 000252          CLR @#MMVEC+2
10789 037242 012737 056446 000004          MOV #HLTSER,@#ERRVEC ;SETUP TIMEOUT INST. TRAP VECTOR
10790 037250 005037 000006          CLR @#ERRVEC+2
10791 037254 004737 053724          JSR PC,SIZEPT          ;SETUP PROCESSOR DEPENDENT CONSTANTS
10792 037260 023727 000042 053076  1$: CMP @#42,#ENDAD ;IF IN ACT CHAIN MODE SKIP PRINTING OF PROG TITLE
10793 037266 001405          BEQ 14$
10794 037270 005737 001122          TST $PASS          ;IDENTIFY PROGRAM ON 1ST PASS ONLY
10795 037274 001002          BNE 14$
10796 037276 104400          TYPE

```

| | | | | | | | |
|-------|--------|--------|--------|--------|--------|-------------------|--|
| 10797 | 037300 | 016551 | | | | PNAME | :TYPE PROGRAM NAME |
| 10798 | 037302 | 005037 | 002464 | | 14\$: | CLR ONEINS | :CLEAR SINGLE INST TEST FLAG |
| 10799 | 037306 | 012737 | 000207 | 061174 | | MOV #207,DI | :INHIBIT INTERRUPT DURING INTR SERVICE DIVPI |
| 10800 | 037314 | 012737 | 000414 | 047372 | | MOV #414,TOLTC | :INHIBIT LTC TURN ON |
| 10801 | 037322 | 012737 | 000403 | 047424 | | MOV #403,TOPC2 | :INHIBIT LATENCY & INTERRUPTABILITY TURN ON |
| 10802 | 037330 | 012737 | 000403 | 047434 | | MOV #403,TOPC1 | |
| 10803 | 037336 | 005037 | 002054 | | | CLR ERRCT | :CLEAR ERROR COUNT |
| 10804 | 037342 | 005037 | 002162 | | | CLR DEN | :CLEAR D-SPACE ENABLE FLAG |
| 10805 | 037346 | 005037 | 003024 | | | CLR LCNT | :CLEAR LTC COUNT |
| 10806 | 037352 | 005037 | 002540 | | | CLR LATEN | :CLEAR LATENCY TESTING FLAG |
| 10807 | 037356 | 005037 | 002132 | | | CLR FATAL | :CLEAR FATAL ERROR INDICATOR |
| 10808 | 037362 | 005037 | 001760 | | | CLR RANDOM | :CLEAR RANDOM EXERCISE MODE FLAG |
| 10809 | 037366 | 005037 | 002042 | | | CLR NOERDS | :CLEAR 'NO-ERROR DISPLAY' SWITCH |
| 10810 | 037372 | 004737 | 055032 | | | JSR PC,SETPAR | :SETUP PAR'S (MEM MGMT) |
| 10811 | 037376 | 013700 | 001654 | | | MOV PSEED,R0 | :FORM RNG PRINT SEED MASK |
| 10812 | 037402 | 005100 | | | | COM R0 | |
| 10813 | 037404 | 005200 | | | | INC R0 | |
| 10814 | 037406 | 010037 | 002076 | | | MOV R0,MSEED | |
| 10816 | 037412 | 005737 | 001746 | | | TST DENS | :ENTER NORMAL SEED? |
| 10817 | 037416 | 001011 | | | | BNE 61\$ | :BRANCH IF NO |
| 10819 | 037420 | 013737 | 063552 | 063544 | | MOV KRNCON,RNCON | :INITIALIZE RANDOM # GENERATOR |
| 10820 | 037426 | 013737 | 063554 | 063546 | | MOV KRP1,RP1 | |
| 10821 | 037434 | 013737 | 063556 | 063550 | | MOV KRP2,RP2 | |
| 10822 | 037 42 | 012737 | 072100 | 072076 | 61\$: | MOV #IL2D,INPTBL | :INITIALIZE INPUT TABLE POINTER |
| 10823 | 037450 | 005737 | 002074 | | | TST FSRUN | :FIELD SERVICE TYPE RUN |
| 10824 | 037454 | 001434 | | | | BEQ 13\$ | :BRANCH IF NO TO ENTER DIALOG WITH USER |
| 10825 | | | | | | | :DETERMINE IF LINE CLOCK IS AVAILABLE FOR |
| 10826 | | | | | | | : FIELD SERVICE TYPE RUN |
| 10827 | 037456 | 004737 | 062312 | | 31\$: | JSR PC,LTCP | :IS LTC ON SYSTEM? |
| 10828 | 037462 | 000137 | 037530 | | | JMP 32\$ | :NO - CANT TEST INTERRUPTABILITY |
| 10829 | 037466 | 005737 | 001122 | | | TST \$PASS | :IDENTIFY INTR SOURCE ON 1ST PASS |
| 10830 | 037472 | 001005 | | | | BNE 103\$ | |
| 10831 | 037474 | | | | | PRINTB #KW11L | :INDICATE THAT LINE CLOCK WILL BE USED |
| (6) | 037474 | 012746 | 013544 | | | MOV #KW11L,-(SP) | |
| (3) | 037500 | 010600 | | | | MOV SP,R0 | |
| (4) | 037502 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 10832 | | | | | | | :FOR INTERRUPT SOURCE. |
| 10833 | 037506 | 004737 | 062234 | | 103\$: | JSR PC,LTCSUP | :SYNC UP TO LTC |
| 10834 | 037512 | 004737 | 062256 | | | JSR PC,LTCNT | :DETERMINE COUNT PER CLOCK TICK |
| 10835 | 037516 | 012777 | 062044 | 143272 | | MOV #LTCIS,@LTCIV | :SETUP LTC INTR VECTOR |
| 10836 | 037524 | 000137 | 040306 | | | JMP FDIALG | :SKIP OVER DIALOG WITH USER |
| 10837 | 037530 | | | | 32\$: | PRINTB #NOINT | :PRINT CANT TEST INTR MESSAGE |
| (6) | 037530 | 012746 | 013325 | | | MOV #NOINT,-(SP) | |
| (3) | 037534 | 010600 | | | | MOV SP,R0 | |
| (4) | 037536 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 10838 | 037542 | 000137 | 040306 | | | JMP FDIALG | |
| 10839 | 037546 | | | | 13\$: | PRINTB #ASKINT | :ASK IF INTERRUPTABILITY MODE IS DESIRED? |
| (6) | 037546 | 012746 | 012715 | | | MOV #ASKINT,-(SP) | |
| (3) | 037552 | 010600 | | | | MOV SP,R0 | |
| (4) | 037554 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 10840 | 037560 | 004737 | 064460 | | | JSR PC,YORN | :ACCEPT ASCIZ FROM TTY |
| 10841 | 037564 | 000137 | 040076 | | | JMP ARMQ | :N RESPONSE |
| 10842 | 037570 | 000137 | 037604 | | | JMP 5\$ | :Y RESPONSE |
| 10843 | 037574 | 000137 | 040076 | | | JMP ARMQ | :R OR H RESPONSE (ILLEGAL HERE) |

```

10844 037600 000137 040076          JMP ARMQ          ;C RESPONSE (ILLEGAL HERE)
10845
10846 037604          5$: PRINTB #ASKSRC      ;ASK FOR INTERRUPT SOURCE
    (6) 037604 012746 013030      MOV #ASKSRC,-(SP)
    (3) 037610 010600          MOV SP,R0
    (4) 037612 004737 065304      JSR PC,FPRINT
10847 037616 004737 064460      JSR PC,YORN      ;ACCEPT ASCIZ
10848 037622 000137 037642      JMP 52$          ;(N) KW11-P @100KHZ
10849 037626 000137 037666      JMP 53$          ;(Y) KW11-P EXT OSC
10850 037632 000137 037726      JMP 54$          ;(R) LINE TIME CLOCK
10851 037636 000137 037760      JMP 55$          ;(C) KW11-P @10KHZ
10852
10853 037642          52$: ;MAKE KW11-P @100KHZ THE INTERRUPT SOURCE
10854 037642 004737 061602      JSR PC,PC1CK     ;CHECK FOR (& SETUP) P-CLK 1
10855 037646 000137 040002      JMP MNOPC1       ;NOT PRESENT ON SYSTEM RETURN
10856 037652 004737 061716      JSR PC,PC2CK     ;P-CLK EXISTS RETURN - CHECK FOR (& SETUP) 2ND PCLK
10857                                     ; FOR LATENCY TESTING
10858 037656 000137 040020      JMP MNOPC2       ;NOT PRESENT ON SYSTEM RETURN
10859 037662 000137 040032      JMP ADIQ         ;2ND PCLK EXISTS
10860
10861 037666          53$: ;MAKE KW11-P WITH EXTERNAL OSCILLATOR THE
10862                                     ; INTERRUPT SOURCE
10863 037666 004737 061602      JSR PC,PC1CK     ;CHECK FOR (& SETUP) P-CLK ON SYSTEM
10864 037672 000137 040002      JMP MNOPC1       ;NOT PRESENT ON SYSTEM RETURN
10865 037676 052777 000006 142610  BIS #6,@PC1CSR   ;SET PCLK 1 FOR EXTERNAL OSCILLATOR
10866 037704 004737 061716      JSR PC,PC2CK     ;CHECK FOR 2ND P-CLK FOR LATENCY TESTING
10867 037710 000137 040020      JMP MNOPC2       ;NOT PRESENT RETURN
10868 037714 052777 000006 142602  BIS #6,@PC2CSR   ;SET PCLK2 FOR EXTERNAL OSC
10869 037722 000137 040032      JMP ADIQ
10870
10871 037726          54$: ;MAKE LINE TIME CLOCK THE INTR SOURCE
10872 037726 004737 062312      JSR PC,LTCP      ;CHECK FOR LTC ON SYSTEM
10873 037732 000137 040002      JMP MNOLTC       ;NOT PRESENT RETURN
10874 037736 004737 062234      JSR PC,LTCSUP    ;SYNC UP TO LTC
10875 037742 004737 062256      JSR PC,LTCNT     ;DETERMINE COUNT PER CLOCK TICK
10876 037746 012777 062044 143042  MOV #LTCIS,@LTCIV ;SETUP LTC INTR VECTOR
10877 037754 000137 040076      JMP ARMQ
10878
10879 037760          55$: ;MAKE KW11-P @10KHZ THE INTERRUPT SOURCE
10880 037760 004737 061602      JSR PC,PC1CK     ;CHECK FOR P-CLK ON SYSTEM
10881 037764 000137 040002      JMP MNOPC1       ;NOT PRESENT RETURN
10882 037770 052777 000002 142516  BIS #2,@PC1CSR   ;SET PCLK FOR 10KHZ (NO LATENCY TESTING)
10883 037776 000137 040076      JMP ARMQ
10884
10885 040002          MNOLTC:
10886 040002          MNOPC1: PRINTB #NOINT      ;PRINT CANT TEST INTERRUPTABILITY MESSAGE
    (6) 040002 012746 013325      MOV #NOINT,-(SP)
    (3) 040006 010600          MOV SP,R0
    (4) 040010 004737 065304      JSR PC,FPRINT
10887 040014 000137 040076      JMP ARMQ
10888 040020          MNOPC2: PRINTB #NOLAT     ;PRINT CANT TEST LATENCY MESSAGE
    (6) 040020 012746 013254      MOV #NOLAT,-(SP)
    (3) 040024 010600          MOV SP,R0
    (4) 040026 004737 065304      JSR PC,FPRINT

```

```

10889 040032          ADIQ: PRINTB #ASKDI          ;ASK IF USER ALLOWS AN INTERRUPT DURING CIS
      (6) 040032 012746 013143          MOV #ASKDI,-(SP)
      (3) 040036 010600          MOV SP,R0
      (4) 040040 004737 065304          JSR PC,FPRINT
10890                                     ; INST EXECUTED ON NORMAL INTR SERVICE ROUTINE
10891 040044 004737 064460          JSR PC,YORN          ;ACCEPT ASCIZ FROM TTY
10892 040050 000137 040076          JMP ARMQ            ;N RESPONSE
10893 040054 000137 040070          JMP 1$             ;Y RESPONSE
10894 040060 000137 040076          JMP ARMQ            ;ILLEGAL RESPONSE
10895 040064 000137 040076          JMP ARMQ            ;ILLEGAL RESPONSE
10896 040070 013737 001672 061174 1$: MOV KNOP,DI        ;OVERWRITE 'RTS PC' TO ALLOW P-CLK
10897                                     ; INTERRUPT DURING CIS INST EXECUTED
10898                                     ; WITHIN NORMAL P-CLK INTERRUPT SERVICE ROUTINE.
10899
10900
10901
10902 040076          ARMQ: PRINTB #ASKRM          ;ASK IF RANDOM EXERCISE MODE IS DESIRED?
      (6) 040076 012746 012455          MOV #ASKRM,-(SP)
      (3) 040102 010600          MOV SP,R0
      (4) 040104 004737 065304          JSR PC,FPRINT
10903 040110 004737 064460          JSR PC,YORN          ;ACCEPT ASCIZ FROM TTY
10904 040114 000137 040152          JMP 2$             ;N RESPONSE
10905 040120 000137 040134          JMP 3$             ;Y RESPONSE
10906 040124 000137 040246          JMP 1$             ;R OR H RESPONSE (ILLEGAL HERE)
10907 040130 000137 040246          JMP 1$             ;C RESPONSE (ILLEGAL HERE)
10908 040134 012737 177777 001760 3$: MOV #177777,RANDOM  ;SET RANDOM FLAG
10909 040142 012737 072022 072076  MOV #IDUM,INPTBL    ;SET INPUT TABLE POINTER TO DUMMY INPUT TABLE
10910 040150 000436
10911 040152          2$: PRINTB #ASKMOD          ;ASK FOR PROCESSOR TEST MODE
      (6) 040152 012746 012523          MOV #ASKMOD,-(SP)
      (3) 040156 010600          MOV SP,R0
      (4) 040160 004737 065304          JSR PC,FPRINT
10912 040164 004737 057046          JSR PC,KSORU        ;ACCEPT ASCIZ FROM TTY AND SETUP MODE WORD
10913 040170 000770          BR 2$             ;ILLEGAL CHAR RETURN - ASK AGAIN
10914 040172          20$: PRINTB #ASKMM          ;ASK FOR MEM MGMT TEST MODE
      (6) 040172 012746 012607          MOV #ASKMM,-(SP)
      (3) 040176 010600          MOV SP,R0
      (4) 040200 004737 065304          JSR PC,FPRINT
10915 040204 004737 064460          JSR PC,YORN
10916 040210 000137 040242          JMP 23$            ;N RESPONSE - MEM MGMT OFF
10917 040214 000137 040232          JMP 22$            ;D RESPONSE - D SPACE ENABLED
10918 040220 000137 040246          JMP 1$             ;H RESPONSE - D SPACE DISABLED
10919 040224 000137 040172          JMP 20$            ;ILLEGAL RESPONSE - ASK AGAIN
10920 040230 000406          BR 1$
10921 040232 012737 177777 002162 22$: MOV #177777,DEN      ;SET D ENABLED FLAG
10922 040240 000402          BR 1$
10923 040242 005037 002156          23$: CLR MMFLG      ;SET NO MEM MGMT FLAG
10924 040246          1$: PRINTB #ASK          ;ASK FOR SPECIFIC INST TO TEST
      (6) 040246 012746 012406          MOV #ASK,-(SP)
      (3) 040252 010600          MOV SP,R0
      (4) 040254 004737 065304          JSR PC,FPRINT
10925 040260 004737 067560          JSR PC,ACASZ        ;ACCEPT ASCIZ FROM TTY
10926 040264 005737 002456          TST ACINST          ;DEFAULTED TO ALL INSTRUCTIONS?
10927 040270 001406          BEQ FDIALG         ;BRANCH IF YES

```

| | | | | | | | |
|-------|--------|--------|--------|--------|--------------------------|--|--|
| 10928 | 040272 | 004737 | 067456 | | JSR PC,SFCI | | :LOOK FOR MATCH BETWEEN INST ENTERED |
| 10929 | | | | | | | :AND LIST OF CIS INST ASCII. |
| 10930 | 040276 | 000763 | | | BR 1\$ | | :NO MATCH RETURN |
| 10931 | 040300 | 012737 | 177777 | 002464 | MOV #177777,ONEINS | | :MATCH - SET SINGLE INST TESTING FLAG |
| 10932 | 040306 | 013737 | 072076 | 002134 | FDIALG: MOV INPTBL,INPTP | | :INITIALIZATION |
| 10933 | 040314 | 005037 | 001420 | | CLR TOTTC | | :ZERO COUNT OF TOTAL TESTS EXECUTED |
| 10934 | 040320 | 005037 | 001416 | | CLR TOTTC | | |
| 10935 | 040324 | 005037 | 001422 | | CLR INVTC | | :ZERO COUNT OF INVALID TESTS - TESTS ABORTED |
| 10936 | 040330 | 005037 | 001424 | | CLR REDTC | | :ZERO COUNT OF REDUNDANT TESTS - TESTS ABORTED. |
| 10937 | 040334 | 005002 | | | CLR R2 | | |
| 10938 | 040336 | 013737 | 001672 | 047444 | MOV KNOP,PREINS | | :INSERT NOP BEFORE INST UNDER TEST. |
| 10939 | 040344 | 012700 | 120606 | | MOV #XLTBL1,R0 | | :INITIALIZE MOVTC TRANSLATION TABLES |
| 10940 | 040350 | 012701 | 121206 | | MOV #ELTBL,R1 | | : AS FOLLOWS; 1 IN LOC 0, 2 IN LOC 1, ETC. |
| 10941 | 040354 | 005202 | | | 11\$: INC R2 | | |
| 10942 | 040356 | 110220 | | | MOVB R2,(R0)+ | | |
| 10943 | 040360 | 020001 | | | CMP R0,R1 | | |
| 10944 | 040362 | 103774 | | | BLO 11\$ | | |
| 10945 | 040364 | 005737 | 002074 | | TST FSRUN | | :NORMAL FIELD SERVICE TYPE RUN? |
| 10946 | 040370 | 001426 | | | BEQ NITE | | :BRANCH IF NO |
| 10948 | 040372 | 023727 | 000042 | 053076 | CMP @#42,#ENDAD | | :IF IN ACT CHAIN MODE SKIF OVER PRINTING OF HEADER |
| 10949 | 040400 | 001412 | | | BEQ 15\$ | | |
| 10950 | 040402 | 005737 | 002206 | | TST QVMODE | | :IF IN QVMODE PRINT QV MODE |
| 10951 | 040406 | 001403 | | | BEQ 1\$ | | |
| 10952 | 040410 | 104400 | | | TYPE | | |
| 10953 | 040412 | 015756 | | | QVHDR | | |
| 10954 | 040414 | 000404 | | | BR 15\$ | | |
| 10955 | 040416 | 104400 | | | 1\$: TYPE | | |
| 10956 | 040420 | 015644 | | | FSHDR | | :PRINT FIELD SERVICE HEADER INFO |
| 10984 | 040422 | 104400 | | | 10\$: TYPE | | :PRINT PASS TIME MESSAGE |
| 10985 | 040424 | 016116 | | | FSHDR1 | | |
| 10987 | 040426 | 012737 | 071454 | 002142 | 15\$: MOV #YL2D,EMPTR | | |
| 10989 | 040434 | 042737 | 000007 | 047464 | BIC #7,TINST | | :CLEAR REGISTER FIELD (FOR L2D DISPLAY ONLY) |
| 10990 | 040442 | 004737 | 063370 | | JSR PC,1DINST | | :IDENTIFY INST UNDER TEST |
| 10992 | | | | | | | |
| 10993 | 040446 | 005037 | 041572 | | NITE: CLR MTYPE | | :CONTROL IS PASSED TO THIS POINT WHENEVER |
| 10994 | | | | | | | : ALL TEST CONDITIONS FOR A GIVEN INPUT |
| 10995 | | | | | | | : TABLE HAVE BEEN EXHAUSTED. PRIOR |
| 10996 | | | | | | | : TO ENTRY TO THIS POINT, THE INPUT |
| 10997 | | | | | | | : TABLE POINTER (INPTP) HAS BEEN UPDATED |
| 10998 | | | | | | | : TO POINT TO THE NEXT INPUT TABLE |
| 10999 | | | | | | | : OF TEST CONDITIONS. IN RANDOM MODE |
| 11000 | | | | | | | : CONTROL IS PASSED HERE FOLLOWING EACH TEST. |
| 11001 | 040452 | 005037 | 002466 | | CLR PZCODE | | |
| 11002 | 040456 | 005037 | 002450 | | CLR PKPTW | | |
| 11003 | 040462 | 005037 | 002434 | | CLR DECINS | | |
| 11004 | 040466 | 005037 | 002452 | | CLR NDESC | | |
| 11005 | 040472 | 005737 | 001760 | | TST RANDOM | | :RUNNING IN RANDOM EXERCISE MODE? |
| 11006 | 040476 | 001436 | | | BEQ 2\$ | | :BRANCH IF NO |
| 11007 | 040500 | 004737 | 062404 | | JSR PC,SRNGST | | :SAVE RANDOM # GENERATOR STATE AT START OF EACH TEST |
| 11008 | 040504 | 013700 | 001420 | | MOV TOTTC,R0 | | :PRINT RANDOM # GEN SEED? |
| 11009 | 040510 | 043700 | 002076 | | BIC MSEED,R0 | | |
| 11010 | 040514 | 001016 | | | BNE 3\$ | | :BRANCH IF NO |
| 11012 | 040516 | 005737 | 002044 | | TST PROGD | | :PROGRESS DISPLAY? |
| 11013 | 040522 | 001013 | | | BNE 3\$ | | :BRANCH IF NO |

| | | | | | | |
|-------|--------|--------|--------|--------|------------------------------|--|
| 11014 | 040524 | | | | PRINTB #FORM36,RNCON,RP1,RP2 | :PRINT 3 SEED CONSTANTS |
| (9) | 040524 | 013746 | 063550 | | MOV RP2,-(SP) | |
| (8) | 040530 | 013746 | 063546 | | MOV RP1,-(SP) | |
| (7) | 040534 | 013746 | 063544 | | MOV RNCON,-(SP) | |
| (6) | 040540 | 012746 | 014240 | | MOV #FORM36,-(SP) | |
| (3) | 040544 | 010600 | | | MOV SP,R0 | |
| (4) | 040546 | 004737 | 065304 | | JSR PC,FPRINT | |
| 11015 | | | | | | :NOTE: TO USE SEED, LOAD 3 CONSTANTS INTO |
| 11016 | | | | | | : KRNCON,KRP1,KRP2 THEN RESTART AT LOC 204. |
| 11021 | 040552 | 005737 | 002464 | 3\$: | TST ONEINS | :SINGLE INST TESTING? |
| 11022 | 040556 | 001002 | | | BNE 1\$ | :BRANCH IF YES |
| 11023 | 040560 | 004737 | 062670 | | JSR PC,GENRI | :GENERATE NEXT RANDOM CIS INST TO TEST |
| 11024 | 040564 | 004737 | 062730 | 1\$: | JSR PC,LDINPT | :LOAD UP DUMMY INPUT TABLE USING RANDOM # GEN. |
| 11025 | 040570 | 004737 | 063142 | | JSR PC,LDCON | :LOAD MISCELLANEOUS CONSTANTS USING RANDOM # GEN. |
| 11026 | 040574 | 013700 | 002134 | 2\$: | MOV INPTP,R0 | :R0 POINTS TO ENTRY IN INPUT TABLE |
| 11027 | 040600 | 016001 | 000002 | | MOV 2(R0),R1 | |
| 11028 | 040604 | 042701 | 177776 | | BIC #177776,R1 | :CLEAR OUT ALL BUT TYPE BIT IN ENTRY TYPE WORD |
| 11029 | 040610 | 006301 | | | ASL R1 | |
| 11030 | 040612 | 062701 | 001426 | | ADD #ITYPE,R1 | |
| 11031 | 040616 | 000171 | 000000 | | JMP @R1) | :DISPATCH ON INPUT TABLE ENTRY TYPE |
| 11032 | | | | | | |
| 11033 | 040622 | | | TYPE0: | MOV #177777,MTYPE | :INPUT PARAMETERS FULLY SPECIFIED IN INPUT TABLE ENTRY |
| 11034 | 040622 | 012737 | 177777 | 041572 | MOV R0,R1 | :SET TYPE FLAG TO INDICATE TYPE 0 ENTRY |
| 11035 | 040630 | 010001 | | | ADD #4,R1 | :SETUP PARAMETER TABLE POINTERS |
| 11036 | 040632 | 062701 | 000004 | | MOV #PTP01,R2 | :R1 POINTS TO IP1 |
| 11037 | 040636 | 012702 | 001570 | | MOV R1,(R2)+ | :R2 POINTS TO TOP OF PARAMETER TABLE POINTER LIST |
| 11038 | 040642 | 010122 | | 1\$: | ADD #2,R1 | :LOAD PTP |
| 11039 | 040644 | 062701 | 000002 | | CMP #PTP24,R2 | :ALL PTP'S LOADED |
| 11040 | 040650 | 022702 | 001636 | | BHIS 1\$ | :BRANCH IF NO |
| 11041 | 040654 | 103372 | | | | |
| 11042 | | | | | | |
| 11043 | 040656 | 005037 | 041574 | 2\$: | CLR ID1 | |
| 11044 | 040662 | 005037 | 041576 | | CLR ID2 | |
| 11045 | 040666 | 011001 | | | MOV (R0),R1 | :UPDATE PTP CONTENTS FOR INDIRECTLY |
| 11046 | 040670 | 006301 | | | ASL R1 | : SPECIFIED DATA DESCRIPTORS. |
| 11047 | 040672 | 062701 | 004360 | | ADD #INSTID,R1 | |
| 11048 | 040676 | 111137 | 041574 | | MOVB (R1),ID1 | |
| 11049 | 040702 | 005201 | | | INC R1 | |
| 11050 | 040704 | 111137 | 041576 | | MOVB (R1),ID2 | :ID1 AND ID2 CONTAIN OFFSETS INTO PTP TABLE |
| 11051 | 040710 | 005737 | 041574 | | TST ID1 | |
| 11052 | 040714 | 001445 | | | BEQ LDCOD | :BRANCH IF OFFSET = 0. NO UPDATE REQUIRED |
| 11053 | 040716 | 006337 | 041574 | | ASL ID1 | |
| 11054 | 040722 | 062737 | 001566 | 041574 | ADD #PTP,ID1 | :USE OFFSET TO GET PTP ENTRY |
| 11055 | 040730 | 017701 | 000640 | | MOV @ID1,R1 | :USE ENTRY TO GET ADDRESS OF DATA DESCRIPTOR |
| 11056 | 040734 | 011177 | 000634 | | MOV (R1),@ID1 | |
| 11057 | 040740 | 005737 | 041576 | | TST ID2 | |
| 11058 | 040744 | 001431 | | | BEQ LDCOD | :BRANCH IF SECOND OFFSET = 0 |
| 11059 | 040746 | 006337 | 041576 | | ASL ID2 | |
| 11060 | 040752 | 062737 | 001566 | 041576 | ADD #PTP,ID2 | :HANDLE SECOND OFFSET SAME AS FIRST |
| 11061 | 040760 | 017701 | 000612 | | MOV @ID2,R1 | |
| 11062 | 040764 | 011177 | 000606 | | MOV (R1),@ID2 | |
| 11063 | 040770 | 000417 | | | BR LDCOD | |
| 11064 | | | | | | |
| 11065 | 040772 | | | TYPE1: | | :INPUT PARAMETERS SPECIFIED IN TABLES |

| | | | | | | |
|-------|--------|--------|--------|--------|------------------|--|
| 11066 | 040772 | 010001 | | | MOV R0,R1 | :SETUP PARAMETER TABLE POINTERS |
| 11067 | 040774 | 062701 | 000004 | | ADD #4,R1 | :R1 POINTS TO IP1 |
| 11068 | 041000 | 012702 | 001570 | | MOV #PTP01,R2 | :R2 POINTS TO TOP OF PARAMETER TABLE POINTER (PTP) LIST. |
| 11069 | 041004 | 012112 | | 1\$: | MOV (R1)+,(R2) | :LOAD PTP FROM IP |
| 11070 | 041006 | 005712 | | | TST (R2) | :PTP=0 ? |
| 11071 | 041010 | 001402 | | | BEQ 2\$ | :YES - DON'T ADVANCE IT |
| 11072 | 041012 | 062712 | 000002 | | ADD #2,(R2) | :ADVANCE PTP TO FIRST ENTRY |
| 11073 | 041016 | 062702 | 000002 | 2\$: | ADD #2,R2 | :UPDATE POINTER |
| 11074 | 041022 | 020227 | 001640 | | CMP R2,#PTP24+2 | :ALL PTP'S LOADED? |
| 11075 | 041026 | 002766 | | | BLT 1\$ | :NO |
| 11076 | | | | | | |
| 11077 | 041030 | 011001 | | LDCOD: | MOV (R0),R1 | :LOAD OCTAL CODING FOR CIS INST |
| 11078 | 041032 | 010137 | 002272 | | MOV R1,OCTIC | : UNDER TEST INTO EINST & TINST |
| 11079 | 041036 | 010137 | 002132 | | MOV R1,FATAL | :LOAD CODING FOR INST UNDER TEST INTO FATAL |
| 11080 | | | | | | : ERROR INDICATOR WORD |
| 11081 | 041042 | 006301 | | | ASL R1 | |
| 11082 | 041044 | 062701 | 003722 | | ADD #OINST,R1 | |
| 11083 | 041050 | 011137 | 045676 | | MOV (R1),EINST | |
| 11084 | 041054 | 011137 | 047464 | | MOV (R1),TINST | |
| 11085 | 041060 | 004737 | 062550 | | JSR PC,SRNGSW | :SAVE STATE OF RANDOM # GEN. AS STATE W |
| 11086 | | | | | | |
| 11087 | 041064 | 004737 | 062574 | NTC: | JSR PC,RRNGSW | :CONTROL IS PASSED TO THIS POINT TO EXECUTE |
| 11088 | | | | | | : NEXT TEST CONDITION FOR GIVEN INPUT |
| 11089 | | | | | | : TABLE. PARAMETER TABLE POINTERS |
| 11090 | | | | | | : HAVE BEEN UPDATED TO POINT TO NEXT |
| 11091 | | | | | | : TEST CONDITION PRIOR TO ENTRY |
| 11092 | | | | | | : TO THIS POINT. |
| 11093 | | | | | | :RESTORE RANDOM # GEN. TO STATE W. |
| 11094 | 041070 | 013701 | 002272 | | MOV OCTIC,R1 | :LOAD # OF INST DESC INTO NDESC |
| 11095 | 041074 | 006301 | | | ASL R1 | :LOAD DATA TYPE CONTROL WORDS |
| 11096 | 041076 | 006301 | | | ASL R1 | : PKPTW,ZPM,SXTYPE |
| 11097 | 041100 | 063701 | 004256 | | ADD DECTYP,R1 | |
| 11098 | 041104 | 012137 | 002274 | | MOV (R1)+,TW1 | |
| 11099 | 041110 | 113737 | 002274 | 002450 | MOVB TW1,PKPTW | :PKPTW IDENTIFIES STARTING DATA TYPE FOR EACH INST TESTI |
| 11100 | 041116 | 113737 | 002275 | 002452 | MOVB TW1+1,NDESC | |
| 11101 | 041124 | 011137 | 002276 | | MOV (R1),TW2 | |
| 11102 | 041130 | 113701 | 002276 | | MOVB TW2,R1 | |
| 11103 | 041134 | 042701 | 177770 | | BIC #177770,R1 | |
| 11104 | 041140 | 010137 | 002436 | | MOV R1,S1TYPE | :S1TYPE IDENTIFIES 1ST STRING DESC DATA TYPE |
| 11105 | 041144 | 113701 | 002276 | | MOVB TW2,R1 | |
| 11106 | 041150 | 042701 | 177707 | | BIC #177707,R1 | |
| 11107 | 041154 | 006201 | | | ASR R1 | |
| 11108 | 041156 | 006201 | | | ASR R1 | |
| 11109 | 041160 | 006201 | | | ASR R1 | |
| 11110 | 041162 | 010137 | 002440 | | MOV R1,S2TYPE | |
| 11111 | 041166 | 113701 | 002276 | | MOVB TW2,R1 | |
| 11112 | 041172 | 006301 | | | ASL R1 | |
| 11113 | 041174 | 006301 | | | ASL R1 | |
| 11114 | 041176 | 000301 | | | SWAB R1 | |
| 11115 | 041200 | 010137 | 002442 | | MOV R1,S3TYPE | |
| 11116 | 041204 | 113701 | 002277 | | MOVB TW2+1,R1 | |
| 11117 | 041210 | 006201 | | | ASR R1 | |
| 11118 | 041212 | 010137 | 002454 | | MOV R1,ZPM | |
| 11119 | 041216 | 005737 | 001760 | | TST RANDOM | :IN RANDOM EXERCISE MODE? |

| | | | | | | |
|-------|--------|--------|--------|---------------|--------------------|--|
| 11174 | 041466 | 020027 | 120602 | 1\$: | CMP R0,#PSTCSK | |
| 11175 | 041472 | 001403 | | | BEQ 2\$ | |
| 11176 | 041474 | 012720 | 055555 | | MOV #055555,(R0)+ | |
| 11177 | 041500 | 000772 | | | BR 1\$ | |
| 11178 | 041502 | 013700 | 002134 | 2\$: | MOV INPTP,R0 | :R0 POINTS TO ENTRY IN INPUT TABLE |
| 11179 | 041506 | 011001 | | | MOV (R0),R1 | :SETUP POINTER TO PROPER CIS |
| 11180 | 041510 | 006301 | | | ASL R1 | : INSTRUCTION FLOW TABLE |
| 11181 | 041512 | 062701 | 001436 | | ADD #INO,R1 | |
| 11182 | 041516 | 011137 | 001742 | | MOV (R1),FLOPTR | |
| 11183 | 041522 | 011001 | | | MOV(R0),R1 | :SETUP POINTER TO PROPER ERROR |
| 11184 | 041524 | 006301 | | | ASL R1 | : MESSAGE HEADER |
| 11185 | 041526 | 062701 | 067730 | | ADD #INEM,R1 | |
| 11186 | 041532 | 011137 | 002142 | | MOV (R1),EMPTR | |
| 11187 | | | | | | |
| 11188 | 041536 | | | XINST: | | |
| 11189 | 041536 | 017701 | 140200 | | MOV @FLOPTR,R1 | :GET NEXT ENTRY FROM INST. FLOW TABLE |
| 11190 | 041542 | 006101 | | | ROL R1 | |
| 11191 | 041544 | 006101 | | | ROL R1 | |
| 11192 | 041546 | 006101 | | | ROL R1 | |
| 11193 | 041550 | 006101 | | | ROL R1 | |
| 11194 | 041552 | 006101 | | | ROL R1 | |
| 11195 | 041554 | 042701 | 177760 | | BIC #177760,R1 | :LOOK ONLY AT FLOW TABLE ENTRY COMMAND |
| 11196 | 041560 | 006301 | | | ASL R1 | : FORM INDEX INTO FLOW DISPATCH TABLE |
| 11197 | 041562 | 062701 | 001526 | | ADD #FLODIS,R1 | |
| 11198 | 041566 | 000171 | 000000 | | JMP @(R1) | :DISPATCH ON FLOW COMMAND |
| 11199 | 041572 | 000000 | | MTYPE: | .WORD 0 | |
| 11200 | 041574 | 000000 | | ID1: | .WORD 0 | |
| 11201 | 041576 | 000000 | | ID2: | .WORD 0 | |
| 11202 | | | | | | |
| 11203 | 041600 | 005737 | 002074 | CTLG: | TST FSRUN | :DVT TYPE RUN? |
| 11204 | 041604 | 001006 | | | BNE NOCTC | :BRANCH IF NO |
| 11205 | 041606 | 000137 | 036710 | | JMP DVTST | :RESTART DVT TYPE RUN |
| 11206 | | | | | | |
| 11207 | | | | | | |
| 11208 | 041612 | | | FC01: | | :FLOW COMMAND = 01 - COPY TEST OPERAND |
| 11209 | | | | | | : FROM PARAMETER TABLE INTO TRN. |
| 11210 | 041612 | 004737 | 063164 | | JSR PC,EXTBK | :HANDLE OPERATOR REQUESTS |
| 11211 | 041616 | 000137 | 041600 | | JMP CTLG | :CNTL C RETURN FROM SUBROUTINE. |
| 11212 | 041622 | 032737 | 000100 | 047464 NOCTC: | BIT #100,TINST | :THE IN-LINE TEST CASE ALWAYS FOLLOWS THE SAME |
| 11213 | 041630 | 001067 | | | BNE FCRTN | :REG TEST CASE - THEREFORE DO NOT REFILL THE TRNS |
| 11214 | | | | | | : BECAUSE SXTYPES ARE ALREADY UPDATED FOR NEXT REG |
| 11215 | | | | | | : TEST CONDITION. |
| 11216 | 041632 | 004737 | 053242 | | JSR PC,PF1 | :FORM PARAMETER TABLE POINTER FROM |
| 11217 | | | | | | : PF1 FIELD OF FLOW TABLE ENTRY |
| 11218 | 041636 | 004737 | 053456 | | JSR PC,RF4 | :FORM TEST OPERAND POINTER FROM RF4 FIELD |
| 11219 | | | | | | : OF FLOW TABLE ENTRY. |
| 11220 | 041642 | 017711 | 140270 | | MOV @PTPTR,(R1) | :COPY TEST OPERAND FROM PARAMETER TABLE |
| 11221 | 041646 | 010102 | | | MOV R1,R2 | |
| 11222 | 041650 | 004737 | 053440 | | JSR PC,RF3X | :LOAD R1 WITH TWICE CONTENTS OF RF3 FIELD OF |
| 11223 | 041654 | 005701 | | | TST R1 | : FLOW TABLE ENTRY |
| 11224 | 041656 | 001454 | | | BEQ FCRTN | :BRANCH IF FIELD CONTAINS ZERO |
| 11225 | 041660 | 012737 | 177777 | 002434 | MOV #177777,DECINS | :SET FLAG TO INDICATE THAT INST IS A DECIMAL INST. |
| 11226 | 041666 | 020127 | 000002 | | CMP R1,#2 | :IS TEST OPERAND PART OF 1ST DECIMAL DESC |
| 11227 | | | | | | : OPERAND FOR INST? |

| | | | | | | | |
|-------|--------|--------|--------|--------|----------------------|--|---|
| 11228 | 041672 | 001013 | | | BNE 1\$ | | :BRANCH IF NO |
| 11229 | 041674 | 013704 | 002436 | | MOV S1TYPE,R4 | | |
| 11230 | 041700 | 006304 | | | ASL R4 | | |
| 11231 | 041702 | 006304 | | | ASL R4 | | |
| 11232 | 041704 | 006304 | | | ASL R4 | | |
| 11233 | 041706 | 006304 | | | ASL R4 | | |
| 11234 | 041710 | 000304 | | | SWAB R4 | | |
| 11235 | 041712 | 042704 | 107777 | | BIC #107777,R4 | | |
| 11236 | 041716 | 050412 | | | BIS R4,(R2) | | : 'OR' SRC1 TYPE FIELD INTO TEST OPERAND |
| 11237 | 041720 | 000433 | | | BR FCRTN | | |
| 11238 | 041722 | 020127 | 000004 | 1\$: | CMP R1,#4 | | : IS TEST OPERAND PART OF 2ND DECIMAL DESC : OPERAND FOR INST? |
| 11239 | | | | | | | :BRANCH IF NO |
| 11240 | 041726 | 001013 | | | BNE 2\$ | | |
| 11241 | 041730 | 013704 | 002440 | | MOV S2TYPE,R4 | | |
| 11242 | 041734 | 006304 | | | ASL R4 | | |
| 11243 | 041736 | 006304 | | | ASL R4 | | |
| 11244 | 041740 | 006304 | | | ASL R4 | | |
| 11245 | 041742 | 006304 | | | ASL R4 | | |
| 11246 | 041744 | 000304 | | | SWAB R4 | | |
| 11247 | 041746 | 042704 | 107777 | | BIC #107777,R4 | | |
| 11248 | 041752 | 050412 | | | BIS R4,(R2) | | : 'OR' SRC2 TYPE FIELD INTO TEST OPERAND |
| 11249 | 041754 | 000415 | | | BR FCRTN | | |
| 11250 | 041756 | 020127 | 000006 | 2\$: | CMP R1,#6 | | : IS TEST OPERAND PART OF 3RD DECIMAL DESC : OPERAND FOR INST? |
| 11251 | | | | | | | :BRANCH IF NO |
| 11252 | 041762 | 001012 | | | BNE FCRTN | | |
| 11253 | 041764 | 013704 | 002442 | | MOV S3TYPE,R4 | | |
| 11254 | 041770 | 006304 | | | ASL R4 | | |
| 11255 | 041772 | 006304 | | | ASL R4 | | |
| 11256 | 041774 | 006304 | | | ASL R4 | | |
| 11257 | 041776 | 006304 | | | ASL R4 | | |
| 11258 | 042000 | 000304 | | | SWAB R4 | | |
| 11259 | 042002 | 042704 | 107777 | | BIC #107777,R4 | | : 'OR' DST TYPE FIELD INTO TEST OPERAND |
| 11260 | 042006 | 050412 | | | BIS R4,(R2) | | |
| 11261 | | | | | | | |
| 11262 | 042010 | 062737 | 000002 | 001742 | FCRTN: ADD #2,FLOPTR | | :UPDATE FLOW TABLE POINTER TO NEXT COMMAND |
| 11263 | 042016 | 000137 | 041536 | | JMP XINST | | |
| 11264 | | | | | | | |
| 11265 | 042022 | | | | FC02: | | :FLOW COMMAND = 02 - GENERATE TEST OPERAND : FROM PARAMETER TABLE ENTRY. |
| 11266 | | | | | | | :SKIP THIS FLOW COMMAND FOR INLINE CASE |
| 11267 | 042022 | 032737 | 000100 | 047464 | BIT #100,TINST | | |
| 11268 | 042030 | 001367 | | | BNE FCRTN | | |
| 11269 | 042032 | 005737 | 041572 | | TST MTYPE | | |
| 11270 | 042036 | 001410 | | | BEQ 1\$ | | :BRANCH IF NOT TYPE 0 ENTRY |
| 11271 | 042040 | 004737 | 053242 | | JSR PC,PF1 | | :FORM PARAMETER TABLE POINTER FROM : PF1 FIELD OF FLOW TABLE ENTRY. |
| 11272 | | | | | | | :FORM TEST OPERAND POINTER FROM : RF4 FIELD OF FLOW TABLE ENTRY |
| 11273 | 042044 | 004737 | 053456 | | JSR PC,RF4 | | :COPY TEST OPERAND FROM PARAMETER TABLE |
| 11274 | | | | | | | |
| 11275 | 042050 | 017711 | 140062 | | MOV @PTPTR,(R1) | | |
| 11276 | 042054 | 000137 | 042010 | | JMP FCRTN | | |
| 11277 | 042060 | 004737 | 053242 | 1\$: | JSR PC,PF1 | | :FORM PARAMETER TABLE POINTER FROM PF1 : FIELD OF FLOW TABLE ENTRY. |
| 11278 | | | | | | | :EXECUTE PARAMETER TABLE ENTRY TO : GENERATE TEST OPERAND. |
| 11279 | 042064 | 004777 | 140046 | | JSR PC,@PTPTR | | |
| 11280 | | | | | | | |
| 11281 | 042070 | 000137 | 042010 | | JMP FCRTN | | |

| | | | | | | | | | | |
|-------|--------|--------|--------|--------|---------|----------------|--|--|---|--|
| 11282 | | | | | | | | | | |
| 11283 | 042074 | 005237 | 001424 | | REDNTC: | INC REDTC | | | :TEST CONDITION REDUNDANT - ABORT TEST. | |
| 11284 | 042100 | 005737 | 001420 | | | TST TOTTC | | | :DID TEST COUNT OVERFLOW ON LAST INCREMENT? | |
| 11285 | 042104 | 001002 | | | | BNE 1\$ | | | :BRANCH IF NO | |
| 11286 | 042106 | 005337 | 001416 | | | DEC TOTTC | | | | |
| 11287 | 042112 | 005337 | 001420 | | 1\$: | DEC TOTTC | | | | |
| 11288 | 042116 | 000137 | 052326 | | | JMP NXTTC | | | | |
| 11289 | | | | | | | | | | |
| 11290 | 042122 | | | | FC03: | | | | :FLOW COMMAND = 03 - VERIFY THAT STRING'S | |
| 11291 | | | | | | | | | : LOWER ADDRESS LIMIT FALLS WITHIN TEST BUFFER. | |
| 11292 | 042122 | 032737 | 000100 | 047464 | | BIT #100,TINST | | | :SKIP THIS FLOW COMMAND FOR INLINE CASE | |
| 11293 | 042130 | 001327 | | | | BNE FCRTN | | | | |
| 11294 | 042132 | 032737 | 000010 | 002140 | | BIT #10,SPHAND | | | :SPECIAL HANDLING REQUEST? | |
| 11295 | 042140 | 001323 | | | | BNE FCRTN | | | :SKIP THIS FLOW COMMAND IF YES. | |
| 11296 | 042142 | 004737 | 053242 | | | JSR PC,PF1 | | | :FORM PARAMETER TABLE POINTER TO STRING | |
| 11297 | | | | | | | | | : SURROUND LENGTH FROM PF1 FIELD OF FLOW | |
| 11298 | | | | | | | | | : TABLE ENTRY. | |
| 11299 | 042146 | 004737 | 053456 | | | JSR PC,RF4 | | | :FORM TEST OPERAND POINTER TO STRING.ADR FROM | |
| 11300 | | | | | | | | | : RF4 FIELD OF FLOW TABLE ENTRY | |
| 11301 | 042152 | 011101 | | | | MOV (R1),R1 | | | | |
| 11302 | 042154 | 167701 | 137756 | | | SUB @PTPTR,R1 | | | :SUBTRACT STRING.SURR.LEN FROM STRING.ADR | |
| 11303 | | | | | | | | | : TO GET STRING.SURR.ADR. | |
| 11304 | 042160 | 020127 | 000020 | | | CMP R1,#20 | | | | |
| 11305 | 042164 | 002475 | | | | BLT NXTC | | | :STRING.SURR.ADR < 20 -SKIP THIS TEST CONDITION | |
| 11306 | | | | | | | | | : (20 ALLOWS SPACE FOR IN-LINE DESCRIPTORS AT BEGINNING | |
| 11307 | 042166 | 000137 | 042010 | | | JMP FCRTN | | | | |
| 11308 | | | | | | | | | | |
| 11309 | 042172 | | | | FC04: | | | | :FLOW COMMAND = 04 - VERIFY THAT STRINGS | |
| 11310 | | | | | | | | | : UPPER ADDRESS LIMIT FALLS WITHIN TEST | |
| 11311 | | | | | | | | | : BUFFER. | |
| 11312 | 042172 | 032737 | 000100 | 047464 | | BIT #100,TINST | | | :SKIP THIS FLOW COMMAND FOR INLINE CASE | |
| 11313 | 042200 | 001035 | | | | BNE 1\$ | | | | |
| 11314 | 042202 | 032737 | 000001 | 002140 | | BIT #1,SPHAND | | | :SPECIAL HANDLING REQUEST | |
| 11315 | 042210 | 001031 | | | | BNE 1\$ | | | :SKIP VERIFICATION IF YES | |
| 11316 | 042212 | 004737 | 053242 | | | JSR PC,PF1 | | | :FORM PARAMETER TABLE POINTER TO STRING.SURR.LEN | |
| 11317 | | | | | | | | | : FROM PF1 FIELD OF FLOW TABLE ENTRY | |
| 11318 | 042216 | 004737 | 053426 | | | JSR PC,RF3 | | | :FORM TEST OPERAND POINTER TO STRING.LEN | |
| 11319 | | | | | | | | | : FROM RF3 FIELD OF FLOW TABLE ENTRY | |
| 11320 | 042222 | 011101 | | | | MOV (R1),R1 | | | | |
| 11321 | 042224 | 005737 | 002434 | | | TST DECINS | | | :IS INST A DECIMAL INST? | |
| 11322 | 042230 | 001402 | | | | BEQ 2\$ | | | :BRANCH IF NO | |
| 11323 | 042232 | 043701 | 002444 | | | BIC TYPFLD,R1 | | | :CLEAR TYPE FIELD SO AS NOT TO | |
| 11324 | | | | | | | | | : DISTORT UPPER ADDRESS CALCULATION. | |
| 11325 | 042236 | 017702 | 137674 | | 2\$: | MOV @PTPTR,R2 | | | | |
| 11326 | 042242 | 060102 | | | | ADD R1,R2 | | | :R2 NOW CONTAINS SUM OF STRING.SURR.LEN | |
| 11327 | | | | | | | | | : AND STRING.LEN | |
| 11328 | 042244 | 004737 | 053456 | | | JSR PC,RF4 | | | :FORM TEST OPERAND POINTER TO STRING.ADR | |
| 11329 | | | | | | | | | : FROM RF4 FIELD OF FLOW TABLE ENTRY. | |
| 11330 | 042250 | 061102 | | | | ADD (R1),R2 | | | :R2 NOW CONTAINS STRING.ADR + STRING.LEN | |
| 11331 | | | | | | | | | : + STRING.SURR.ADR | |
| 11332 | 042252 | 005737 | 001760 | | | TST RANDOM | | | :RANDOM EXERCISE MODE? | |
| 11333 | 042256 | 001403 | | | | BEQ 3\$ | | | :BRANCH IF NO. | |
| 11334 | 042260 | 020237 | 001644 | | | CMP R2,RTBLEN | | | :COMPARE ADDRESS WITH END OF RANDOM TEST BUFFER | |
| 11335 | 042264 | 000402 | | | | BR 4\$ | | | | |

| | | | | | | | |
|-------|--------|--------|--------|--------|-------------------|--|------------------------------------|
| 11336 | 042266 | 020237 | 001642 | 3\$: | CMP R2, TBLEN | ; DOES THIS ADDRESS EXCEED TEST BUFFER | |
| 11337 | | | | | | ; LENGTH ? | |
| 11338 | 042272 | 003032 | | 4\$: | BGT NXTC | ; YES - SKIP THIS TEST CONDITION | |
| 11339 | 042274 | 000137 | 042010 | 1\$: | JMP FCRTN | | |
| 11340 | | | | | | | |
| 11341 | 042300 | | | FC05: | | ; FLOW COMMAND = 05 - ADJUST TEST OPERANDS | |
| 11342 | | | | | | ; TO INCLUDE BASE ADDRESS OF TEST BUFFER. | |
| 11343 | 042300 | 032737 | 000100 | 047464 | BIT #100, TINST | ; SKIP THIS FLOW COMMAND FOR INLINE CASE | |
| 11344 | 042306 | 001022 | | | BNE EC05 | | |
| 11345 | 042310 | 004737 | 053340 | | JSR PC, RF1 | ; FORM 1ST TEST OPERAND POINTER FROM RF1 | |
| 11346 | | | | | | ; FIELD OF FLOW TABLE ENTRY | |
| 11347 | 042314 | 063711 | 001640 | | ADD TBADR, (R1) | ; ADD TEST BUFFER BASE ADDRESS TO OPERAND | |
| 11348 | 042320 | 004737 | 053372 | | JSR PC, RF2 | ; FORM 2ND TEST OPERAND POINTER FROM RF2 | |
| 11349 | | | | | | ; FIELD OF FLOW TABLE ENTRY | |
| 11350 | 042324 | 020127 | 003624 | | CMP R1, #TRN | ; IF R1 STILL POINTS TO #TRN THEN THERE WAS | |
| 11351 | | | | | | ; ONLY ONE TEST OPERAND TO BE UPDATED | |
| 11352 | 042330 | 001411 | | | BEQ EC05 | ; UPDATING COMPLETE | |
| 11353 | 042332 | 063711 | 001640 | | ADD TBADR, (R1) | ; ADD TEST BUFFER BASE ADDRESS TO OPERAND | |
| 11354 | 042336 | 004737 | 053426 | | JSR PC, RF3 | ; FORM 3RD TEST OPERAND POINTER FROM RF3 | |
| 11355 | | | | | | ; FIELD OF FLOW TABLE ENTRY. | |
| 11356 | 042342 | 020127 | 003624 | | CMP R1, #TRN | ; WAS THERE A THIRD ENTRY? | |
| 11357 | 042346 | 001402 | | | BEQ EC05 | ; NO - UPDATING COMPLETE | |
| 11358 | 042350 | 063711 | 001640 | | ADD TBADR, (R1) | ; ADD TEST BUFFER BASE ADDRESS TO OPERAND | |
| 11359 | 042354 | 000137 | 042010 | EC05: | JMP FCRTN | | |
| 11360 | | | | | | | |
| 11361 | 042360 | 005237 | 001422 | NXTC: | INC INVTC | ; TEST CONDITION INVALID - ABORT TEST | |
| 11362 | 042364 | 005737 | 001420 | | TST TOTTC | ; DID TEST COUNT OVERFLOW ON LAST INCREMENT? | |
| 11363 | 042370 | 001002 | | | BNE 1\$ | ; BRANCH IF NO | |
| 11364 | 042372 | 005337 | 001416 | | DEC TOTTC | | |
| 11365 | 042376 | 005337 | 001420 | 1\$: | DEC TOTTC | | |
| 11366 | 042402 | 000137 | 052326 | | JMP NXTC | | |
| 11367 | | | | | | | |
| 11368 | 042406 | | | FC06: | | ; FLOW COMMAND = 06 - INITIALIZE TEST BUFFER | |
| 11369 | | | | | | ; TO AND INCREMENTING SEQUENCE. | |
| 11370 | 042406 | 032737 | 000100 | 047464 | BIT #100, TINST | | |
| 11371 | 042414 | 001402 | | | BEQ 4\$ | | |
| 11372 | 042416 | 004737 | 062644 | | JSR PC, RRNGSV | ; RESTORE RANDOM # GEN TO STATE V | |
| 11373 | 042422 | 004737 | 062620 | 4\$: | JSR PC, SRNGSV | ; SAVE RANDOM # GENERATOR STATE AS STATE V | |
| 11374 | 042426 | 013737 | 001640 | 001754 | MOV TBADR, TBEND | | |
| 11375 | 042434 | 005737 | 001760 | | TST RANDOM | ; RANDOM EXERCISE MODE? | |
| 11376 | 042440 | 001404 | | | BEQ 2\$ | ; BRANCH IF NO | |
| 11377 | 042442 | 063737 | 001644 | 001754 | ADD RTBLEN, TBEND | ; THE BUFFER SIZE FOR RANDOM EXERCISE MODE | |
| 11378 | | | | | | ; IS FIXED AT 10000 BYTES. | |
| 11379 | 042450 | 000403 | | | BR 3\$ | | |
| 11380 | 042452 | 063737 | 001642 | 001754 | 2\$: | ADD TBLEN, TBEND | ; SETUP A POINTER TO END OF BUFFER |
| 11381 | 042460 | 013701 | 001640 | 3\$: | MOV TBADR, R1 | | |
| 11382 | 042464 | 013702 | 001640 | | MOV TBADR, R2 | ; POINT R1 & R2 TO START OF BUFFER | |
| 11383 | 042470 | 013721 | 001660 | | MOV INCSQ1, (R1)+ | | |
| 11384 | 042474 | 013721 | 001662 | | MOV INCSQ2, (R1)+ | ; LOAD THE FIRST TWO BUFFER LOCATIONS | |
| 11385 | 042500 | 012211 | | 1\$: | MOV (R2)+, (R1) | | |
| 11386 | 042502 | 061221 | | | ADD (R2), (R1)+ | ; CONTENTS OF NEXT LOC = SUM OF CONTENTS | |
| 11387 | | | | | | ; OF PREVIOUS 2 LOCATIONS. | |
| 11388 | 042504 | 023701 | 001754 | | CMP TBEND, R1 | | |
| 11389 | 042510 | 003373 | | | BGT 1\$ | | |

11390 042512 000137 042010

JMP FCRTN

| | | | | | | | | | |
|-------|--------|--------|--------|--------|------|----------------------|--|--|--|
| 11392 | 042516 | | | | | FC07: | | | :FLOW COMMAND = 07 - INSERT STRING IN |
| 11393 | | | | | | | | | : TEST BUFFER. |
| 11394 | 042516 | 005037 | 002472 | | | CLR SAVSRF | | | :RANDOM EXERCISE MODE? |
| 11395 | 042522 | 005737 | 001760 | | | TST RANDOM | | | :BRANCH IF NO |
| 11396 | 042526 | 001406 | | | | BEQ 1\$ | | | :IS INST A CHAR STRING INST? |
| 11397 | 042530 | 023727 | 002272 | 000011 | | CMP OCTIC,#11 | | | :BRANCH IF NO |
| 11398 | 042536 | 101002 | | | | BHI 1\$ | | | :IN RANDOM MODE, NO CHAR STRINGS NEED TO BE |
| 11399 | 042540 | 000137 | 042010 | | | 2\$: JMP FCRTN | | | : INSERTED FOR CHAR TYPE INSTRUCTIONS. |
| 11400 | | | | | | | | | : RANDOM CHAR BYTES ARE DERIVED BY |
| 11401 | | | | | | | | | : RANDOMIZING THE 'SEED' CONSTANTS USED |
| 11402 | | | | | | | | | : TO INITIALIZE THE ENTIRE BUFFER. |
| 11403 | | | | | | | | | :SPECIAL HANDLING REQUEST? |
| 11404 | 042544 | 032737 | 000020 | 002140 | 1\$: | BIT #20,SPHAND | | | :SKIP INSERTING STRINGS IF YES |
| 11405 | 042552 | 001372 | | | | BNE 2\$ | | | :FORM PARAMETER TABLE POINTER TO STRING |
| 11406 | 042554 | 004737 | 053242 | | | JSR PC,PF1 | | | : DESCRIPTOR FROM PF1 FIELD OF FLOW TABLE ENTRY |
| 11407 | | | | | | | | | :FORM TEST OPERAND POINTER TO STRING.LEN |
| 11408 | 042560 | 004737 | 053426 | | | JSR PC,RF3 | | | : FROM RF3 FIELD OF FLOW TABLE ENTRY |
| 11409 | | | | | | | | | :SAVE POINTER |
| 11410 | 042564 | 010137 | 002104 | | | MOV R1,TRL | | | :FORM TEST OPERAND POINTER TO STRING.ADR |
| 11411 | 042570 | 004737 | 053456 | | | JSR PC,RF4 | | | : FROM RF4 FIELD OF FLOW |
| 11412 | | | | | | | | | : TABLE ENTRY. |
| 11413 | | | | | | | | | :SAVE POINTER |
| 11414 | 042574 | 010137 | 002102 | | | MOV R1,TRA | | | :SAVE RANDOM NUMBER GEN. STATE X |
| 11415 | 042600 | 004737 | 062430 | | | JSR PC,SRNGSX | | | :INSERT STRING IN TEST BUFFER |
| 11416 | 042604 | 004537 | 043072 | | | JSR R5,ISTG | | | :POINTER TO STRING DESCRIPTOR |
| 11417 | 042610 | 002136 | | | | PTPTR | | | :POINTER TO STRING.LEN |
| 11418 | 042612 | 002104 | | | | TRL | | | :POINTER TO STRING.ADR |
| 11419 | 042614 | 002102 | | | | TRA | | | |
| 11420 | | | | | | | | | |
| 11421 | 042616 | 005737 | 002472 | | | TST SAVSRF | | | :SAVE STRING FOR ERROR PRINTOUT? |
| 11422 | 042622 | 001436 | | | | BEQ A3X | | | |
| 11423 | 042624 | 004737 | 053242 | | | JSR PC,PF1 | | | :YES - RESTORE POINTER TO STRING DESCRIPTOR |
| 11424 | 042630 | 004737 | 053426 | | | JSR PC,RF3 | | | :RESTORE POINTER TO STRING LENGTH |
| 11425 | 042634 | 010137 | 002104 | | | MOV R1,TRL | | | |
| 11426 | 042640 | 005737 | 002270 | | | TST FILLS2 | | | :WHERE SHOULD STRING BE STORED? |
| 11427 | 042644 | 001406 | | | | BEQ 4\$ | | | |
| 11428 | 042646 | 012737 | 002502 | 002470 | | MOV #INSRC2+2,SAVPTR | | | :STORE STRING IN BUFSR2 |
| 11429 | 042654 | 011137 | 002500 | | | MOV (R1),INSRC2 | | | :SAVE STRING LEN IN BUFFER DESCRIPTOR |
| 11430 | 042660 | 000410 | | | | BR 5\$ | | | |
| 11431 | 042662 | 012737 | 002476 | 002470 | 4\$: | MOV #INSRC1+2,SAVPTR | | | :SAVE STRING IN BUFSR1 |
| 11432 | 042670 | 012737 | 177777 | 002270 | | MOV #177777,FILLS2 | | | :SIGNAL THAT BUFSR1 IS OCCUPIED. |
| 11433 | 042676 | 011137 | 002474 | | | MOV (R1),INSRC1 | | | :SAVE STRING LEN IN BUFFER DESCRIPTOR |
| 11434 | 042702 | 004737 | 062454 | | | 5\$: JSR PC,RRNGSX | | | :RESTORE RANDOM NUMBER GEN TO STATE X. |
| 11435 | 042706 | 004537 | 043072 | | | JSR R5,ISTG | | | :INSERT STRING IN SAVE BUFFER |
| 11436 | 042712 | 002136 | | | | PTPTR | | | :POINTER TO STRING DESCRIPTOR |
| 11437 | 042714 | 002104 | | | | TRL | | | :POINTER TO STRING LEN |
| 11438 | 042716 | 002470 | | | | SAVPTR | | | :POINTER TO STRING ADDRESS (EITHER BUFSR1 OR BUFSR2) |
| 11439 | | | | | | | | | |
| 11440 | 042720 | 062737 | 000002 | 001742 | A3X: | ADD #2,FLOPTR | | | :LOOK AT NEXT FLOW COMMAND? |
| 11441 | 042726 | 017701 | 137010 | | | MOV @FLOPTR,R1 | | | |
| 11442 | 042732 | 042701 | 017777 | | | BIC #017777,R1 | | | |
| 11443 | 042736 | 005701 | | | | TST R1 | | | :IS IT = 0 - A CONTINUATION OF THE 07 |
| 11444 | | | | | | | | | : COMMAND? |
| 11445 | 042740 | 001050 | | | | BNE 2\$ | | | :NO - DON'T INSERT SURROUND STRINGS |

| | | | | | | | |
|-------|--------|--------|--------|--------|-------|--------------------|--|
| 11446 | 042742 | 004737 | 053242 | | | JSR PC,PF1 | :YES - FORM PARAMETER TABLE POINTER |
| 11447 | | | | | | | : TO STRING.SURR.LEN |
| 11448 | 042746 | 013737 | 002136 | 002046 | | MOV PTPTR,SURLEN | |
| 11449 | 042754 | 004737 | 053314 | | | JSR PC,PF2 | :FORM PARAMETER TABLE POINTER TO SURR DATA |
| 11450 | | | | | | | : DESCRIPTOR |
| 11451 | 042760 | 017737 | 137120 | 002104 | | MOV @TRL,TRL | :FORM SURR.ADR (UPPER PORTION) |
| 11452 | 042766 | 005737 | 002434 | | | TST DECINS | :INST = DECIMAL? |
| 11453 | 042772 | 001403 | | | | BEQ 3\$ | :BRANCH IF NO |
| 11454 | 042774 | 042737 | 070000 | 002104 | | BIC #070000,TRL | :CLEAR TYPE FIELD FROM STRING LENGTH |
| 11455 | 043002 | 067737 | 137074 | 002104 | 3\$: | ADD @TRA,TRL | :SURR.ADR = STRING.ADR + STRING.LEN |
| 11456 | 043010 | 012737 | 002104 | 002050 | | MOV #TRL,SURADR | |
| 11457 | 043016 | 004537 | 043072 | | | JSR R5,ISTG | :INSERT UPPER HALF OF SURR STRING |
| 11458 | 043022 | 002136 | | | | PTPTR | :POINTER TO SURR.DATA DESCRIPTOR |
| 11459 | 043024 | 002046 | | | | SURLEN | :POINTER TO SURR.LEN |
| 11460 | 043026 | 002050 | | | | SURADR | :POINTER TO SURR.ADR |
| 11461 | 043030 | 017737 | 137046 | 002104 | | MOV @TRA,TRL | :FORM SURR STRING ADDRESS (LOWER PORTION) |
| 11462 | 043036 | 167737 | 137004 | 002104 | | SUB @SURLEN,TRL | :SURR.ADR = STRING.ADR - SURR.LEN |
| 11463 | 043044 | 004537 | 043072 | | | JSR R5,ISTG | :INSERT LOWER HALF OF SURR STRING |
| 11464 | 043050 | 002136 | | | | PTPTR | |
| 11465 | 043052 | 002046 | | | | SURLEN | |
| 11466 | 043054 | 002050 | | | | SURADR | |
| 11467 | 043056 | 000137 | 042010 | | 1\$: | JMP FCRTN | |
| 11468 | 043062 | 162737 | 000002 | 001742 | 2\$: | SUB #2,FLOPTR | :RESTORE FLOW COMMAND POINTER |
| 11469 | 043070 | 000772 | | | | BR 1\$ | |
| 11470 | | | | | | | |
| 11471 | 043072 | | | | ISTG: | | :SUBROUTINE TO INSERT STRING IN TEST BUFFER. |
| 11472 | 043072 | 013501 | | | | MOV @(R5)+,R1 | |
| 11473 | 043074 | 012137 | 002106 | | | MOV (R1)+,STGDS1 | :GET STRING DATA DESCRIPTOR |
| 11474 | 043100 | 011137 | 002110 | | | MOV (R1),STGDS2 | |
| 11475 | 043104 | 013501 | | | | MOV @(R5)+,R1 | :GET STRING.LEN |
| 11476 | 043106 | 011137 | 002112 | | | MOV (R1),STGLN | |
| 11477 | 043112 | 032737 | 000002 | 002140 | | BIT #2,SPHAND | :SPECIAL HANDLING REQUEST? |
| 11478 | 043120 | 001403 | | | | BEQ 1\$ | :BRANCH IF NO |
| 11479 | 043122 | 042737 | 100000 | 002112 | | BIC #100000,STGLN | :YES - STRIP BIT 15 FROM LENGTH |
| 11480 | 043130 | 013501 | | | 1\$: | MOV @(R5)+,R1 | :SETUP STRING STARTING ADDRESS |
| 11481 | 043132 | 011137 | 002114 | | | MOV (R1),STGAD | |
| 11482 | 043136 | 013701 | 002106 | | | MOV STGDS1,R1 | |
| 11483 | 043142 | 042737 | 160000 | 002106 | | BIC #160000,STGDS1 | :STRIP OFF 'TYPE' FROM 1ST WORD OF DATA DESCRIPTOR |
| 11484 | 043150 | 042701 | 017777 | | | BIC #17777,R1 | :LOOK ONLY AT DESCRIPTOR TYPE |
| 11485 | 043154 | 005701 | | | | TST R1 | |
| 11486 | 043156 | 001426 | | | | BEQ DSTYP0 | :DATA DESCRIPTOR IS TYPE 0. |
| 11487 | 043160 | 022701 | 020000 | | | CMP #020000,R1 | |
| 11488 | 043164 | 001445 | | | | BEQ DSTYP1 | :DATA DESCRIPTOR IS TYPE 1 |
| 11489 | 043166 | 022701 | 040000 | | | CMP #040000,R1 | |
| 11490 | 043172 | 001461 | | | | BEQ DSTYP2 | :DATA DESCRIPTOR IS TYPE 2 |
| 11491 | 043174 | 022701 | 060000 | | | CMP #060000,R1 | |
| 11492 | 043200 | 001516 | | | | BEQ DSTYP3 | :DATA DESCRIPTOR IS TYPE 3 |
| 11493 | 043202 | 022701 | 100000 | | | CMP #100000,R1 | : |
| 11494 | 043206 | 001404 | | | | BEQ 2\$ | :DATA DESCRIPTOR IS TYPE 4 |
| 11495 | 043210 | 022701 | 120000 | | | CMP #120000,R1 | |
| 11496 | 043214 | 001405 | | | | BEQ 3\$ | :DATA DESCRIPTOR IS TYPE 5 |
| 11497 | 043216 | 000000 | | | | HALT | :**DATA DESCRIPTOR NOT TYPE 0,1,2,3,4, OR 5. |
| 11498 | 043220 | 005037 | 001770 | | 2\$: | CLR RANDTA | |
| 11499 | 043224 | 000137 | 044334 | | | JMP DSTYP4 | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|---|
| 11500 | 043230 | 000137 | 045402 | | 3\$: | JMP DSTYP5 | |
| 11501 | | | | | | | |
| 11502 | 043234 | | | | DSTYP0: | | : ALL BYTES OF STRING ARE IDENTICAL |
| 11503 | 043234 | 005737 | 002434 | | | TST DECINS | : INST = DECIMAL? |
| 11504 | 043240 | 001403 | | | | BEQ 3\$ | : BRANCH IF NO |
| 11505 | 043242 | 042737 | 070000 | 002112 | | BIC #070000,STGLN | : CLEAR TYPE FIELD STRING LENGTH WORD |
| 11506 | 043250 | 013701 | 002114 | | 3\$: | MOV STGAD,R1 | : R1 CONTAINS STRING STARTING ADDRESS |
| 11507 | 043254 | 005737 | 002112 | | 1\$: | TST STGLN | : ENTIRE STRING BEEN INSERTED? |
| 11508 | 043260 | 001405 | | | | BEQ 2\$ | : YES |
| 11509 | 043262 | 113721 | 002106 | | | MOVB STGDS1,(R1)+ | : NO - MOVE STRING DATA BYTE INTO NEXT |
| 11510 | | | | | | | : TEST BUFFER LOCATION |
| 11511 | 043266 | 005337 | 002112 | | | DEC STGLN | |
| 11512 | 043272 | 000770 | | | | BR 1\$ | |
| 11513 | 043274 | 000137 | 045500 | | 2\$: | JMP EISTG | |
| 11514 | | | | | | | |
| 11515 | 043300 | | | | DSTYP1: | | : STRING BYTE N = STRING BYTE N-1 + INC. |
| 11516 | 043300 | 013701 | 002114 | | | MOV STGAD,R1 | : R1 CONTAINS STRING STARTING ADDRESS |
| 11517 | 043304 | 005737 | 002112 | | 1\$: | TST STGLN | : ENTIRE STRING BEEN INSERTED? |
| 11518 | 043310 | 001410 | | | | BEQ 2\$ | : YES |
| 11519 | 043312 | 113721 | 002110 | | | MOVB STGDS2,(R1)+ | |
| 11520 | 043316 | 063737 | 002106 | 002110 | | ADD STGDS1,STGDS2 | : NO FORM AND INSERT NEXT STRING BYTE |
| 11521 | 043324 | 005337 | 002112 | | | DEC STGLN | |
| 11522 | 043330 | 000765 | | | | BR 1\$ | |
| 11523 | 043332 | 000137 | 045500 | | 2\$: | JMP EISTG | |
| 11524 | | | | | | | |
| 11525 | 043336 | | | | DSTYP2: | | : INSERT BYTES FROM GIVEN STRING |
| 11526 | 043336 | 013701 | 002114 | | | MOV STGAD,R1 | : SET R1 TO STARTING ADDRESS OF STRING TO BE |
| 11527 | | | | | | | : FORMED IN TEST BUFFER. |
| 11528 | 043342 | 005737 | 002106 | | | TST STGDS1 | : IS GIVEN STRING LENGTH = 0? |
| 11529 | 043346 | 001431 | | | | BEQ 3\$ | : YES - DON'T DO ANY INSERTING |
| 11530 | 043350 | 013737 | 002106 | 002116 | | MOV STGDS1,SAVSL | : NO - SAVE STRING LENGTH IN CASE STRING |
| 11531 | 043356 | 013737 | 002110 | 002120 | | MOV STGDS2,SAVSA | : TO BE FORMED IS LONGER THAN GIVEN STRING. |
| 11532 | 043364 | 005737 | 002112 | | 1\$: | TST STGLN | : ENTIRE STRING BEEN INSERTED? |
| 11533 | 043370 | 001420 | | | | BEQ 3\$ | : YES |
| 11534 | 043372 | 117721 | 136512 | | | MOVB @STGDS2,(R1)+ | : NO - INSERT STRING BYTE FROM GIVEN STRING |
| 11535 | 043376 | 005237 | 002110 | | | INC STGDS2 | : UPDATE GIVEN STRING ADDRESS TO NEXT BYTE |
| 11536 | 043402 | 005337 | 002106 | | | DEC STGDS1 | |
| 11537 | 043406 | 001006 | | | | BNE 2\$ | : ALL BYTES IN GIVEN STRING USED? |
| 11538 | 043410 | 013737 | 002116 | 002106 | | MOV SAVSL,STGDS1 | : YES - STRING BEING FORMED IS LONGER THAN |
| 11539 | 043416 | 013737 | 002120 | 002110 | | MOV SAVSA,STGDS2 | : GIVEN STRING. RESET STRING ADDRESS |
| 11540 | | | | | | | : BACK TO BEGINNING OF GIVEN STRING. |
| 11541 | 043424 | 005337 | 002112 | | 2\$: | DEC STGLN | : DECREMENT COUNT OF # OF CHARACTERS YET |
| 11542 | | | | | | | : TO BE INSERTED. |
| 11543 | 043430 | 000755 | | | | BR 1\$ | |
| 11544 | 043432 | 000137 | 045500 | | 3\$: | JMP EISTG | |
| 11545 | | | | | | | |
| 11546 | 043436 | | | | DSTYP3: | | : DECIMAL STRING - ALL DIGITS IDENTICAL |
| 11547 | 043436 | 022737 | 010000 | 002106 | | CMP #10000,STGDS1 | : IS FORMED STRING TO BE PACKED OR ZONED DECIMAL? |
| 11548 | 043444 | 101070 | | | | BHI PTYP3P | |
| 11549 | 043446 | | | | TYP3Z: | | : ZONED |
| 11550 | 043446 | 013701 | 002112 | | | MOV STGLN,R1 | : GET STRING DESC. TYPE FIELD |
| 11551 | 043452 | 000301 | | | | SWAB R1 | |
| 11552 | 043454 | 006201 | | | | ASR R1 | |
| 11553 | 043456 | 006201 | | | | ASR R1 | |

| | | | | | | |
|-------|--------|--------|--------|--------|------------------------|--|
| 11554 | 043460 | 006201 | | | ASR R1 | |
| 11555 | 043462 | 006201 | | | ASR R1 | |
| 11556 | 043464 | 042701 | 177770 | | BIC #177770,R1 | |
| 11557 | 043470 | 010137 | 002304 | | MOV R1,STGTYP | ;SAVE TYPE |
| 11558 | 043474 | 042737 | 177740 | 002112 | BIC #177740,STGLN | ;STRIP TYPE OFF STRING LENGTH WORD |
| 11559 | 043502 | 013737 | 002106 | 002124 | MOV STGDS1,SIGN | ;STRIP OFF SIGN FROM DATA DESCRIPTOR WORD 1 |
| 11560 | 043510 | 042737 | 177760 | 002124 | BIC #177760,SIGN | |
| 11561 | 043516 | 006237 | 002106 | | ASR STGDS1 | ;GET AND RIGHT ADJUST ZONED DATA BYTE |
| 11562 | 043522 | 006237 | 002106 | | ASR STGDS1 | |
| 11563 | 043526 | 006237 | 002106 | | ASR STGDS1 | |
| 11564 | 043532 | 006237 | 002106 | | ASR STGDS1 | |
| 11565 | 043536 | 042737 | 177400 | 002106 | BIC #177400,STGDS1 | ;GOT ZONED DATA BYTE (HIGH NIBBLE & DIGIT) |
| 11566 | | | | | | ; RIGHT ADJUSTED IN STGDS1. |
| 11567 | 043544 | 013702 | 002106 | | MOV STGDS1,R2 | |
| 11568 | 043550 | 042702 | 177760 | | BIC #177760,R2 | |
| 11569 | 043554 | 010237 | 002302 | | MOV R2,STGDIG | ;SAVE JUST THE DIGIT IN STGDIG |
| 11570 | 043560 | 005737 | 002112 | | TST STGLN | ;STRING TO BE FORMED HAVE 0 LENGTH? |
| 11571 | 043564 | 001535 | | | BEQ TFS | ;YES - NOTE: A ZERO LENGTH ZONED |
| 11572 | | | | | | ; STRING OCCUPIES NO MEMORY (EXCEPT SEPARATE TYPES). |
| 11573 | 043566 | 013701 | 002114 | | MOV STGAD,R1 | ;SET R1 TO STARTING ADDRESS OF STRING |
| 11574 | | | | | | ; TO BE FORMED IN TEST BUFFER. |
| 11575 | 043572 | 113721 | 002106 | 1\$: | MOVB STGDS1,(R1)+ | ;NO - INSERT NEXT ZONED DATA BYTE. |
| 11576 | 043576 | 005337 | 002112 | | DEC STGLN | |
| 11577 | 043602 | 005737 | 002112 | | TST STGLN | ;ENTIRE STRING BEEN INSERTED? |
| 11578 | 043606 | 001371 | | | BNE 1\$ | ;NO |
| 11579 | 043610 | 013702 | 002304 | | MOV STGTYP,R2 | ;YES - INSERT SIGN BYTE |
| 11580 | 043614 | 006302 | | | ASL R2 | |
| 11581 | 043616 | 062702 | 002306 | | ADD #TYPTAB,R2 | |
| 11582 | 043622 | 000172 | 000000 | | JMP @ (R2) | ;VECTOR TO APPROPRIATE STRING TYPE ROUTINE |
| 11583 | | | | | | ; TO ENTER SIGN BYTE. |
| 11584 | | | | | | |
| 11585 | 043626 | 000137 | 044114 | | PTYP3P: JMP TYP3P | |
| 11586 | | | | | | |
| 11587 | 043632 | 142741 | 000360 | | TYP3Z: BICB #360,-(R1) | ;SIGNED ZONED |
| 11588 | | | | | | ;CLEAR OUT THE HIGH NIBBLE OF LEAST |
| 11589 | | | | | | ; SIGNIFICANT STRING BYTE. |
| 11590 | 043636 | 006337 | 002124 | | ASL SIGN | |
| 11591 | 043642 | 006337 | 002124 | | ASL SIGN | |
| 11592 | 043646 | 006337 | 002124 | | ASL SIGN | |
| 11593 | 043652 | 006337 | 002124 | | ASL SIGN | |
| 11594 | 043656 | 153711 | 002124 | | BISB SIGN,(R1) | ; 'OR' IN SIGN |
| 11595 | 043662 | 000512 | | | BR EXTYP | |
| 11596 | 043664 | 000511 | | | TYPUZ: BR EXTYP | ;UNSIGNED ZONED - NO ACTION REQUIRED |
| 11597 | | | | | | |
| 11598 | 043666 | | | | TYPTO: | ;TRAILING OVERPUNCHED |
| 11599 | 043666 | 022737 | 000003 | 002124 | CMP #3,SIGN | ;IS SIGN = +? |
| 11600 | 043674 | 001416 | | | BEQ 1\$ | ;BRANCH IF YES |
| 11601 | 043676 | 004737 | 063464 | | JSR PC,RN | ;RANDOMLY SELECT FROM 2 NEGATIVE SIGN TABLES |
| 11602 | 043702 | 032700 | 000001 | | BIT #1,R0 | |
| 11603 | 043706 | 001403 | | | BEQ 3\$ | |
| 11604 | 043710 | 012702 | 002342 | | MOV #NEGTB1,R2 | |
| 11605 | 043714 | 000402 | | | BR 2\$ | |
| 11606 | 043716 | 012702 | 002330 | 3\$: | MOV #NEGTAB,R2 | ;SIGN IS NEGATIVE |
| 11607 | 043722 | 063702 | 002302 | 2\$: | ADD STGDIG,R2 | |

| | | | | | | | |
|-------|--------|--------|--------|--------|-----------------|---------------|---|
| 11608 | 043726 | 111241 | | | MOVB (R2),-(R1) | | :COPY ENCODED SIGN FROM TABLE INTO STRING |
| 11609 | 043730 | 000467 | | | BR EXTYP | | |
| 11610 | 043732 | 004737 | 063464 | 1\$: | JSR PC,RN | | :RANDOMLY SELECT FROM 3 POSITIVE SIGN TABLES. |
| 11611 | 043736 | 032700 | 000001 | | BIT #1,R0 | | |
| 11612 | 043742 | 001403 | | | BEQ 4\$ | | |
| 11613 | 043744 | 012702 | 002366 | | MOV #POSTB1,R2 | | |
| 11614 | 043750 | 000764 | | | BR 2\$ | | |
| 11615 | 043752 | 032700 | 000002 | 4\$: | BIT #2,R0 | | |
| 11616 | 043756 | 001403 | | | BEQ 5\$ | | |
| 11617 | 043760 | 012702 | 002400 | | MOV #POSTB2,R2 | | |
| 11618 | 043764 | 000756 | | | BR 2\$ | | |
| 11619 | 043766 | 012702 | 002354 | 5\$: | MOV #POSTAB,R2 | | |
| 11620 | 043772 | 000753 | | | BR 2\$ | | |
| 11621 | | | | | | | |
| 11622 | 043774 | | | | TYPLO: | | :LEADING OVERPUNCHED |
| 11623 | 043774 | 013701 | 002114 | | MOV STGAD,R1 | | :SETUP POINTER TO MOST SIGN. BYTE OF STRING |
| 11624 | 044000 | 005201 | | | INC R1 | | |
| 11625 | 044002 | 000731 | | | BR TYPTO | | |
| 11626 | | | | | | | |
| 11627 | 044004 | | | | TYPTS: | | :TRAILING SEPARATE |
| 11628 | 044004 | 022737 | 000003 | 002124 | CMP #3,SIGN | | :IS SIGN + |
| 11629 | 044012 | 001403 | | | BEQ 1\$ | | :BRANCH IF YES |
| 11630 | 044014 | 112711 | 000055 | | MOVB #055,(R1) | | :SIGN = -; COPY SIGN BYTE INTO STRING |
| 11631 | 044020 | 000433 | | | BR EXTYP | | |
| 11632 | 044022 | 004737 | 063464 | 1\$: | JSR PC,RN | | :RANDOMLY SELECT BETWEEN 2 POSITIVE SEPARATE SIGNS. |
| 11633 | 044026 | 032700 | 000001 | | BIT #1,R0 | | |
| 11634 | 044032 | 001403 | | | BEQ 2\$ | | |
| 11635 | 044034 | 112711 | 000040 | | MOVB #040,(R1) | | |
| 11636 | 044040 | 000423 | | | BR EXTYP | | |
| 11637 | 044042 | 112711 | 000053 | 2\$: | MOVB #053,(R1) | | :SIGN = +; COPY SIGN BYTE INTO BYTE STRING |
| 11638 | 044046 | 000420 | | | BR EXTYP | | |
| 11639 | | | | | | | |
| 11640 | 044050 | | | | TYPLS: | | :LEADING SEPARATE |
| 11641 | 044050 | 013701 | 002114 | | MOV STGAD,R1 | | :SETUP POINTER TO BYTE BEFORE MOST SIGN. |
| 11642 | | | | | | | : DIGIT OF STRING. |
| 11643 | 044054 | 005301 | | | DEC R1 | | |
| 11644 | 044056 | 000752 | | | BR TYPTS | | |
| 11645 | | | | | | | |
| 11646 | 044060 | | | | TFS: | | :0 LENGTH STRING |
| 11647 | 044060 | 022737 | 000005 | 002304 | CMP #5,STGTYP | | :IS STRING TYPE = LEADING SEPARATE? |
| 11648 | 044066 | 001001 | | | BNE 1\$ | | :BRANCH IF NO |
| 11649 | 044070 | 000767 | | | BR TYPLS | | :INSERT SIGN AT 'A-1' |
| 11650 | 044072 | 022737 | 000004 | 002304 | 1\$: | CMP #4,STGTYP | :IS STRING TYPE = TRAILING SEPARATE |
| 11651 | 044100 | 001003 | | | BNE EXTYP | | :BRANCH IF NO |
| 11652 | 044102 | 013701 | 002114 | | MOV STGAD,R1 | | :INSERT SIGN AT 'A' |
| 11653 | 044106 | 000736 | | | BR TYPTS | | |
| 11654 | 044110 | 000137 | 045500 | | EXTYP: | JMP EISTG | |
| 11655 | | | | | | | |
| 11656 | 044114 | | | | TYP3P: | | :PACKED |
| 11657 | 044114 | 013701 | 002112 | | MOV STGLN,R1 | | :GET STRING DESC. TYPE FIELD |
| 11658 | 044120 | 000301 | | | SWAB R1 | | |
| 11659 | 044122 | 006201 | | | ASR R1 | | |
| 11660 | 044124 | 006201 | | | ASR R1 | | |
| 11661 | 044126 | 006201 | | | ASR R1 | | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|--|
| 11662 | 044130 | 006201 | | | | ASR R1 | |
| 11663 | 044132 | 042701 | 177770 | | | BIC #177770,R1 | |
| 11664 | 044136 | 010137 | 002304 | | | MOV R1,STGTYP | :SAVE TYPE |
| 11665 | 044142 | 042737 | 177740 | 002112 | | BIC #177740,STGLN | :STRIP TYPE OFF STRING LENGTH WORD |
| 11666 | 044150 | 113737 | 002106 | 002124 | | MOVB STGDS1,SIGN | :LOAD SIGN WITH DIGIT-SIGN BYTE |
| 11667 | 044156 | 113701 | 002106 | | | MOVB STGDS1,R1 | :FORM DIGIT-DIGIT BYTE |
| 11668 | 044162 | 006201 | | | | ASR R1 | :RIGHT ADJUST DIGIT |
| 11669 | 044164 | 006201 | | | | ASR R1 | |
| 11670 | 044166 | 006201 | | | | ASR R1 | |
| 11671 | 044170 | 006201 | | | | ASR R1 | |
| 11672 | 044172 | 042701 | 177760 | | | BIC #177760,R1 | :CLEAR OUT ALL BUT DIGIT |
| 11673 | 044176 | 042737 | 177417 | 002106 | | BIC #177417,STGDS1 | :CLEAR OUT ALL BUT DIGIT IN DATA DESCRIPTOR WORD 1 |
| 11674 | 044204 | 050137 | 002106 | | | BIS R1,STGDS1 | :STGDS1 NOW CONTAINS DIGIT-DIGIT BYTE |
| 11675 | 044210 | 110137 | 002110 | | | MOVB R1,STGDS2 | :SAVE 0000-DIGIT BYTE IN CASE STRING LENGTH |
| 11676 | | | | | | | : IS EVEN - I.E. MOST SIGNIF DIGIT BYTE CONTAINS |
| 11677 | | | | | | | : ONLY A SINGLE DIGIT. |
| 11678 | 044214 | 013701 | 002114 | | | MOV STGAD,R1 | :SET R1 TO STARTING ADDRESS OF STRING |
| 11679 | | | | | | | : TO BE FORMED IN TEST BUFFER. |
| 11680 | 044220 | 005737 | 002112 | | | TST STGLN | :STRING TO BE FORMED HAVE ZERO LENGTH? |
| 11681 | 044224 | 001004 | | | | BNE 3\$ | :NO |
| 11682 | 044226 | 042737 | 177760 | 002124 | | BIC #177760,SIGN | :YES - INSERT SIGN. NOTE: A ZERO LENGTH |
| 11683 | 044234 | 000424 | | | | BR INSIGN | : PACKED STRING OCCUPIES 1 BYTE. |
| 11684 | 044236 | 013737 | 002112 | 002116 | 3\$: | MOV STGLN,SAVSL | :STRING TO BE FORMED HAVE EVEN LENGTH? |
| 11685 | 044244 | 042737 | 177776 | 002116 | | BIC #177776,SAVSL | |
| 11686 | 044252 | 001003 | | | | BNE 1\$ | |
| 11687 | 044254 | 113721 | 002110 | | | MOVB STGDS2,(R1)+ | :YES - INSERT 0000-DIGIT BYTE |
| 11688 | 044260 | 000404 | | | | BR 2\$ | |
| 11689 | 044262 | 113721 | 002106 | | 1\$: | MOVB STGDS1,(R1)+ | :INSERT NEXT PACKED DATA BYTE |
| 11690 | 044266 | 005337 | 002112 | | | DEC STGLN | |
| 11691 | 044272 | 005337 | 002112 | | 2\$: | DEC STGLN | |
| 11692 | 044276 | 005737 | 002112 | | | TST STGLN | :ENTIRE STRING BEEN INSERTED? |
| 11693 | 044302 | 003367 | | | | BGT 1\$ | :NO - CONTINUE INSERTING |
| 11694 | 044304 | 005301 | | | | DEC R1 | :YES - BACKUP |
| 11695 | 044306 | 022737 | 000007 | 002304 | INSIGN: | CMP #7,STGTYP | :IS STRING TYPE UNSIGNED? |
| 11696 | 044314 | 001003 | | | | BNE 1\$ | :BRANCH IF NO |
| 11697 | 044316 | 052737 | 000017 | 002124 | | BIS #17,SIGN | :UNSIGNED PACKED STRING SIGN MUST = (1111) |
| 11698 | 044324 | 113711 | 002124 | | 1\$: | MOVB SIGN,(R1) | :INSERT SIGN |
| 11699 | 044330 | 000137 | 045500 | | | JMP E1STG | |
| 11700 | | | | | | | |
| 11701 | 044334 | | | | DSTYP4: | | :DECIMAL STRING - USER DEFINED DIGIT STRING |
| 11702 | 044334 | 032737 | 000002 | 002454 | | BIT #2,ZPM | :IS FORMED STRING TO BE PACKED OR ZONED |
| 11703 | 044342 | 001002 | | | | BNE TYP4Z | :BRANCH IF ZONED |
| 11704 | 044344 | 000137 | 045054 | | | JMP TYP4P | |
| 11705 | | | | | | | |
| 11706 | 044350 | | | | TYP4Z: | | :ZONED |
| 11707 | 044350 | 013701 | 002112 | | | MOV STGLN,R1 | :GET STRING DESC. TYPE FIELD |
| 11708 | 044354 | 000301 | | | | SWAB R1 | |
| 11709 | 044356 | 006201 | | | | ASR R1 | |
| 11710 | 044360 | 006201 | | | | ASR R1 | |
| 11711 | 044362 | 006201 | | | | ASR R1 | |
| 11712 | 044364 | 006201 | | | | ASR R1 | |
| 11713 | 044366 | 042701 | 177770 | | | BIC #177770,R1 | |
| 11714 | 044372 | 010137 | 002304 | | | MOV R1,STGTYP | :SAVE TYPE |
| 11715 | 044376 | 042737 | 177740 | 002112 | | BIC #177740,STGLN | :STRIP TYPE OFF STRING LENGTH WORD |

| | | | | | | |
|-------|--------|--------|--------|--------|-------------------------|--|
| 11716 | 044404 | 042737 | 010000 | 002106 | BIC #10000,STGDS1 | :STRIP OFF ALL BUT LENGTH |
| 11717 | 044412 | 013701 | 002114 | | MOV STGAD,R1 | :INSERTION IS DONE FROM LS TO MS DIGIT |
| 11718 | 044416 | 063701 | 002112 | | ADD STGLN,R1 | :FORM IN R1 ADDRESS WHERE FIRST BYTE IS TO BE INSERTED |
| 11719 | 044422 | 010137 | 002432 | | MOV R1,ONEBEY | :SAVE PTR TO 1 BYTE BEYOND LS END OF STRING |
| 11720 | 044426 | 013737 | 002110 | 002120 | MOV STGDS2,SAVSA | :SAVE GIVEN STRING DESCRIPTOR WORDS |
| 11721 | 044434 | 013737 | 002106 | 002116 | MOV STGDS1,SAVSL | : |
| 11722 | 044442 | 063737 | 002106 | 002110 | ADD STGDS1,STGDS2 | :FORM IN STGDS2 ADDRESS WHERE FIRST BYTE IS TO BE |
| 11723 | | | | | | : TAKEN FROM |
| 11724 | 044450 | 005337 | 002110 | | DEC STGDS2 | |
| 11725 | 044454 | 005737 | 001770 | | TST RANDTA | :USE RANDOM SIGN? |
| 11726 | 044460 | 001403 | | | BEQ 4\$ | |
| 11727 | 044462 | 004737 | 044714 | | JSR PC,GETSGN | :YES |
| 11728 | 044466 | 000416 | | | BR 5\$ | |
| 11729 | 044470 | 117737 | 135414 | 002124 | 4\$: MOVB @STGDS2,SIGN | |
| 11730 | 044476 | 006237 | 002124 | | ASR SIGN | |
| 11731 | 044502 | 006237 | 002124 | | ASR SIGN | |
| 11732 | 044506 | 006237 | 002124 | | ASR SIGN | |
| 11733 | 044512 | 006237 | 002124 | | ASR SIGN | |
| 11734 | 044516 | 042737 | 177760 | 002124 | BIC #177760,SIGN | |
| 11735 | 044524 | 005737 | 002112 | | 5\$: TST STGLN | :STRING TO BE FORMED HAVE 0 LENGTH? |
| 11736 | 044530 | 001002 | | | BNE 1\$ | :BRANCH IF NO |
| 11737 | 044532 | 000137 | 044060 | | JMP TFS | |
| 11738 | 044536 | 005737 | 002112 | | 1\$: TST STGLN | :ENTIRE STRING BEEN INSERTED? |
| 11739 | 044542 | 001432 | | | BEQ WONSGN | :BRANCH IF YES |
| 11740 | 044544 | 005737 | 001770 | | TST RANDTA | :USE RANDOM DATA? |
| 11741 | 044550 | 001404 | | | BEQ 3\$ | :BRANCH IF NO |
| 11742 | 044552 | 004737 | 045414 | | JSR PC,GRZDB | :GENERATE IN R0 A RANDOM ZONED DATA BYTE |
| 11743 | 044556 | 110041 | | | MOVB R0,-(R1) | :INSERT BYTE |
| 11744 | 044560 | 000420 | | | BR 2\$ | |
| 11745 | 044562 | 117741 | 135322 | | 3\$: MOVB @STGDS2,-(R1) | :INSERT NEXT BYTE |
| 11746 | 044566 | 005337 | 002110 | | DEC STGDS2 | :UPDATE POINTERS |
| 11747 | 044572 | 005337 | 002106 | | DEC STGDS1 | |
| 11748 | 044576 | 001011 | | | BNE 2\$ | :GIVEN STRING EXHAUSTED? BRANCH IF NO |
| 11749 | 044600 | 013737 | 002116 | 002106 | MOV SAVSL,STGDS1 | :RESET POINTERS BACK TO BEGINNING OF GIVEN STRING |
| 11750 | 044606 | 013737 | 002120 | 002110 | MOV SAVSA,STGDS2 | |
| 11751 | 044614 | 063737 | 002106 | 002110 | ADD STGDS1,STGDS2 | |
| 11752 | 044622 | 005337 | 002112 | | 2\$: DEC STGLN | :DECREMENT COUNT OF # OF DIGITS TO BE INSERTED |
| 11753 | 044626 | 000743 | | | BR 1\$ | :RETURN |
| 11754 | 044630 | 013702 | 002304 | | WONSGN: MOV STGTYP,R2 | :WORK ON INSERTING SIGN BYTE |
| 11755 | 044634 | 006302 | | | ASL R2 | |
| 11756 | 044636 | 062702 | 002412 | | ADD #PTYPTA,R2 | |
| 11757 | 044642 | 000172 | 000000 | | JMP @(R2) | |
| 11758 | | | | | | |
| 11759 | 044646 | | | | PTYPTO: | :FIX UP POINTERS TO ENABLE USE OF |
| 11760 | | | | | | : TYP3Z ROUTINES. |
| 11761 | 044646 | 013701 | 002432 | | MOV ONEBEY,R1 | :GET LEAST SIGN DIGIT |
| 11762 | 044652 | 114101 | | | MOVB -(R1),R1 | |
| 11763 | 044654 | 042701 | 177760 | | PTO: BIC #177760,R1 | |
| 11764 | 044660 | 010137 | 002302 | | MOV R1,STGDIG | :SAVE IN STGDIG |
| 11765 | 044664 | 004737 | 044714 | | JSR PC,GETSGN | |
| 11766 | 044670 | 013701 | 002432 | | MOV ONEBEY,R1 | :SETUP R1 TO ONE BYTE BEYOND STRING |
| 11767 | 044674 | 023727 | 002304 | 000002 | CMP STGTYP,#2 | |
| 11768 | 044702 | 001402 | | | BEQ 1\$ | |
| 11769 | 044704 | 000137 | 043774 | | JMP TYPLO | |

| | | | | | | |
|-------|--------|--------|--------|---------|-------------------|--|
| 11770 | 044710 | 000137 | 043666 | 1\$: | JMP TYPTO | |
| 11771 | | | | | | |
| 11772 | 044714 | 005737 | 001770 | GETSGN: | TST RANDTA | ;USE RANDOM SIGN? |
| 11773 | 044720 | 001415 | | | BEQ 1\$ | ;BRANCH IF NO |
| 11774 | 044722 | 004737 | 063464 | | JSR PC,RN | ;GET A RANDOM # |
| 11775 | 044726 | 032700 | 000001 | | BIT #1,R0 | ;USE LEAST SIGN BIT TO PICK A SIGN |
| 11776 | 044732 | 001404 | | | BEQ 2\$ | |
| 11777 | 044734 | 012737 | 000007 | 002124 | MOV #7,SIGN | ;LS BIT = 1; MAKE SIGN NEG. |
| 11778 | 044742 | 000417 | | | BR 3\$ | |
| 11779 | 044744 | 012737 | 000003 | 002124 | 2\$: | MOV #3,SIGN |
| 11780 | 044752 | 000413 | | | BR 3\$ | ;LS BIT = 0;MAKE SIGN POS. |
| 11781 | 044754 | 013701 | 002432 | 1\$: | MOV ONEBEY,R1 | ;SUBROUTINE TO GET SIGN FROM INSERTED STRING |
| 11782 | 044760 | 114101 | | | MOVB -(R1),R1 | |
| 11783 | 044762 | 042701 | 177417 | | BIC #177417,R1 | |
| 11784 | 044766 | 006201 | | | ASR R1 | |
| 11785 | 044770 | 006201 | | | ASR R1 | ;RIGHT ADJUST |
| 11786 | 044772 | 006201 | | | ASR R1 | |
| 11787 | 044774 | 006201 | | | ASR R1 | |
| 11788 | 044776 | 010137 | 002124 | | MOV R1,SIGN | ;SAVE SIGN IN 'SIGN' |
| 11789 | 045002 | 000207 | | 3\$: | RTS PC | |
| 11790 | | | | | | |
| 11791 | 045004 | | | PTYPSZ: | | ;INSERT SIGN TYPE SIGN |
| 11792 | 045004 | 013701 | 002432 | | MOV ONEBEY,R1 | |
| 11793 | 045010 | 000137 | 043632 | | JMP TYPSZ | |
| 11794 | | | | | | |
| 11795 | 045014 | | | PTYPLO: | | ;INSERT LEADING OVERPUNCH SIGN |
| 11796 | 045014 | 117701 | 135074 | | MOVB @STGAD,R1 | ;GET MOST SIGN. DIGIT |
| 11797 | 045020 | 000137 | 044654 | | JMP PTO | |
| 11798 | | | | | | |
| 11799 | 045024 | | | PTYPTS: | | ;INSERT TRAILING SEPARATE SIGN |
| 11800 | 045024 | 004737 | 044714 | | JSR PC,GETSGN | ;GET SIGN TO INSERT |
| 11801 | 045030 | 013701 | 002432 | | MOV ONEBEY,R1 | |
| 11802 | 045034 | 000137 | 044004 | | JMP TYPTS | |
| 11803 | | | | | | |
| 11804 | 045040 | | | PTYPLS: | | ;INSERT LEADING SEPARATE SIGN |
| 11805 | 045040 | 004737 | 044714 | | JSR PC,GETSGN | |
| 11806 | 045044 | 013701 | 002432 | | MOV ONEBEY,R1 | |
| 11807 | 045050 | 000137 | 044050 | | JMP TYPLS | |
| 11808 | | | | | | |
| 11809 | 045054 | 013701 | 002112 | TYP4P: | MOV STGLN,R1 | ;GET STRING DESC. TYPE FIELD |
| 11810 | 045060 | 000301 | | | SWAB R1 | |
| 11811 | 045062 | 006201 | | | ASR R1 | |
| 11812 | 045064 | 006201 | | | ASR R1 | |
| 11813 | 045066 | 006201 | | | ASR R1 | |
| 11814 | 045070 | 006201 | | | ASR R1 | |
| 11815 | 045072 | 042701 | 177770 | | BIC #177770,R1 | |
| 11816 | 045076 | 010137 | 002304 | | MOV R1,STGTYP | ;SAVE TYPE |
| 11817 | 045102 | 042737 | 177740 | 002112 | BIC #177740,STGLN | ;STRIP TYPE OFF STRING LENGTH WORD |
| 11818 | 045110 | 013737 | 002112 | 002122 | MOV STGLN,SAVSGL | ;PACKED - SAVE STRING LENGTH |
| 11819 | 045116 | 006237 | 002112 | | ASR STGLN | ;INSERTION IS DONE FROM LS TO MS DIGIT |
| 11820 | 045122 | 013701 | 002112 | | MOV STGLN,R1 | ;FORM IN R1 ADDRESS WHERE 1ST BYTE IS TO BE INSERTED |
| 11821 | 045126 | 063701 | 002114 | | ADD STGAD,R1 | |
| 11822 | 045132 | 006237 | 002106 | | ASR STGDS1 | |
| 11823 | 045136 | 013737 | 002106 | 002116 | MOV STGDS1,SAVSL | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|---|
| 11824 | 045144 | 013737 | 002110 | 002120 | | MOV STGDS2,SAVSA | ;SAVE GIVEN STRING DESCRIPTOR WORDS |
| 11825 | 045152 | 063737 | 002106 | 002110 | | ADD STGDS1,STGDS2 | ;FORM IN STGDS2 ADDRESS WHERE 1ST BYTE |
| 11826 | 045160 | 005201 | | | | INC R1 | ;IS TO BE TAKEN FROM |
| 11827 | 045162 | 005737 | 002112 | | 1\$: | TST STGLN | ;ENTIRE STRING BEEN INSERTED? |
| 11828 | 045166 | 100432 | | | | BMI 3\$ | ;BRANCH IF YES |
| 11829 | 045170 | 005737 | 001770 | | | TST RANDTA | ;USE RANDOM DATA? |
| 11830 | 045174 | 001404 | | | | BEQ 5\$ | ;BRANCH IF NO |
| 11831 | 045176 | 004737 | 045460 | | | JSR PC,GRPDB | ;GENERATE IN R0 A RANDOM PACKED DATA BYTE |
| 11832 | 045202 | 110041 | | | | MOVB R0,-(R1) | ;INSERT BYTE |
| 11833 | 045204 | 000420 | | | | BR 2\$ | |
| 11834 | 045206 | 117741 | 134676 | | 5\$: | MOVB @STGDS2,-(R1) | ;INSERT NEXT BYTE |
| 11835 | 045212 | 005337 | 002110 | | | DEC STGDS2 | |
| 11836 | 045216 | 005337 | 002106 | | | DEC STGDS1 | ;UPDATE POINTERS |
| 11837 | 045222 | 002011 | | | | BGE 2\$ | ;GIVEN STRING EXHAUSTED? BRANCH IF NO |
| 11838 | 045224 | 013737 | 002116 | 002106 | | MOV SAVSL,STGDS1 | ;RESET POINTERS BACK TO BEGINNING OF GIVEN STRING |
| 11839 | 045232 | 013737 | 002120 | 002110 | | MOV SAVSA,STGDS2 | |
| 11840 | 045240 | 063737 | 002106 | 002110 | | ADD STGDS1,STGDS2 | |
| 11841 | 045246 | 005337 | 002112 | | 2\$: | DEC STGLN | ;DECREMENT COUNT OF # OF BYTES TO BE INSERTED |
| 11842 | 045252 | 000743 | | | | BR 1\$ | |
| 11843 | 045254 | 032737 | 000001 | 002122 | 3\$: | BIT #1,SAVSGL | ;IS STRING LENGTH ODD? |
| 11844 | 045262 | 001002 | | | | BNE 4\$ | ;BRANCH IF YES - DONE |
| 11845 | 045264 | 142721 | 000360 | | | BICB #360,(R1)+ | ;ZERO NIBBLE IN MOST SIGN BYTE |
| 11846 | 045270 | 013701 | 002122 | | 4\$: | MOV SAVSGL,R1 | ;CALCULATE SIGN ADDRESS |
| 11847 | 045274 | 006201 | | | | ASR R1 | |
| 11848 | 045276 | 063701 | 002114 | | | ADD STGAD,R1 | |
| 11849 | 045302 | 022737 | 000007 | 002304 | | CMP #7,STGTYP | ;IS STRING TYPE UNSIGNED? |
| 11850 | 045310 | 001431 | | | | BEQ 6\$ | ;BRANCH IF YES |
| 11851 | 045312 | 005737 | 001770 | | | TST RANDTA | ;USE RANDOM SIGN? |
| 11852 | 045316 | 001470 | | | | BEQ EISTG | ;BRANCH IF NO |
| 11853 | 045320 | 142711 | 000017 | | | BICB #17,(R1) | ;CLEAR OUT FOR SIGN |
| 11854 | 045324 | 004737 | 063464 | | | JSR PC,RN | |
| 11855 | 045330 | 032700 | 000001 | | | BIT #1,R0 | |
| 11856 | 045334 | 001403 | | | | BEQ 7\$ | |
| 11857 | 045336 | 152711 | 000013 | | | BISB #13,(R1) | ;MAKE SIGN NEGATIVE |
| 11858 | 045342 | 000456 | | | | BR EISTG | |
| 11859 | 045344 | 032700 | 000002 | | 7\$: | BIT #2,R0 | |
| 11860 | 045350 | 001403 | | | | BEQ 10\$ | |
| 11861 | 045352 | 152711 | 000016 | | | BISB #16,(R1) | ;MAKE SIGN + |
| 11862 | 045356 | 000450 | | | | BR EISTG | |
| 11863 | 045360 | 032700 | 000004 | | 10\$: | BIT #4,R0 | |
| 11864 | 045364 | 001403 | | | | BEQ 6\$ | |
| 11865 | 045366 | 152711 | 000012 | | | BISB #12,(R1) | ;MAKE SIGN + |
| 11866 | 045372 | 000442 | | | | BR EISTG | |
| 11867 | 045374 | 152711 | 000017 | | 6\$: | BISB #17,(R1) | ;OVERWRITE SIGN TO (1111) |
| 11868 | 045400 | 000437 | | | | BR EISTG | |
| 11869 | | | | | | | |
| 11870 | 045402 | | | | DSTYP5: | | ;DECIMAL STRING - RANDOM DATA & SIGN |
| 11871 | 045402 | 012737 | 177777 | 001770 | | MOV #177777,RANDTA | ;SET RANDOM DATA FLAG |
| 11872 | 045410 | 000137 | 044334 | | | JMP DSTYP4 | ;USE DSTYP4 ROUTINES. |
| 11873 | | | | | | | |
| 11874 | 045414 | | | | GRZDB: | | ;SUBROUTINE TO GENERATE A RANDOM ZONED |
| 11875 | | | | | | | ;DATA BYTE IN R0. |
| 11876 | 045414 | 004737 | 063464 | | | JSR PC,RN | ;GET A RANDOM #. |
| 11877 | 045420 | 042700 | 177740 | | | BIC #BS128,R0 | ;STRIP OFF ALL BUT LS NIBBLE |

| | | | | | | | |
|-------|--------|--------|--------|--------|--------------|--|---|
| 11878 | 045424 | 020027 | 000011 | | CMP R0,#11 | | :VALID NIBBLE = 0 TO 11. |
| 11879 | 045430 | 101402 | | | BLOS 1\$ | | :BRANCH IF NIBBLE IS VALID |
| 11880 | 045432 | 042700 | 000010 | | BIC #10,R0 | | :CONVERT INVALID NIBBLE TO A VALID ONE |
| 11881 | 045436 | 010037 | 001774 | 1\$: | MOV R0,RNIB | | :SAVE NIBBLE |
| 11882 | 045442 | 004737 | 063464 | | JSR PC,RN | | :GET ANOTHER RANDOM # |
| 11883 | 045446 | 043700 | 001766 | | BIC ZMSK,R0 | | :MASK OFF ALL BUT ZONE FIELD |
| 11884 | 045452 | 053700 | 001774 | | BIS RNIB,R0 | | :OR - IN THE DECIMAL NIBBLE |
| 11885 | 045456 | 000207 | | | RTS PC | | :RETURN WITH ZONED DATA BYTE IN R0 |
| 11886 | | | | | | | |
| 11887 | 045460 | | | GRPDB: | | | :SUBROUTINE TO GENERATE A RANDOM PACKED |
| 11888 | | | | | | | :DATA BYTE IN R0. |
| 11889 | 045460 | 004737 | 045414 | | JSR PC,GRZDB | | :GET A RANDOM ZONED DATA BYTE |
| 11890 | 045464 | 020027 | 000231 | | CMP R0,#231 | | :HIGH NIBBLE MUST BE <=9 |
| 11891 | 045470 | 101402 | | | BLOS 1\$ | | :BRANCH IF VALID |
| 11892 | 045472 | 042700 | 000200 | | BIC #200,R0 | | :CONVERT TO VALID |
| 11893 | 045476 | 000207 | | 1\$: | RTS PC | | |
| 11894 | 045500 | 000205 | | EISTG: | RTS R5 | | |
| 11895 | | | | | | | |

| | | | | | | | |
|-------|--------|--------|--------|--------|-----------|-----------------|--|
| 11897 | 045502 | | | | FC10: | | :FLOW COMMAND = 10 - COPY TEST BUFFER INTO |
| 11898 | | | | | | | : EMULATION BUFFER |
| 11899 | 045502 | 032737 | 000100 | 047464 | | BIT #100,TINST | :THE IN-LINE TEST CASE ALWAYS FOLLOWS |
| 11900 | | | | | | | : THE SAME REGISTER TEST CASE - THEREFORE |
| 11901 | | | | | | | : THEIR IS NO NEED TO REEMULATE. |
| 11902 | 045510 | 001010 | | | | BNE 2\$ | |
| 11903 | 045512 | 013701 | 001640 | | | MOV TBADR,R1 | :R1 POINTS TO START OF TEST BUFFER |
| 11904 | 045516 | 013702 | 001646 | | | MOV EBADR,R2 | :R2 POINTS TO START OF EMULATION BUFFER |
| 11905 | 045522 | 012122 | | | 1\$: | MOV (R1)+,(R2)+ | :COPY NEXT WORD |
| 11906 | 045524 | 023701 | 001754 | | | CMP TBEND,R1 | :COPY COMPLETE ? |
| 11907 | 045530 | 003374 | | | | BGT 1\$ | :NO |
| 11908 | 045532 | 000137 | 042010 | | 2\$: | JMP FCRTN | :YES |
| 11909 | | | | | | | |
| 11910 | 045536 | | | | FC11: | | :FLOW COMMAND = 11 - SETUP EMULATION |
| 11911 | | | | | | | : OPERANDS IN ERO - ER5 AND EMULATE |
| 11912 | | | | | | | : THE CIS INSTRUCTION UNDER TEST. |
| 11913 | 045536 | 032737 | 000100 | 047464 | | BIT #100,TINST | :THE IN-LINE TEST CASE ALWAYS FOLLOWS THE SAME |
| 11914 | | | | | | | : REGISTER TEST CASE - THEREFORE THEIR IS NO |
| 11915 | | | | | | | : NEED TO REEMULATE. |
| 11916 | 045544 | 001060 | | | | BNE FINEM | |
| 11917 | 045546 | 013737 | 003624 | 003664 | | MOV TR0,ERO | :COPY ERN DIRECTLY FROM TRN |
| 11918 | 045554 | 013737 | 003626 | 003666 | | MOV TR1,ER1 | |
| 11919 | 045562 | 013737 | 003630 | 003670 | | MOV TR2,ER2 | |
| 11920 | 045570 | 013737 | 003632 | 003672 | | MOV TR3,ER3 | |
| 11921 | 045576 | 013737 | 003634 | 003674 | | MOV TR4,ER4 | |
| 11922 | 045604 | 013737 | 003636 | 003676 | | MOV TR5,ER5 | |
| 11923 | 045612 | 013737 | 003640 | 003700 | | MOV TR6,ER6 | |
| 11924 | | | | | | | |
| 11925 | | | | | | | :ADJUST ERN SPECIFIED IN FLOW TABLE ENTRY |
| 11926 | | | | | | | : TO ACCOUNT FOR EMULATION VERSUS TEST |
| 11927 | | | | | | | : BUFFER STARTING ADDRESS |
| 11928 | 045620 | 004737 | 053340 | | | JSR PC,RF1 | :FORM POINTER TO FIRST TEST OPERAND TO |
| 11929 | | | | | | | : BE ADJUSTED. |
| 11930 | 045624 | 020127 | 003624 | | | CMP R1,#TRN | :ANY OPERANDS TO BE ADJUSTED? |
| 11931 | 045630 | 001420 | | | | BEQ EC11 | :BRANCH IF NO. |
| 11932 | 045632 | 004737 | 045712 | | | JSR PC,ADJEOP | :ADJUST OPERAND |
| 11933 | 045636 | 004737 | 053372 | | | JSR PC,RF2 | :FORM POINTER TO SECOND TEST OPERAND TO |
| 11934 | | | | | | | : BE ADJUSTED. |
| 11935 | 045642 | 020127 | 003624 | | | CMP R1,#TRN | :IF R1 STILL POINTS TO #TRN THEN THERE |
| 11936 | | | | | | | : WAS ONLY ONE OPERAND TO BE ADJUSTED. |
| 11937 | 045646 | 001411 | | | | BEQ EC11 | :ADJUSTING COMPLETE |
| 11938 | 045650 | 004737 | 045712 | | | JSR PC,ADJEOP | :ADJUST NEXT EMULATION OPERAND |
| 11939 | 045654 | 004737 | 053426 | | | JSR PC,RF3 | :FORM POINTER TO THIRD TEST OPERAND TO |
| 11940 | | | | | | | : BE ADJUSTED. |
| 11941 | 045660 | 020127 | 003624 | | | CMP R1,#TRN | :WAS THERE A THIRD OPERAND TO BE ADJUSTED? |
| 11942 | 045664 | 001402 | | | | BEQ EC11 | :NO - ADJUSTING COMPLETE |
| 11943 | 045666 | 004737 | 045712 | | | JSR PC,ADJEOP | :YES - ADJUST IT. |
| 11944 | | | | | | | |
| 11945 | 045672 | | | | EC11: | | :EMULATION OPERANDS ALL SET. |
| 11946 | | | | | | | :EMULATE CIS INST. |
| 11947 | 045672 | 004737 | 016700 | | | JSR PC,EMULATE | :CALL EMULATOR |
| 11948 | 045676 | 000000 | | | EINSTR: 0 | | :INSTRUCTION |
| 11949 | 045700 | 003664 | | | | ERN | :POINTER TO REGISTER OPERANDS |
| 11950 | 045702 | 003702 | | | | ERNR | :POINTER TO REGISTER RESULTS |

| | | | | | | | | | | | |
|-------|--------|--------|--------|--------|--|--------------------------|--|--|--|--|---|
| 11951 | 045704 | 003720 | | | | ECCR | | | | | : POINTER TO COND. CODE RESULTS |
| 11952 | 045706 | 000137 | 042010 | | | FINEM: JMP FCRTN | | | | | |
| 11953 | | | | | | | | | | | |
| 11954 | 045712 | | | | | ADJEOP: | | | | | : SUBROUTINE TO UPDATE EMULATION OPERANDS : TO REFLECT EMULATION BUFFER STARTING ADDRESS. |
| 11955 | | | | | | | | | | | |
| 11956 | 045712 | 162701 | 003624 | | | SUB #TRN,R1 | | | | | |
| 11957 | 045716 | 062701 | 003664 | | | ADD #ERN,R1 | | | | | : POINT R1 TO EMULATION OPERAND RATHER THAN : TEST OPERAND. |
| 11958 | | | | | | | | | | | |
| 11959 | 045722 | 163711 | 001640 | | | SUB TBADR,(R1) | | | | | : SUBTRACT OUT TEST BUFFER STARTING ADDRESS |
| 11960 | 045726 | 063711 | 001646 | | | ADD EBADR,(R1) | | | | | : ADD IN EMULATION BUFFER STARTING ADDRESS |
| 11961 | 045732 | 000207 | | | | RTS PC | | | | | |
| 11962 | | | | | | | | | | | |
| 11963 | 045734 | | | | | FC12: | | | | | : FLOW COMMAND = 12 - SETUP REGISTERS (OR INLINE POINTERS : AND COND CODES AND EXECUTE CIS INSTRUCTION. : SAVE RESULTS. : SAVE TRNS IN TEMPORARY STORAGE |
| 11964 | | | | | | | | | | | |
| 11965 | | | | | | | | | | | |
| 11966 | 045734 | 013737 | 003624 | 002230 | | MOV TRO,TTR0 | | | | | |
| 11967 | 045742 | 013737 | 003626 | 002232 | | MOV TR1,TTR1 | | | | | |
| 11968 | 045750 | 013737 | 003630 | 002234 | | MOV TR2,TTR2 | | | | | |
| 11969 | 045756 | 013737 | 003632 | 002236 | | MOV TR3,TTR3 | | | | | |
| 11970 | 045764 | 013737 | 003634 | 002240 | | MOV TR4,TTR4 | | | | | |
| 11971 | 045772 | 013737 | 003636 | 002242 | | MOV TR5,TTR5 | | | | | |
| 11972 | 046000 | 032737 | 000100 | 047464 | | BIT #100,TINST | | | | | : INST TYPE (REG OR IN-LINE)? |
| 11973 | 046006 | 001014 | | | | BNE INLINE | | | | | : BRANCH IF INLINE TYPE |
| 11974 | 046010 | 023727 | 047464 | 076027 | | CMP TINST,#76027 | | | | | |
| 11975 | 046016 | 001406 | | | | BEQ NONOP | | | | | |
| 11976 | 046020 | 023727 | 047464 | 076067 | | CMP TINST,#76067 | | | | | |
| 11977 | 046026 | 001402 | | | | BEQ NONOP | | | | | |
| 11978 | 046030 | 000137 | 046664 | | | JMP REGOP | | | | | |
| 11979 | 046034 | 000137 | 046730 | | | NONOP: JMP LTSTOP | | | | | |
| 11980 | 046040 | 004737 | 053352 | | | INLINE: JSR PC,RF1X | | | | | : GET # OF IN-LINE DESCRIPTOR POINTERS |
| 11981 | 046044 | 006201 | | | | ASR R1 | | | | | |
| 11982 | 046046 | 012702 | 000005 | | | MOV #5,R2 | | | | | : CALCULATE BRANCH TO INSERT IMMEDIATELY : FOLLOWING IN LINE DESCRIPTORS. |
| 11983 | | | | | | | | | | | |
| 11984 | 046052 | 160102 | | | | SUB R1,R2 | | | | | |
| 11985 | 046054 | 062702 | 000400 | | | ADD #400,R2 | | | | | |
| 11986 | 046060 | 010237 | 001756 | | | MOV R2,SBR | | | | | : SAVE BRANCH INST |
| 11987 | 046064 | 005002 | | | | CLR R2 | | | | | |
| 11988 | 046066 | 013703 | 001640 | | | MOV TBADR,R3 | | | | | |
| 11989 | 046072 | 005701 | | | | 2\$: TST R1 | | | | | |
| 11990 | 046074 | 001410 | | | | BEQ 1\$ | | | | | : BRANCH IF ALL IN-LINE PTRS HAVE BEEN INSERTED. |
| 11991 | 046076 | 010362 | 047466 | | | MOV R3,TINST+2(R2) | | | | | : INSERT DESCRIPTOR POINTERS IN-LINE |
| 11992 | 046102 | 005301 | | | | DEC R1 | | | | | : UPDATE FOR NEXT IN-LINE PTR |
| 11993 | 046104 | 062702 | 000002 | | | ADD #2,R2 | | | | | |
| 11994 | 046110 | 062703 | 000004 | | | ADD #4,R3 | | | | | |
| 11995 | 046114 | 000766 | | | | BR 2\$ | | | | | : RETURN TO WORK ON NEXT IN-LINE PTR. |
| 11996 | 046116 | 012737 | 000001 | 002226 | | 1\$: MOV #1,NXFLD | | | | | : INITIALIZE FIELD PTR TO SECOND FIELD |
| 11997 | 046124 | 004737 | 053604 | | | 11\$: JSR PC,RFN | | | | | : GET NEXT FIELD CONTENTS. |
| 11998 | 046130 | 020127 | 003624 | | | CMP R1,#TRN | | | | | : ALL OPERANDS INSERTED IN-LINE? |
| 11999 | 046134 | 001407 | | | | BEQ 3\$ | | | | | : BRANCH IF YES |
| 12000 | 046136 | 005337 | 001756 | | | DEC SBR | | | | | |
| 12001 | 046142 | 011162 | 047466 | | | MOV (R1),TINST+2(R2) | | | | | : MOVE REGISTER OPERAND INTO IN-LINE LOC |
| 12002 | 046146 | 062702 | 000002 | | | ADD #2,R2 | | | | | : UPDATE TO NEXT IN-LINE LOC |
| 12003 | 046152 | 000764 | | | | BR 11\$ | | | | | : RETURN TO WORK ON NEXT IN-LINE OPERAND |
| 12004 | 046154 | 013762 | 001756 | 047466 | | 3\$: MOV SBR,TINST+2(R2) | | | | | : INSERT BRANCH |

| | | | | | | | |
|-------|--------|--------|--------|--------|-------|-----------------------|--|
| 12005 | 046162 | 012737 | 047466 | 002214 | | MOV #TINST+2,ICOMPC | :SAVE BRANCH LOCATION FOR PC |
| 12006 | 046170 | 060237 | 002214 | | | ADD R2,ICOMPC | : CHECK ON IN-LINE INSTRUCTION COMPLETION |
| 12007 | 046174 | 062702 | 000002 | | 12\$: | ADD #2,R2 | :INSERT HALTS FOLLOWING BRANCH |
| 12008 | 046200 | 022702 | 000014 | | | CMP #14,R2 | |
| 12009 | 046204 | 001404 | | | | BEQ 4\$ | |
| 12010 | 046206 | 013762 | 001674 | 047466 | | MOV KHALT,TINST+2(R2) | |
| 12011 | 046214 | 000767 | | | | BR 12\$ | |
| 12012 | | | | | | | |
| 12013 | 046216 | 062737 | 000002 | 001742 | 4\$: | ADD #2,FLOPTR | :LOOK AT NEXT FLOW COMMAND (=00) |
| 12014 | 046224 | 013702 | 001640 | | | MOV TBADR,R2 | :INITIALIZE POINTERS TO TEST & EMULATION BUFFERS |
| 12015 | 046230 | 013703 | 001646 | | | MOV EBADR,R3 | |
| 12016 | 046234 | 005037 | 002226 | | | CLR NXFLD | :INITIALIZE FIELD PTR TO 1ST FIELD |
| 12017 | 046240 | 004737 | 053604 | | 41\$: | JSR PC,RFN | :GET NEXT FIELD CONTENTS |
| 12018 | 046244 | 020127 | 003624 | | | CMP R1,#TRN | :ALL BUFFER DESCRIPTORS FILLED? |
| 12019 | 046250 | 001441 | | | | BEQ 5\$ | :BRANCH IF YES |
| 12020 | 046252 | 020127 | 003632 | | | CMP R1,#TR3 | :FOR CVTLN & CVTLP THE MEMORY ORDER |
| 12021 | 046256 | 001021 | | | | BNE 42\$ | : OF THE LONG INTEGER LOW & HIGH |
| 12022 | 046260 | 023727 | 047464 | 076157 | | CMP TINST,#76157 | : IS REVERSED FROM THE REGISTER ORDER |
| 12023 | 046266 | 001404 | | | | BEQ 43\$ | |
| 12024 | 046270 | 023727 | 047464 | 076177 | | CMP TINST,#76177 | |
| 12025 | 046276 | 001011 | | | | BNE 42\$ | |
| 12026 | 046300 | 011112 | | | 43\$: | MOV (R1),(R2) | |
| 12027 | 046302 | 011113 | | | | MOV (R1),(R3) | |
| 12028 | 046304 | 162701 | 000002 | | | SUB #2,R1 | |
| 12029 | 046310 | 011162 | 000002 | | | MOV (R1),2(R2) | |
| 12030 | 046314 | 011163 | 000002 | | | MOV (R1),2(R3) | |
| 12031 | 046320 | 000410 | | | | BR 44\$ | |
| 12032 | 046322 | 011162 | 000002 | | 42\$: | MOV (R1),2(R2) | :COPY BUFFER DESCRIPTORS FROM REGISTER |
| 12033 | 046326 | 011163 | 000002 | | | MOV (R1),2(R3) | : DESCRIPTORS. |
| 12034 | 046332 | 162701 | 000002 | | | SUB #2,R1 | |
| 12035 | 046336 | 011112 | | | | MOV (R1),(R2) | |
| 12036 | 046340 | 011113 | | | | MOV (R1),(R3) | |
| 12037 | 046342 | 062702 | 000004 | | 44\$: | ADD #4,R2 | :UPDATE BUFFER DESCRIPTOR POINTERS |
| 12038 | 046346 | 062703 | 000004 | | | ADD #4,R3 | |
| 12039 | 046352 | 000732 | | | | BR 41\$ | |
| 12040 | 046354 | 062737 | 000002 | 001742 | 5\$: | ADD #2,FLOPTR | :RETURN TO FILL NEXT DESCRIPTOR WORDS. |
| 12041 | 046362 | 005737 | 002266 | | | TST RPTFLG | :LOOK AT NEXT FLOW COMMAND (=00) |
| 12042 | 046366 | 001025 | | | | BNE 6\$ | :IS TEST BEING REPEATED?? |
| 12043 | | | | | | | :BRANCH IF YES (DONT UPDATE EMUL BUFFER |
| 12044 | 046370 | 013703 | 001646 | | | MOV EBADR,R3 | : DESCRIPTORS - THEY ARE ALREADY UP-TO-DATE). |
| 12045 | 046374 | 005037 | 002226 | | | CLR NXFLD | :INITIALIZE POINTER TO EMULATION BUFFER. |
| 12046 | 046400 | 004737 | 053504 | | 51\$: | JSR PC,RFN | :INITIALIZE FIELD PTR TO 1ST FIELD |
| 12047 | 046404 | 005701 | | | | TST R1 | :GET NEXT FIELD CONTENTS |
| 12048 | 046406 | 001415 | | | | BEQ 6\$ | :ALL RESULT EMULATOR BUFFER DESCRIPTORS UPDATED? |
| 12049 | 046410 | 020127 | 000016 | | | CMP R1,#16 | :BRANCH IF YES |
| 12050 | 046414 | 001407 | | | | BEQ 52\$ | :DESCRIPTOR TO BE UPDATED? |
| 12051 | 046416 | 062701 | 003702 | | | ADD #ERNR,R1 | :BRANCH IF NO |
| 12052 | 046422 | 011113 | | | | MOV (R1),(R3) | :FORM POINTER INTO EMUL. RESULT STACK |
| 12053 | 046424 | 162701 | 000002 | | | SUB #2,R1 | :COPY EMUL. RESULT INTO BUFFER |
| 12054 | 046430 | 011163 | 000002 | | | MOV (R1),2(R3) | |
| 12055 | 046434 | 062703 | 000004 | | 52\$: | ADD #4,R3 | :UPDATE EMUL. BUFFER POINTER |
| 12056 | 046440 | 000757 | | | | BR 51\$ | :RETURN TO WORK ON NEXT EMUL. RESULT. |
| 12057 | | | | | | | |
| 12058 | | | | | | | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|--|
| 12059 | 046442 | 013737 | 001720 | 003624 | 6\$: | MOV PAT0,TR0 | ;INITIALIZE TRX'S TO #PATTERN |
| 12060 | 046450 | 013737 | 001722 | 003626 | | MOV PAT1,TR1 | |
| 12061 | 046456 | 013737 | 001724 | 003630 | | MOV PAT2,TR2 | |
| 12062 | 046464 | 013737 | 001726 | 003632 | | MOV PAT3,TR3 | |
| 12063 | 046472 | 013737 | 001730 | 003634 | | MOV PAT4,TR4 | |
| 12064 | 046500 | 013737 | 001732 | 003636 | | MOV PAT5,TR5 | |
| 12065 | | | | | | | |
| 12066 | 046506 | 062737 | 000002 | 001742 | 7\$: | ADD #2,FLOPTR | ;OVERWRITE WITH #PATTERN ALL ERNRS EXCEPT |
| 12067 | | | | | | | ; THOSE THAT CONTAIN A REGISTER RESULT. |
| 12068 | | | | | | | ;LOOK AT NEXT FLOW COMMAND (=00) |
| 12069 | 046514 | 013737 | 003702 | 002246 | | MOV ER0R,TER0R | ;COPY ERNRS INTO TEMP SPACE AT TERNR |
| 12070 | 046522 | 013737 | 003704 | 002250 | | MOV ER1R,TER1R | |
| 12071 | 046530 | 013737 | 003706 | 002252 | | MOV ER2R,TER2R | |
| 12072 | 046536 | 013737 | 003710 | 002254 | | MOV ER3R,TER3R | |
| 12073 | 046544 | 013737 | 003712 | 002256 | | MOV ER4R,TER4R | |
| 12074 | 046552 | 013737 | 003714 | 002260 | | MOV ER5R,TER5R | |
| 12075 | | | | | | | |
| 12076 | 046560 | 013737 | 001720 | 003702 | 8\$: | MOV PAT0,ER0R | ;OVERWRITE ERNR STACK WITH # PATTERN |
| 12077 | 046566 | 013737 | 001722 | 003704 | | MOV PAT1,ER1R | |
| 12078 | 046574 | 013737 | 001724 | 003706 | | MOV PAT2,ER2R | |
| 12079 | 046602 | 013737 | 001726 | 003710 | | MOV PAT3,ER3R | |
| 12080 | 046610 | 013737 | 001730 | 003712 | | MOV PAT4,ER4R | |
| 12081 | 046616 | 013737 | 001732 | 003714 | | MOV PAT5,ER5R | |
| 12082 | | | | | | | |
| 12083 | 046624 | 005037 | 002226 | | 9\$: | CLR NXFLD | ;INITIALIZE FIELD PTR TO 1ST FIELD |
| 12084 | 046630 | 004737 | 053504 | | 91\$: | JSR PC,RFNX | ;GET NEXT FIELD CONTENTS |
| 12085 | 046634 | 020127 | 000016 | | | CMP R1,#16 | ;END OF REGISTER RESULT LIST? |
| 12086 | 046640 | 001407 | | | | BEQ 10\$ | ;BRANCH IF YES |
| 12087 | 046642 | 010102 | | | | MOV R1,R2 | |
| 12088 | 046644 | 062701 | 003702 | | | ADD #ERNR,R1 | |
| 12089 | 046650 | 062702 | 002246 | | | ADD #TER0R,R2 | |
| 12090 | 046654 | 011211 | | | | MOV (R2),(R1) | ;COPY REGISTER RESULT BACK INTO ERNR STACK |
| 12091 | | | | | | | ; FOR TEMPORARY STORAGE. |
| 12092 | 046656 | 000764 | | | | BR 91\$ | ;RETURN TO WORK ON NEXT REGISTER RESULT. |
| 12093 | 046660 | 000137 | 046730 | | 10\$: | JMP LTSTOP | |
| 12094 | | | | | | | |
| 12095 | 046664 | 013737 | 001676 | 047466 | REGOP: | MOV KBR5,TINST+2 | |
| 12096 | 046672 | 013737 | 001674 | 047470 | | MOV KHALT,TINST+4 | |
| 12097 | 046700 | 013737 | 001674 | 047472 | | MOV KHALT,TINST+6 | |
| 12098 | 046706 | 013737 | 001674 | 047474 | | MOV KHALT,TINST+10 | |
| 12099 | 046714 | 013737 | 001674 | 047476 | | MOV KHALT,TINST+12 | |
| 12100 | 046722 | 013737 | 001674 | 047500 | | MOV KHALT,TINST+14 | |
| 12101 | | | | | | | |
| 12102 | 046730 | 010637 | 002224 | | LTSTOP: | MOV SP,TSP | ;SAVE STACK POINTER |
| 12103 | 046734 | 022737 | 076026 | 047464 | | CMP #076026,TINST | ;IS INST A L2D6? |
| 12104 | 046742 | 001003 | | | | BNE 2\$ | ;BRANCH IF NO |
| 12105 | 046744 | 013706 | 003640 | | 3\$: | MOV TR6,SP | ;LOAD DESC 6 INST - USE TR6 CONTENTS AS SP |
| 12106 | 046750 | 000411 | | | | BR 1\$ | |
| 12107 | 046752 | 022737 | 076066 | 047464 | 2\$: | CMP #076066,TINST | ;IS INST A L3D6? |
| 12108 | 046760 | 001771 | | | | BEQ 3\$ | ;BRANCH IF YES |
| 12109 | 046762 | 012737 | 120602 | 003716 | | MOV #CSTACK,ER6R | |
| 12110 | 046770 | 012706 | 120602 | | | MOV #CSTACK,SP | |
| 12111 | 046774 | 010637 | 003640 | | 1\$: | MOV SP,TR6 | ;LOAD TR6 FOR INPUT REG DISPLAY |
| 12112 | 047000 | 013737 | 003720 | 003642 | | MOV ECCR,TCC | ;SETUP CONDITION CODES |

| | | | | | | | |
|-------|--------|--------|--------|--------|--------------------|------|--|
| 12113 | 047006 | 013701 | 047464 | | MOV TINST,R1 | | :IS INST A L2D OR L3D? |
| 12114 | 047012 | 042701 | 000007 | | BIC #7,R1 | | |
| 12115 | 047016 | 022701 | 076020 | | CMP #076020,R1 | | :IF YES SETUP CONDITION CODES=EXPECTED CODES. |
| 12116 | 047022 | 001405 | | | BEQ 4\$ | | |
| 12117 | 047024 | 022701 | 076060 | | CMP #076060,R1 | | |
| 12118 | 047030 | 001402 | | | BEQ 4\$ | | |
| 12119 | 047032 | 005137 | 003642 | | COM TCC | | :OTHERWISE SETUP CC=COMPLEMENT OF EXPECTED CC. |
| 12120 | 047036 | 004737 | 054332 | 4\$: | JSR PC,SELREG | | :SELECT REGISTER SET AND SETUP CONTENTS OF REGISTER SET NOT SELECTED. |
| 12121 | | | | | | | |
| 12122 | 047042 | 012737 | 000240 | 047502 | MOV #NOP,TINRET | | :RESTORE NOP TO CIS INST RETURN POINT |
| 12123 | 047050 | 004737 | 055520 | | JSR PC,SELMD | | :SELECT MODE & I/D ENABLES; LOAD MMR3 TO REFLECT D-SPACE ENABLE/DIS SELECTION |
| 12124 | | | | | | | |
| 12125 | 047054 | 005737 | 002160 | | TST MODE | | :TEST MODE = KERNEL? |
| 12126 | 047060 | 001403 | | | BEQ 17\$ | | :BRANCH IF YES |
| 12127 | 047062 | 012737 | 000000 | 047502 | MOV #HALT,TINRET | | :LOAD HALT AT INST UNDER TEST RETURN ADDRESS (ALLOWS TRAPPING BACK TO KERNEL MODE AFTER CIS INST EXECUTION). |
| 12128 | | | | | | | |
| 12129 | | | | | | | |
| 12130 | 047070 | 005737 | 002156 | 17\$: | TST MMFLG | | :IS MEMORY MGMT AVAILABLE? |
| 12131 | 047074 | 001411 | | | BEQ 8\$ | | :BRANCH IF NO |
| 12132 | 047076 | 004737 | 055744 | | JSR PC,SETPDR | | :SETUP MEM MGMT PDR'S |
| 12133 | | | | | | | :NOTE: PAR'S ARE SETUP AT BEGINNING OF PROG |
| 12134 | | | | | | | :KERNEL,USER & SUPV SPACES ARE ALWAYS MAPPED TO SAME PHYSICAL MEMORY) |
| 12135 | | | | | | | :ALLOW MEMORY MANAGEMENT TURN ON |
| 12136 | 047102 | 012737 | 000240 | 047446 | MOV #NOP,TOMM | | :ALLOW MEMORY MANAGEMENT TURN OFF |
| 12137 | 047110 | 012737 | 000240 | 047512 | MOV #NOP,TOFMM | | |
| 12138 | 047116 | 000406 | | | BR 9\$ | | |
| 12139 | 047120 | 012737 | 000403 | 047446 | MOV #403,TOMM | 8\$: | :INHIBIT MEM MGMT TURN ON |
| 12140 | 047126 | 012737 | 000402 | 047512 | MOV #402,TOFMM | | |
| 12141 | 047134 | 013700 | 003624 | | MOV TR0,R0 | 9\$: | :LOAD TEST OPERANDS INTO REGISTERS |
| 12142 | 047140 | 010037 | 002562 | | MOV R0,STATR0 | | :SAVE STATE OF CIS |
| 12143 | 047144 | 013701 | 003626 | | MOV TR1,R1 | | |
| 12144 | 047150 | 010137 | 002564 | | MOV R1,STATR1 | | |
| 12145 | 047154 | 013702 | 003630 | | MOV TR2,R2 | | |
| 12146 | 047160 | 010237 | 002566 | | MOV R2,STATR2 | | |
| 12147 | 047164 | 013703 | 003632 | | MOV TR3,R3 | | |
| 12148 | 047170 | 010337 | 002570 | | MOV R3,STATR3 | | |
| 12149 | 047174 | 013704 | 003634 | | MOV TR4,R4 | | |
| 12150 | 047200 | 010437 | 002572 | | MOV R4,STATR4 | | |
| 12151 | 047204 | 013705 | 003636 | | MOV TR5,R5 | | |
| 12152 | 047210 | 010537 | 002574 | | MOV R5,STATR5 | | |
| 12153 | 047214 | 010637 | 002576 | | MOV SP,STATR6 | | |
| 12154 | 047220 | 005737 | 002160 | | TST MODE | | :IS MODE = KERNEL? |
| 12155 | 047224 | 001003 | | | BNE 5\$ | | :BRANCH IF NO |
| 12156 | 047226 | 162737 | 000006 | 002576 | SUB #6,STATR6 | | :ADJUST SAVED STACK POINTER TO ACCOUNT FOR INTERRUPT STACK PUSH (KERNEL = CIS INST STACK) |
| 12157 | | | | | | | :FORM PSW TO BE USED UPON ENTRY TO CIS INST |
| 12158 | 047234 | 042737 | 177760 | 003642 | BIC #177760,TCC | 5\$: | |
| 12159 | 047242 | 017737 | 132416 | 001752 | MOV @TPSW,TSTPSW | | |
| 12160 | 047250 | 042737 | 170017 | 001752 | BIC #170017,TSTPSW | | |
| 12161 | 047256 | 053737 | 003642 | 001752 | BIS TCC,TSTPSW | | |
| 12162 | 047264 | 022737 | 000001 | 002160 | CMP #1,MODE | | :TEST MODE = SUPERVISOR? |
| 12163 | 047272 | 001004 | | | BNE 6\$ | | :BRANCH IF NO |
| 12164 | 047274 | 052737 | 040000 | 001752 | BIS #040000,TSTPSW | | :SET CURRENT MODE = SUPV IN TSTPSW |
| 12165 | 047302 | 000407 | | | BR 7\$ | | |
| 12166 | 047304 | 022737 | 000003 | 002160 | CMP #3,MODE | 6\$: | :TEST MODE = USER? |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|--|
| 12167 | 047312 | 001015 | | | | BNE 10\$ | :BRANCH IF NO |
| 12168 | 047314 | 052737 | 140000 | 001752 | | BIS #140000,@TSPSW | :SET CURR MODE = USER IN TSTPSW |
| 12169 | 047322 | 013777 | 001752 | 132334 | 7\$: | MOV TSTPSW,@TSPSW | :MODE = USER + SUPV; SWITCH TO TEST MODE |
| 12170 | 047330 | 013706 | 002576 | | | MOV STATR6,SP | :SETUP TEST MODE R6 |
| 12171 | 047334 | 042777 | 140000 | 132322 | | BIC #140000,@TSPSW | :SWITCH BACK TO KERNEL |
| 12172 | 047342 | 013706 | 002224 | | | MOV TSP,SP | :SETUP KERNEL MODE R6 |
| 12173 | 047346 | 013737 | 001752 | 002560 | 10\$: | MOV TSTPSW,STATPS | |
| 12174 | 047354 | 005037 | 002542 | | | CLR INTCT | :CLEAR INTERRUPT COUNT |
| 12175 | 047361 | 042737 | 040000 | 002132 | | BIC #40000,FATAL | :CLEAR INTERRUPT INDICATOR IN FATAL ERROR WORD |
| 12176 | 047366 | 005037 | 002534 | | | CLR PROGCT | :CLEAR PROGRESS COUNT |
| 12177 | 047372 | 000414 | | | TOLTC: | BR TOPC2 | :OVERWRITTEN WITH A NOP IF LTC IS USED FOR : INTERRUPT TESTING |
| 12178 | | | | | | | |
| 12179 | 047374 | 013737 | 003024 | 003026 | | MOV LCNT,VLCNT | :RESET LTC COUNTER |
| 12180 | 047402 | 004737 | 062234 | | | JSR PC,LTC SUP | :SYNC UP TO LTC |
| 12181 | 047406 | 052777 | 000100 | 133406 | | BIS #100,@LKS | :ENABLE LTC INTR |
| 12182 | 047414 | 005337 | 003026 | | 1\$: | DEC VLCNT | :KILL MOST OF TIME BEFORE EXPECTED : INTERRUPT |
| 12183 | | | | | | | |
| 12184 | 047420 | 001375 | | | | BNE 1\$ | |
| 12185 | 047422 | 000410 | | | | BR PREINS | |
| 12186 | 047424 | 000403 | | | TOPC2: | BR TOPC1 | :OVERWRITTEN WITH A NOP IF LATENCY IS BEING TESTED |
| 12187 | 047426 | 052777 | 000001 | 133070 | | BIS #001,@PC2CSR | :TURN ON P-CLK2 |
| 12188 | 047434 | 000403 | | | TOPC1: | BR PREINS | :OVERWRITTEN WITH A NOP IF INTERRUPTABILITY IS B |
| 12189 | 047436 | 052777 | 000101 | 133050 | | BIS #101,@PC1CSR | :TURN ON P-CLK1 |
| 12190 | 047444 | 000240 | | | PREINS: | NOP | :REPLACED WITH A HALT IF OPERATOR REQUESTED |
| 12191 | | | | | | | : 'HALT AT CIS INST' |
| 12192 | 047446 | 000403 | | | TOMM: | BR GO | :OVERWRITTEN WITH A NOP IF MEM MGMT TEST STATE = ON |
| 12193 | 047450 | 052737 | 000001 | 177572 | | BIS #1,@MMR0 | :TURN ON MEMORY MGMT |
| 12194 | 047456 | 013777 | 001752 | 132200 | GO: | MOV TSTPSW,@TSPSW | :SET PSW TO DESIRED STATE (PRIOR TO THIS MODE=KERNEL) |
| 12195 | 047464 | 000000 | | | TINST: | .WORD 0 | :EXECUTE CIS INST UNDER TEST. |
| 12196 | 047466 | 000405 | | | | BR TINRET | |
| 12197 | 047470 | 000000 | | | | HALT | |
| 12198 | 047472 | 000000 | | | | HALT | :IF PROGRAM STOPS AT ANY ONE OF THESE HALTS, : THEN THE CIS INSTRUCTION EXECUTION |
| 12199 | 047474 | 000000 | | | | HALT | : RETURNED WITH INCORRECT PC. |
| 12200 | 047476 | 000000 | | | | HALT | |
| 12201 | 047500 | 000000 | | | | HALT | |
| 12202 | 047502 | 000240 | | | TINRET: | NOP | :OVERWRITTEN WITH A HALT IF MODE = USER OR SUPV |
| 12203 | | | | | | | :NOTE: HALT IS USED TO TRAP BACK TO KERNEL MODE |
| 12204 | 047504 | 017737 | 132154 | 120604 | | MOV @TSPSW,SAVKCC | :SAVE CC RESULTS |
| 12205 | 047512 | 000402 | | | TOFMM: | BR CSCC | :OVERWRITTEN WITH A NOP IF MEM MGMT IS AVAILABLE |
| 12206 | 047514 | 005037 | 177572 | | | CLR @MMR0 | :TURN OFF MEMORY MGMT |
| 12207 | 047520 | 013737 | 120604 | 003662 | CSCC: | MOV SAVKCC,TCCR | |
| 12208 | 047526 | 042737 | 177760 | 003662 | SUHRET: | BIC #177760,TCCR | |
| 12209 | 047534 | 010037 | 003644 | | | MOV R0,TR0R | :SAVE REGISTER RESULTS |
| 12210 | 047540 | 010137 | 003646 | | | MOV R1,TR1R | |
| 12211 | 047544 | 010237 | 003650 | | | MOV R2,TR2R | |
| 12212 | 047550 | 010337 | 003652 | | | MOV R3,TR3R | |
| 12213 | 047554 | 010437 | 003654 | | | MOV R4,TR4R | |
| 12214 | 047560 | 010537 | 003656 | | | MOV R5,TR5R | |
| 12215 | 047564 | 032737 | 030000 | 177776 | | BIT #30000,@PSW | :WAS PREVIOUS MODE USER OR SUPV? |
| 12216 | 047572 | 001404 | | | | BEQ 4\$ | :BRANCH IF NO |
| 12217 | 047574 | 006506 | | | | MFPI SP | :GET PREVIOUS MODE SP |
| 12218 | 047576 | 012637 | 003660 | | | MOV (SP)+,TR6R | :STORE PREVIOUS SP |
| 12219 | 047602 | 000402 | | | | BR 5\$ | |
| 12220 | 047604 | 010637 | 003660 | | 4\$: | MOV SP,TR6R | |

| | | | | | | | | | |
|-------|--------|--------|--------|--------|-------|--------------------|--|--|--|
| 12221 | 047610 | 013706 | 002224 | | 5\$: | MOV TSP,SP | | | |
| 12222 | 047614 | 004737 | 054556 | | | JSR PC,CKUREG | | | ;VERIFY THAT REG SET WHICH WAS NOT |
| 12223 | | | | | | | | | ; SELECTED DID NOT GET CHANGED AND SWITCH TO |
| 12224 | | | | | | | | | ; REGISTER SET 0. |
| 12225 | 047620 | 022737 | 111111 | 120400 | | CMP #111111,PRECSK | | | ;DID CIS INST EXECUTION USE MORE THAN 64 STACK WORDS |
| 12226 | 047626 | 001012 | | | | BNE 2\$ | | | ;BRANCH IF YES |
| 12227 | 047630 | 022737 | 111111 | 120602 | | CMP #111111,PSTCSK | | | ;DID CIS INST DESTROY LOC AT STACK + 2 |
| 12228 | 047636 | 001416 | | | | BEQ 3\$ | | | ;BRANCH IF NO |
| 12229 | 047640 | | | | | PRINTB #STKM2 | | | |
| (6) | 047640 | 012746 | 011704 | | | MOV #STKM2,-(SP) | | | |
| (3) | 047644 | 010600 | | | | MOV SP,R0 | | | |
| (4) | 047646 | 004737 | 065304 | | | JSR PC,FPRINT | | | |
| 12230 | 047652 | 000405 | | | | BR 1\$ | | | |
| 12231 | 047654 | | | | 2\$: | PRINTB #STKM1 | | | |
| (6) | 047654 | 012746 | 011613 | | | MOV #STKM1,-(SP) | | | |
| (3) | 047660 | 010600 | | | | MOV SP,R0 | | | |
| (4) | 047662 | 004737 | 065304 | | | JSR PC,FPRINT | | | |
| 12232 | 047666 | 012737 | 177777 | 002152 | 1\$: | MOV #177777,ERRSTK | | | ;SET STACK ERROR FLAG |
| 12233 | 047674 | 000137 | 042010 | | 3\$: | JMP FCRTN | | | |
| 12234 | | | | | | | | | |
| 12235 | | | | | | | | | |
| 12236 | 047700 | | | | FC13: | | | | ;FLOW COMMAND = 13 - COMPARE TEST RESULTS |
| 12237 | | | | | | | | | ; TO EMULATION RESULTS - INDICATE |
| 12238 | | | | | | | | | ; ANY ERRORS. |
| 12239 | 047700 | 005037 | 002144 | | | CLR ERRCC | | | |
| 12240 | 047704 | 005037 | 002146 | | | CLR ERRREG | | | |
| 12241 | 047710 | 005037 | 002150 | | | CLR ERRBUF | | | |
| 12242 | 047714 | 004737 | 053340 | | | JSR PC,RF1 | | | ;FORM POINTER TO FIRST TEST OPERAND |
| 12243 | | | | | | | | | ; TO BE ADJUSTED. |
| 12244 | 047720 | 022701 | 003642 | | | CMP #TRN+16,R1 | | | ;IF OPERAND = 7 THEN ALL REQUIRED ADJUSTMENTS |
| 12245 | | | | | | | | | ; HAVE BEEN COMPLETED |
| 12246 | 047724 | 001446 | | | | BEQ CCCK | | | ;DONE? |
| 12247 | 047726 | 032737 | 000100 | 047464 | | BIT #100,TINST | | | ;INST TYPE (REG OR INLINE)? |
| 12248 | 047734 | 001030 | | | | BNE ADJI | | | |
| 12249 | 047736 | 004737 | 050736 | | | JSR PC,ADJR | | | ;NO - ADJUST SPECIFIED OPERANDS |
| 12250 | 047742 | 004737 | 053372 | | | JSR PC,RF2 | | | ;FORM POINTER TO 2ND TEST OPERAND |
| 12251 | | | | | | | | | ; TO BE ADJUSTED. |
| 12252 | 047746 | 022701 | 003642 | | | CMP #TRN+16,R1 | | | |
| 12253 | 047752 | 001433 | | | | BEQ CCCK | | | ;DONE? |
| 12254 | 047754 | 004737 | 050736 | | | JSR PC,ADJR | | | ;NO - ADJUST AGAIN |
| 12255 | 047760 | 004737 | 053426 | | | JSR PC,RF3 | | | ;FORM POINTER TO 3RD TEST OPERAND TO BE ADJUSTED |
| 12256 | 047764 | 022701 | 003642 | | | CMP #TRN+16,R1 | | | |
| 12257 | 047770 | 001424 | | | | BEQ CCCK | | | ;DONE? |
| 12258 | 047772 | 004737 | 050736 | | | JSR PC,ADJR | | | ;NO - ADJUST |
| 12259 | 047776 | 004737 | 053456 | | | JSR PC,RF4 | | | ;FORM POINTER TO 4TH TEST OPERAND TO BE ADJUSTED |
| 12260 | 050002 | 022701 | 003642 | | | CMP #TRN+16,R1 | | | |
| 12261 | 050006 | 001415 | | | | BEQ CCCK | | | ;DONE? |
| 12262 | 050010 | 004737 | 050736 | | | JSR PC,ADJR | | | ;NO - ADJUST |
| 12263 | 050014 | 000412 | | | | BR CCCK | | | |
| 12264 | 050016 | 062737 | 000002 | 001742 | ADJI: | ADD #2,FLOPTR | | | ;LOOK AT NEXT FLOW COMMAND (=00) |
| 12265 | 050024 | 004737 | 053340 | | | JSR PC,RF1 | | | ;FORM PTR TO INLINE TYPE INST REG OPERAND |
| 12266 | | | | | | | | | ; TO BE ADJUSTED. |
| 12267 | 050030 | 022701 | 003642 | | | CMP #TRN+16,R1 | | | ;IF OPERAND = 7 THEN NO REG OPERANDS TO ADJUST. |
| 12268 | 050034 | 001402 | | | | BEQ CCCK | | | ;DONE? |

| | | | | | | | |
|-------|--------|--------|--------|--------|----------------------------|--|---|
| 12269 | 050036 | 004737 | 050736 | | JSR PC,ADJR | | :NO ADJUST SPECIFIED OPERAND. |
| 12270 | | | | | | | :NOTE: INLINE TYPE INST NEVER REQUIRE |
| 12271 | | | | | | | : THAT MORE THAN 1 OPERAND BE ADJUSTED. |
| 12272 | 050042 | 042737 | 177760 | 003720 | CCCK: BIC #177760,ECCR | | :CLEAR OUT ALL BUT CONDITION CODES |
| 12273 | 050050 | 005737 | 035744 | | TST EZDF | | :CONDITION UNDER TEST = DIVP BY 0? |
| 12274 | 050054 | 001406 | | | BEQ 1\$ | | :BRANCH IF NO |
| 12275 | 050056 | 042737 | 000014 | 003662 | BIC #14,TCCR | | :MASK OUT ALL BUT C & V COND. CODES |
| 12276 | 050064 | 042737 | 000014 | 003720 | BIC #14,ECCR | | |
| 12277 | 050072 | 004737 | 067664 | | 1\$: JSR PC,RECCC | | :RECORD CONDITION CODE USAGE |
| 12278 | 050076 | 023737 | 003720 | 003662 | CMP ECCR,TCCR | | :CHECK CONDITION CODE RESULTS |
| 12279 | 050104 | 001403 | | | BEQ REGCK | | |
| 12280 | 050106 | 012737 | 177777 | 002144 | MOV #177777,ERRCC | | :SET CONDITION CODE ERROR FLAG |
| 12281 | 050114 | 005737 | 035744 | | REGCK: TST EZDF | | :CONDITION UNDER TEST = DIVP BY 0? |
| 12282 | 050120 | 001104 | | | BNE EZDBCK | | :BRANCH IF YES |
| 12283 | 050122 | 023737 | 003702 | 003644 | CMP EROR,TROR | | :CHECK REGISTER RESULTS |
| 12284 | 050130 | 001030 | | | BNE REGERR | | |
| 12285 | 050132 | 023737 | 003704 | 003646 | CMP ER1R,TR1R | | |
| 12286 | 050140 | 001024 | | | BNE REGERR | | |
| 12287 | 050142 | 023737 | 003706 | 003650 | CMP ER2R,TR2R | | |
| 12288 | 050150 | 001020 | | | BNE REGERR | | |
| 12289 | 050152 | 023737 | 003710 | 003652 | CMP ER3R,TR3R | | |
| 12290 | 050160 | 001014 | | | BNE REGERR | | |
| 12291 | 050162 | 023737 | 003712 | 003654 | CMP ER4R,TR4R | | |
| 12292 | 050170 | 001010 | | | BNE REGERR | | |
| 12293 | 050172 | 023737 | 003714 | 003656 | CMP ER5R,TR5R | | |
| 12294 | 050200 | 001004 | | | BNE REGERR | | |
| 12295 | 050202 | 023737 | 003716 | 003660 | CMP ER6R,TR6R | | |
| 12296 | 050210 | 001403 | | | BEQ BUFCK | | |
| 12297 | 050212 | 012737 | 177777 | 002146 | REGERR: MOV #177777,ERRREG | | :REGISTER ERROR - SET REGISTER ERROR FLAG |
| 12298 | 050220 | 032737 | 000004 | 002140 | BUFCK: BIT #4,SPHAND | | :SPECIAL HANDLING? |
| 12299 | 050226 | 001077 | | | BNE TFORE | | :IF YES SKIP BUFFER CHECK |
| 12300 | 050230 | 013701 | 001640 | | MOV TBADR,R1 | | :CHECK BUFFER RESULTS |
| 12301 | 050234 | 013702 | 001646 | | MOV EBADR,R2 | | |
| 12302 | 050240 | 005737 | 002160 | | TST MODE | | :IS MODE = KERNEL? |
| 12303 | 050244 | 001024 | | | BNE 1\$ | | :BRANCH IF NO |
| 12304 | 050246 | 023727 | 047434 | 000240 | CMP TOPC1,#NOP | | :INTERRUPTABILITY BEING TESTED? |
| 12305 | 050254 | 001404 | | | BEQ 3\$ | | :BRANCH IF YES |
| 12306 | 050256 | 023727 | 047372 | 000240 | CMP TOLTC,#NOP | | |
| 12307 | 050264 | 001014 | | | BNE 1\$ | | :BRANCH IF NO |
| 12308 | 050266 | 023727 | 047464 | 076026 | 3\$: CMP TINST,#76026 | | :IS INST UNDER TEST A L2D6 OR L3D6 |
| 12309 | 050274 | 001404 | | | BEQ 2\$ | | |
| 12310 | 050276 | 023727 | 047464 | 076066 | CMP TINST,#76066 | | |
| 12311 | 050304 | 001004 | | | BNE 1\$ | | |
| 12312 | 050306 | 062701 | 000006 | | 2\$: ADD #6,R1 | | :YES - SPECIAL CASE |
| 12313 | 050312 | 062702 | 000006 | | ADD #6,R2 | | :L2D6 OR L3D6 WITH INTERRUPTABILITY TEST.MODE=KERNEL. |
| 12314 | | | | | | | :DUE TO THE INTERRUPT OCCURRING IMMEDIATELY |
| 12315 | | | | | | | :AFTER THE L2D6 OR L3D6 INST WHEN R6 POINTS |
| 12316 | | | | | | | :TO TBADR (TEST BUFFER), THE FIRST 2 TO 3 WORDS OF |
| 12317 | | | | | | | :TBADR WILL GET DESTROYED. DONT COMPARE |
| 12318 | | | | | | | :FIRST 3 WORDS OF BUFFERS. |
| 12319 | 050316 | 023701 | 001754 | | 1\$: CMP TBEND,R1 | | :REACH END OF BUFFER YET? |
| 12320 | 050322 | 101441 | | | BLOS TFORE | | :YES |
| 12321 | 050324 | 122122 | | | CMPB (R1)+,(R2)+ | | :NO - COMPARE NEXT WORD |
| 12322 | 050326 | 001773 | | | BEQ 1\$ | | |


```

12323 050330 000421          BR BUFERR
12324
12325 050332 013701 001640    EZDBCK: MOV TBADR,R1          ;CHECK BUFFER RESULTS - ZERO DIVP CASE
12326 050336 013702 001646    MOV EBADR,R2          ;DON'T COMPARE ON DST STRING
12327 050342 023701 001754    2$:  CMP TBEND,R1      ;REACH END OF BUFFER YET?
12328 050346 101427          BLOS TFORE           ;YES
12329 050350 020237 035746    CMP R2,EZDBEG        ;AT BEGINNING OF DST STRING?
12330 050354 001005          BNE 1$              ;BRANCH IF NO
12331 050356 063702 035750    ADD EZDEND,R2        ;ADJUST POINTERS TO END OF DST STRING
12332 050362 063701 035750    ADD EZDEND,R1
12333 050366 000765          BR 2$
12334 050370 122122          1$:  CMPB (R1)+,(R2)+    ;COMPARE NEXT BYTE
12335 050372 001763          BEQ 2$
12336 050374 012737 177777 002150 BUFERR: MOV #177777,ERRBUF    ;BUFFER ERROR - SET BUFFER ERROR FLAG
12337 050402 005301          DEC R1
12338 050404 005302          DEC R2
12339 050406 010137 002174    MOV R1,AEADR
12340 050412 010237 002200    MOV R2,EMADR
12341 050416 111137 002176    MOVB (R1),AEDTA
12342 050422 111237 002202    MOVB (R2),EMDTA
12343 050426 005037 002056    TFORE: CLR ERRS        ;CLEAR ERROR INDICATOR
12344 050432 005737 002144    TST ERRCC           ;IF COMPARISON HAS TURNED UP ERRORS
12345                                ; ENTER ERROR DISPLAY ROUTINE.
12346 050436 001020          BNE CISERR
12347 050440 005737 002146    TST ERRREG
12348 050444 001015          BNE CISERR
12349 050446 005737 002150    TST ERRBUF
12350 050452 001012          BNE CISERR
12351 050454 005737 002152    TST ERRSTK
12352 050460 001007          BNE CISERR
12353 050462 005037 002266    CLR RPTFLG          ;CLEAR THE REPEAT TEST FLAG
12354 050466 005737 002042    TST NOERDS          ;DISPLAY EVEN THOUGH THERE WAS NO ERROR?
12355 050472 001017          BNE NOER            ;BRANCH IF YES
12356 050474 000137 051000    JMP SWGPC
12357
12358 050500          CISERR:
12359 050500 005237 002054    INC ERRCT           ;INCREMENT ERROR COUNT
12360 050504 005037 002204    CLR QRYFLG         ;ASSURE PRINTOUT ON ERROR
12361 050510 012737 000001 001114  MOV #1,$MSGTY       ;SET APT MESSAGE TYPE TO 1
12362 050516 013737 002132 001116  MOV FATAL,$FATAL    ;SET APT FATAL ERROR WORD
12363 050524 012737 177777 002056  MOV #177777,ERRS    ;SET ERROR INDICATOR
12364 050532
12366 050532 005737 001760    NOER:  TST RANDOM          ;IN RANDOM MODE ?
12367 050536 001 13          BEQ 40$            ;BRANCH IF NO
12368 050540          PRINTB #FORM36,STRNC,STRP1,STRP2 ;PRINT 3 SEED CONSTANTS
(9) 050540 013746 002002    MOV STRP2,-(SP)
(8) 050544 013746 002000    MOV STRP1,-(SP)
(7) 050550 013746 001776    MOV STRNC,-(SP)
(6) 050554 012746 014240    MOV #FORM36,-(SP)
(3) 050560 010600          MOV SP,R0
(4) 050562 004737 065304    JSR PC,FPRINT
12370 050566 013737 002230 003624 40$:  MOV TTR0,TRO        ;RESTORE TRN'S - NECESSARY IN CASE OPERATOR
12371 050574 013737 002232 003626    MOV TTR1,TR1        ; REQUESTS THE REPEAT OF AN IN-LINE TEST CONDITION.
12372 050602 013737 002234 003630    MOV TTR2,TR2

```

| | | | | | | |
|-------|--------|--------|--------|--------|----------------------|---|
| 12373 | 050610 | 013737 | 002236 | 003632 | MOV TTR3,TR3 | |
| 12374 | 050616 | 013737 | 002240 | 003634 | MOV TTR4,TR4 | |
| 12375 | 050624 | 013737 | 002242 | 003636 | MOV TTR5,TR5 | |
| 12376 | 050632 | 005737 | 002056 | | TST ERRS | :ERROR INDICATOR SET? |
| 12377 | 050636 | 001410 | | | BEQ 1\$ | :BRANCH IF NO |
| 12378 | 050640 | | | | PRINTB #FORM37,ERRCT | :PRINT 'ERROR # XXXXXX' |
| (?) | 050640 | 013746 | 002054 | | MOV ERRCT,-(SP) | |
| (6) | 050644 | 012746 | 014322 | | MOV #FORM37,-(SP) | |
| (3) | 050650 | 010600 | | | MOV SP,R0 | |
| (4) | 050652 | 004737 | 065304 | | JSR PC,FPRINT | |
| 12379 | 050656 | 000405 | | | BR 2\$ | |
| 12380 | 050660 | | | | 1\$: PRINTB #SDASH | :PRINT DASHES |
| (6) | 050660 | 012746 | 016527 | | MOV #SDASH,-(SP) | |
| (3) | 050664 | 010600 | | | MOV SP,R0 | |
| (4) | 050666 | 004737 | 065304 | | JSR PC,FPRINT | |
| 12381 | 050672 | | | | 2\$: PRINTB #DASH | |
| (6) | 050672 | 012746 | 016427 | | MOV #DASH,-(SP) | |
| (3) | 050676 | 010600 | | | MOV SP,R0 | |
| (4) | 050700 | 004737 | 065304 | | JSR PC,FPRINT | |
| 12382 | 050704 | 004737 | 063560 | | JSR PC,INSERR | |
| 12383 | 050710 | 000137 | 041420 | | JMP RTC | :REPEAT TEST RETURN |
| 12384 | 050714 | 000137 | 051000 | | JMP SWOPC | :NORMAL RETURN |
| 12385 | 050720 | 005737 | 002074 | | TST FSRUN | :RESTART RETURN |
| 12386 | | | | | | :FIELD SERVICE OR DESIGN VERIF TYPE RUN? |
| 12387 | 050724 | 001402 | | | BEQ 3\$ | :BRANCH IF DESIGN VERIF TYPE RUN |
| 12388 | 050726 | 000137 | 036514 | | JMP START | :RESTART FIELD SERVICE TYPE RUN |
| 12389 | 050732 | 000137 | 036710 | | 3\$: JMP DVTST | :RESTART DESIGN VERIF TYPE RUN |
| 12390 | | | | | | |
| 12391 | | | | | | |
| 12392 | | | | | .EVEN | |
| 12393 | 050736 | | | | ADJR: | :SUBROUTINE TO SUBTRACT OUT BUFFER STARTING |
| 12394 | | | | | | : ADDRESSES FROM SPECIFIED TEST AND EMULATION |
| 12395 | | | | | | : RESULTS. |
| 12396 | 050736 | 162701 | 003624 | | .SUB #TRN,R1 | |
| 12397 | 050742 | 062701 | 003644 | | ADD #TRNR,R1 | |
| 12398 | 050746 | 163711 | 001640 | | SUB TBADR,(R1) | :SUBTRACT OUT TEST BUFFER STARTING ADDRESS |
| 12399 | | | | | | : FROM TEST OPERAND |
| 12400 | 050752 | 032737 | 000100 | 047464 | BIT #100,TINST | :NOTE: FOR INLINE CASE THE EMULATION STEP WAS |
| 12401 | | | | | | : SKIPPED. THEREFORE DONT 'RE' ADJUST |
| 12402 | | | | | | : EMULATION OPERANDS. |
| 12403 | 050760 | 001006 | | | BNE 1\$ | |
| 12404 | 050762 | 162701 | 003644 | | SUB #TRNR,R1 | :POINT R1 TO EMULATION OPERAND RATHER THAT |
| 12405 | 050766 | 062701 | 003702 | | ADD #ERNR,R1 | : TEST OPERAND. |
| 12406 | 050772 | 163711 | 001646 | | SUB EBADR,(R1) | :SUBTRACT OUT EMULATION BUFFER STARTING |
| 12407 | | | | | | : ADDRESS FROM EMULATION OPERAND. |
| 12408 | 050776 | 000207 | | | 1\$: RTS PC | |
| 12409 | | | | | | |
| 12410 | | | | | | |
| 12411 | | | | | | |
| 12412 | 051000 | | | | FC14: | :FLOW COMMAND = 14 - UPDATE POINTERS |
| 12413 | | | | | | : FOR NEXT TEST CONDITION. |
| 12414 | 051000 | 013701 | 047464 | | SWOPC: MOV TINST,R1 | :IS INST = L2D OR L3D ?? |
| 12415 | 051004 | 042701 | 000007 | | BIC #7,R1 | |
| 12416 | 051010 | 022701 | 076020 | | CMP #076020,R1 | |

| | | | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|--|--|--|
| 12417 | 051014 | 001437 | | | | BEQ 1\$ | | | ;BRANCH IF YES (NO INLINE CASE) |
| 12418 | 051016 | 022701 | 076060 | | | CMP #076060,R1 | | | |
| 12419 | 051022 | 001434 | | | | BEQ 1\$ | | | |
| 12420 | 051024 | 032737 | 140000 | 002140 | | BIT #140000,SPHAND | | | ;SKIP INLINE AND ADDITIONAL DATA TYPE TESTING |
| 12421 | 051032 | 001030 | | | | BNE 1\$ | | | ;BRANCH IF YES |
| 12422 | 051034 | 005737 | 001760 | | | TST RANDOM | | | ;RANDOM EXERCISE MODE? |
| 12423 | 051040 | 001027 | | | | BNE 2\$ | | | ;BRANCH IF YES |
| 12424 | | | | | | | | | ; NOTE: IN RANDOM EXERCISE MODE DATA TYPES ARE |
| 12425 | | | | | | | | | ; SELECTED RANDOMLY FOR EACH TEST CONDITION. |
| 12426 | 051042 | 005737 | 002434 | | | TST DECINS | | | ;DECIMAL INST? |
| 12427 | 051046 | 001424 | | | | BEQ 2\$ | | | ;BRANCH IF NO |
| 12428 | 051050 | 032737 | 000100 | 047464 | | BIT #100,TINST | | | ;INLINE INST JUST TESTED |
| 12429 | 051056 | 001020 | | | | BNE 2\$ | | | ;BRANCH IF YES |
| 12430 | 051060 | 023727 | 002454 | 000001 | | CMP ZPM,#1 | | | ;IS INST ZONED ,PACKED, OR MIXED? |
| 12431 | 051066 | 001420 | | | | BEQ PACKED | | | ;BRANCH IF PACKED |
| 12432 | 051070 | 023727 | 002454 | 000002 | | CMP ZPM,#2 | | | |
| 12433 | 051076 | 001533 | | | | BEQ ZONED | | | ;BRANCH IF ZONED |
| 12434 | 051100 | 023727 | 002454 | 000003 | | CMP ZPM,#3 | | | |
| 12435 | 051106 | 001406 | | | | BEQ 3\$ | | | ;BRANCH IF MIXED ZONED PACKED |
| 12436 | 051110 | 000137 | 051734 | | | JMP MIXDPZ | | | ;MIXED PACKED ZONED |
| 12437 | 051114 | 000137 | 052326 | | 1\$: | JMP NXTTC | | | |
| 12438 | 051120 | 000137 | 052252 | | 2\$: | JMP CFINL | | | |
| 12439 | 051124 | 000137 | 052100 | | 3\$: | JMP MIXDZP | | | |
| 12440 | | | | | | | | | |
| 12441 | 051130 | 032737 | 000001 | 002450 | PACKED: | BIT #1,PKPTW | | | ;SET STRING 1 DESC TYPE BASED ON BIT 0 |
| 12442 | 051136 | 001004 | | | | BNE 1\$ | | | ; CONTENTS OF PKPTW |
| 12443 | 051140 | 012737 | 000006 | 002436 | | MOV #6,S1TYPE | | | |
| 12444 | 051146 | 000403 | | | | BR 2\$ | | | |
| 12445 | 051150 | 012737 | 000007 | 002436 | 1\$: | MOV #7,S1TYPE | | | |
| 12446 | | | | | | | | | |
| 12447 | 051156 | 032737 | 000002 | 002450 | 2\$: | BIT #2,PKPTW | | | ;SET STRING 2 DESC TYPE BASED ON |
| 12448 | 051164 | 001004 | | | | BNE 3\$ | | | ; BIT 1 CONTENTS OF PKPTW |
| 12449 | 051166 | 012737 | 000006 | 002440 | | MOV #6,S2TYPE | | | |
| 12450 | 051174 | 000403 | | | | BR 4\$ | | | |
| 12451 | 051176 | 012737 | 000007 | 002440 | 3\$: | MOV #7,S2TYPE | | | |
| 12452 | | | | | | | | | |
| 12453 | 051204 | 032737 | 000004 | 002450 | 4\$: | BIT #4,PKPTW | | | ;SET STRING 3 DESC TYPE BASED |
| 12454 | 051212 | 001004 | | | | BNE 5\$ | | | ; BIT 2 CONTENTS OF PKPTW |
| 12455 | 051214 | 012737 | 000006 | 002442 | | MOV #6,S3TYPE | | | |
| 12456 | 051222 | 000403 | | | | BR 6\$ | | | |
| 12457 | 051224 | 012737 | 000007 | 002442 | 5\$: | MOV #7,S3TYPE | | | |
| 12458 | | | | | | | | | |
| 12459 | 051232 | 005737 | 002446 | | 6\$: | TST MIXTYP | | | ;MIX TYPES WITHIN INSTRUCTION? |
| 12460 | 051236 | 001011 | | | | BNE 61\$ | | | ;BRANCH IF YES |
| 12461 | 051240 | 005137 | 002450 | | | COM PKPTW | | | ;NO-SWITCH ALL DESC FROM SIGNED |
| 12462 | | | | | | | | | ; TO UNSIGNED (OR VICE VERSA) |
| 12463 | 051244 | 005737 | 002450 | | | TST PKPTW | | | ;ALL TYPES TESTED? |
| 12464 | 051250 | 001002 | | | | BNE 60\$ | | | ;BRANCH IF NO |
| 12465 | 051252 | 000137 | 052252 | | | JMP CFINL | | | ;EXIT TO IN LINE TESTING |
| 12466 | 051256 | 000137 | 051362 | | 60\$: | JMP TYPSET | | | |
| 12467 | 051262 | 005237 | 002450 | | 61\$: | INC PKPTW | | | ;CHANGE TO NEXT MIXED CASE |
| 12468 | 051266 | 023727 | 002452 | 000001 | | CMP NDESC,#1 | | | ;IS THERE ONLY 1 DESC FOR THIS INST? |
| 12469 | 051274 | 001010 | | | | BNE 7\$ | | | ;BRANCH IF NO |
| 12470 | 051276 | 032737 | 000002 | 002450 | | BIT #2,PKPTW | | | ;ALL TYPE MIXTURES TESTED? |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|----------------|---|
| 12471 | 051304 | 001426 | | | 62\$: | BEQ TYPSET | :BRANCH IF NO |
| 12472 | 051306 | 005037 | 002450 | | | CLR PKPTW | :CLEANUP FOR NEXT TEST CONDITION |
| 12473 | 051312 | 000137 | 052252 | | | JMP CFINL | :EXIT TO TEST IN-LINE INST. |
| 12474 | | | | | | | |
| 12475 | 051316 | 023727 | 002452 | 000002 | 7\$: | CMP NDESC,#2 | :ARE THERE 2 DESC. FOR THIS INST? |
| 12476 | 051324 | 001004 | | | | BNE 8\$ | :BRANCH IF NO |
| 12477 | 051326 | 032737 | 000004 | 002450 | | BIT #4,PKPTW | :ALL TYPE MIXTURES TESTED? |
| 12478 | 051334 | 000763 | | | | BR 62\$ | |
| 12479 | | | | | | | |
| 12480 | 051336 | 023727 | 002452 | 000003 | 8\$: | CMP NDESC,#3 | :ARE THERE 3 DESC FOR THIS INST? |
| 12481 | 051344 | 001401 | | | | BEQ 81\$ | |
| 12482 | 051346 | 000000 | | | | HALT | :# OF DESCRIPTORS FOR INST UNDER TEST |
| 12483 | | | | | | | : DOES NOT MAKE SENSE. |
| 12484 | 051350 | 032737 | 000010 | 002450 | 81\$: | BIT #10,PKPTW | |
| 12485 | 051356 | 000137 | 051304 | | | JMP 62\$ | |
| 12486 | | | | | | | |
| 12487 | 051362 | 000137 | 041342 | | TYPSET: | JMP NTCTS | :REPEAT TEST CONDITION WITH A DIFFERENT |
| 12488 | | | | | | | : DATA TYPE. |
| 12489 | | | | | | | |
| 12490 | | | | | | | |
| 12491 | 051366 | 005737 | 002446 | | ZONED: | TST MIXTYP | :MIX DATA TYPES WITHIN INST?? |
| 12492 | 051372 | 001003 | | | | BNE 1\$ | :BRANCH IF YES |
| 12493 | 051374 | 062737 | 000110 | 002450 | | ADD #110,PKPTW | |
| 12494 | 051402 | 005237 | 002450 | | 1\$: | INC PKPTW | |
| 12495 | 051406 | 013701 | 002450 | | 11\$: | MOV PKPTW,R1 | :SET STRING 1 DESC TYPE FROM |
| 12496 | 051412 | 042701 | 177770 | | | BIC #177770,R1 | : BITS 0,1,2 OF PKPTW |
| 12497 | 051416 | 010137 | 002436 | | | MOV R1,S1TYPE | |
| 12498 | | | | | | | |
| 12499 | 051422 | 013701 | 002450 | | | MOV PKPTW,R1 | :SET STRING 2 DESC TYPE FROM |
| 12500 | 051426 | 042701 | 177707 | | | BIC #177707,R1 | : BITS 3,4,5 OF PKPTW. |
| 12501 | 051432 | 006201 | | | | ASR R1 | |
| 12502 | 051434 | 006201 | | | | ASR R1 | |
| 12503 | 051436 | 006201 | | | | ASR R1 | |
| 12504 | 051440 | 010137 | 002440 | | | MOV R1,S2TYPE | |
| 12505 | | | | | | | |
| 12506 | 051444 | 013701 | 002450 | | | MOV PKPTW,R1 | :SET STRING 3 DESC TYPE FROM |
| 12507 | 051450 | 042701 | 177077 | | | BIC #177077,R1 | : BITS 6,7,8 OF PKPTW. |
| 12508 | 051454 | 006301 | | | | ASL R1 | |
| 12509 | 051456 | 006301 | | | | ASL R1 | |
| 12510 | 051460 | 000301 | | | | SWAB R1 | |
| 12511 | 051462 | 010137 | 002442 | | | MOV R1,S3TYPE | |
| 12512 | | | | | | | |
| 12513 | 051466 | 005737 | 002446 | | | TST MIXTYP | :MIX TYPES WITHIN INST? |
| 12514 | 051472 | 001016 | | | | BNE 61\$ | :BRANCH IF YES |
| 12515 | 051474 | 023727 | 002450 | 000666 | | CMP PKPTW,#666 | :ALL TYPES TESTED? |
| 12516 | 051502 | 001327 | | | 2\$: | BNE TYPSET | :BRANCH IF NO |
| 12517 | 051504 | 005037 | 002450 | | 12\$: | CLR PKPTW | |
| 12518 | 051510 | 005037 | 002436 | | | CLR S1TYPE | |
| 12519 | 051514 | 005037 | 002440 | | | CLR S2TYPE | |
| 12520 | 051520 | 005037 | 002442 | | | CLR S3TYPE | |
| 12521 | 051524 | 000137 | 052252 | | | JMP CFINL | :EXIT TO TEST IN-LINE INST |
| 12522 | | | | | | | |
| 12523 | 051530 | 023727 | 002452 | 000001 | 61\$: | CMP NDESC,#1 | :IS THERE ONLY 1 DESC FOR THIS INST |
| 12524 | 051536 | 001007 | | | | BNE 7\$ | :BRANCH IF NO |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|----------------|--|
| 12525 | 051540 | 013701 | 002450 | | | MOV PKPTW,R1 | |
| 12526 | 051544 | 042701 | 177770 | | | BIC #177770,R1 | |
| 12527 | 051550 | 020127 | 000006 | | | CMP R1,#6 | ;ALL TYPES TESTED? |
| 12528 | 051554 | 000752 | | | | BR 2\$ | |
| 12529 | | | | | | | |
| 12530 | 051556 | 023727 | 002452 | 000002 | 7\$: | CMP NDESC,#2 | ;ARE THERE 2 DESC FOR THIS INST? |
| 12531 | 051564 | 001022 | | | | BNE 8\$ | ;BRANCH IF NO |
| 12532 | 051566 | 013701 | 002450 | | | MOV PKPTW,R1 | |
| 12533 | 051572 | 042701 | 177770 | | | BIC #177770,R1 | |
| 12534 | 051576 | 020127 | 000006 | | | CMP R1,#6 | |
| 12535 | 051602 | 001267 | | | | BNE TYPSET | |
| 12536 | 051604 | 062737 | 000002 | 002450 | | ADD #2,PKPTW | |
| 12537 | 051612 | 013701 | 002450 | | | MOV PKPTW,R1 | |
| 12538 | 051616 | 042701 | 177707 | | | BIC #177707,R1 | |
| 12539 | 051622 | 020127 | 000060 | | | CMP R1,#60 | ;ALL TYPE MIXTURES TESTED? |
| 12540 | 051626 | 001267 | | | | BNE 11\$ | |
| 12541 | 051630 | 000725 | | | | BR 12\$ | |
| 12542 | | | | | | | |
| 12543 | 051632 | 023727 | 002452 | 000003 | 8\$: | CMP NDESC,#3 | ;ARE THERE 3 DESC FOR THIS INST? |
| 12544 | 051640 | 001401 | | | | BEQ 81\$ | ;BRANCH IF YES |
| 12545 | 051642 | 000000 | | | | HALT | ;# OF DESC FOR INST UNDER TEST DOES ; NOT MAKE SENSE. |
| 12546 | | | | | | | |
| 12547 | 051644 | 013701 | 002450 | | 81\$: | MOV PKPTW,R1 | |
| 12548 | 051650 | 042701 | 177770 | | | BIC #177770,R1 | |
| 12549 | 051654 | 020127 | 000006 | | | CMP R1,#6 | |
| 12550 | 051660 | 001240 | | | | BNE TYPSET | |
| 12551 | 051662 | 062737 | 000002 | 002450 | | ADD #2,PKPTW | |
| 12552 | 051670 | 013701 | 002450 | | | MOV PKPTW,R1 | |
| 12553 | 051674 | 042701 | 177707 | | | BIC #177707,R1 | |
| 12554 | 051700 | 020127 | 000060 | | | CMP R1,#60 | |
| 12555 | 051704 | 001240 | | | | BNE 11\$ | |
| 12556 | 051706 | 062737 | 000020 | 002450 | | ADD #20,PKPTW | |
| 12557 | 051714 | 013701 | 002450 | | | MOV PKPTW,R1 | |
| 12558 | 051720 | 042701 | 177077 | | | BIC #177077,R1 | |
| 12559 | 051724 | 020127 | 000600 | | | CMP R1,#600 | ;ALL TYPE MIXTURES TESTED? |
| 12560 | 051730 | 001226 | | | | BNE 11\$ | |
| 12561 | 051732 | 000664 | | | | BR 12\$ | |
| 12562 | | | | | | | |
| 12563 | 051734 | 005237 | 002450 | | MIXDPZ: | INC PKPTW | |
| 12564 | 051740 | 005737 | 002446 | | | TST MIXTYP | ;MIX TYPES WITHIN INST? |
| 12565 | 051744 | 001407 | | | | BEQ 1\$ | ;BRANCH IF NO |
| 12566 | 051746 | 032737 | 000007 | 002450 | | BIT #7,PKPTW | ;YES - IF PACK DATA TYPE=0,CHANGE IT TO 6. |
| 12567 | 051754 | 001003 | | | | BNE 1\$ | |
| 12568 | 051756 | 052737 | 000006 | 002450 | | BIS #6,PKPTW | |
| 12569 | 051764 | 013701 | 002450 | | 1\$: | MOV PKPTW,R1 | ;SET STRING 1 DESC TYPE FROM |
| 12570 | 051770 | 042701 | 177770 | | | BIC #177770,R1 | ; BITS 0,1,2 OF PKPTW |
| 12571 | 051774 | 010137 | 002436 | | | MOV R1,S1TYPE | |
| 12572 | | | | | | | |
| 12573 | 052000 | 013701 | 002450 | | | MOV PKPTW,R1 | ;SET STRING 2 DESC TYPE FROM |
| 12574 | 052004 | 042701 | 177707 | | | BIC #177707,R1 | ; BITS 3,4,5 OF PKPTW |
| 12575 | 052010 | 006201 | | | | ASR R1 | |
| 12576 | 052012 | 006201 | | | | ASR R1 | |
| 12577 | 052014 | 006201 | | | | ASR R1 | |
| 12578 | 052016 | 010137 | 002440 | | | MOV R1,S2TYPE | |

| | | | | | | | | | |
|-------|--------|--------|--------|--------|---------|----------------|--|--|--|
| 12579 | | | | | | | | | |
| 12580 | 052022 | 005737 | 002446 | | | TST MIXTYP | | | ;MIX TYPES WITHIN INST? |
| 12581 | 052026 | 001016 | | | | BNE 61\$ | | | ;BRANCH IF YES |
| 12582 | 052030 | 023727 | 002450 | 000010 | | CMP PKPTW,#10 | | | ;ALL TYPES TESTED? |
| 12583 | 052036 | 001016 | | | 60\$: | BNE 62\$ | | | ;BRANCH IF NO |
| 12584 | 052040 | 012737 | 000006 | 002450 | | MOV #06,PKPTW | | | |
| 12585 | 052046 | 012737 | 000006 | 002436 | | MOV #6,S1TYPE | | | ;INITIALIZE S1TYPE & S2TYPE FOR NEXT TEST CONDITION |
| 12586 | 052054 | 005037 | 002440 | | | CLR S2TYPE | | | |
| 12587 | 052060 | 000137 | 052252 | | | JMP CFINL | | | ;EXIT TO TEST IN-LINE INST |
| 12588 | | | | | | | | | |
| 12589 | 052064 | 023727 | 002450 | 000066 | 61\$: | CMP PKPTW,#66 | | | |
| 12590 | 052072 | 000761 | | | | BR 60\$ | | | |
| 12591 | 052074 | 000137 | 051362 | | 62\$: | JMP TYPSET | | | |
| 12592 | | | | | | | | | |
| 12593 | 052100 | 005237 | 002450 | | MIXDZP: | INC PKPTW | | | |
| 12594 | 052104 | 005737 | 002446 | | | TST MIXTYP | | | ;MIX TYPES WITHIN INST?? |
| 12595 | 052110 | 001412 | | | | BEQ 1\$ | | | ;BRANCH IF NO |
| 12596 | 052112 | 013701 | 002450 | | | MOV PKPTW,R1 | | | ;YES - IF ZONED DATA TYPE=6,CHANGE IT TO 0. |
| 12597 | 052116 | 042701 | 177770 | | | BIC #177770,R1 | | | |
| 12598 | 052122 | 020127 | 000006 | | | CMP R1,#6 | | | |
| 12599 | 052126 | 001003 | | | | BNE 1\$ | | | |
| 12600 | 052130 | 062737 | 000002 | 002450 | | ADD #2,PKPTW | | | |
| 12601 | 052136 | 013701 | 002450 | | 1\$: | MOV PKPTW,R1 | | | ;SET STRING 1 DESC TYPE FROM |
| 12602 | 052142 | 042701 | 177770 | | | BIC #177770,R1 | | | ; BITS 0,1,2 OF PKPTW |
| 12603 | 052146 | 010137 | 002436 | | | MOV R1,S1TYPE | | | |
| 12604 | | | | | | | | | |
| 12605 | 052152 | 013701 | 002450 | | | MOV PKPTW,R1 | | | ;SET STRING 2 DESC TYPE FROM |
| 12606 | 052156 | 042701 | 177707 | | | BIC #177707,R1 | | | ; BITS 3,4,5 OF PKPTW |
| 12607 | 052162 | 006201 | | | | ASR R1 | | | |
| 12608 | 052164 | 006201 | | | | ASR R1 | | | |
| 12609 | 052166 | 006201 | | | | ASR R1 | | | |
| 12610 | 052170 | 010137 | 002440 | | | MOV R1,S2TYPE | | | |
| 12611 | 052174 | 005737 | 002446 | | | TST MIXTYP | | | ;MIX TYPES WITHIN INST? |
| 12612 | 052200 | 001016 | | | | BNE 61\$ | | | ;BRANCH IF YES |
| 12613 | 052202 | 023727 | 002450 | 000066 | | CMP PKPTW,#66 | | | ;ALL TYPES TESTED? |
| 12614 | 052210 | 001016 | | | 60\$: | BNE 62\$ | | | ;BRANCH IF NO |
| 12615 | 052212 | 012737 | 000060 | 002450 | | MOV #60,PKPTW | | | |
| 12616 | 052220 | 005037 | 002436 | | | CLR S1TYPE | | | |
| 12617 | 052224 | 012737 | 000006 | 002440 | | MOV #6,S2TYPE | | | |
| 12618 | 052232 | 000137 | 052252 | | | JMP CFINL | | | ;EXIT TO TEST IN-LINE INST |
| 12619 | | | | | | | | | |
| 12620 | 052236 | 022737 | 000100 | 002450 | 61\$: | CMP #100,PKPTW | | | |
| 12621 | 052244 | 000761 | | | | BR 60\$ | | | |
| 12622 | 052246 | 000137 | 051362 | | 62\$: | JMP TYPSET | | | |
| 12623 | | | | | | | | | |
| 12624 | 052252 | 032737 | 000100 | 047464 | CFINL: | BIT #100,TINST | | | ;INST TYPE (REG OR IN-LINE)? |
| 12625 | 052260 | 001022 | | | | BNE NXTTC | | | ;BRANCH IF IN-LINE TYPE |
| 12626 | 052262 | 005737 | 001760 | | | TST RANDOM | | | ;RANDOM EXERCISE MODE? |
| 12627 | 052266 | 001407 | | | | BEQ 1\$ | | | ;BRANCH IF NO |
| 12628 | 052270 | 004737 | 063464 | | | JSR PC,RN | | | ;GET A RANDOM # |
| 12629 | 052274 | 042700 | 177770 | | | BIC #177770,R0 | | | ;LOOK AT 3 LEAST SIGN BITS |
| 12630 | 052300 | 020027 | 000003 | | | CMP R0,#3 | | | ;IF THEY ARE = TO 3 (ARBITRARY CONSTANT) THEN DO IN-LINE |
| 12631 | 052304 | 001010 | | | | BNE NXTTC | | | |
| 12632 | 052306 | 052737 | 000100 | 047464 | 1\$: | BIS #100,TINST | | | ;SWITCH REG OP-CODE AT TINST TO IN-LINE OP-CODE. |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|------------------|--|
| 12633 | 052314 | 052737 | 000100 | 002132 | | BIS #100,FATAL | ;SET IN-LINE FIELD IN FATAL ERROR INDICATOR WORD |
| 12634 | 052322 | 000137 | 041064 | | | JMP NTC | ;REPEAT TEST CONDITION WITH IN-LINE OPCODE. |
| 12635 | 052326 | 042737 | 000100 | 047464 | NXTTC: | BIC #100,TINST | |
| 12636 | 052334 | 042737 | 000100 | 002132 | | BIC #100,FATAL | ;CLR IN-LINE FIELD IN FATAL ERROR INDICATOR |
| 12637 | 052342 | 005737 | 041572 | | | TST MTYPE | ;IF TYPE = 0 THEN READY FOR NEXT INPUT TABLE ENTRY |
| 12638 | 052346 | 001402 | | | | BEQ 1\$ | ;BRANCH IF TYPE NOT = 0 |
| 12639 | 052350 | 000137 | 052646 | | | JMP RFNITE | |
| 12640 | 052354 | 013737 | 001656 | 002126 | 1\$: | MOV IPNU,VIP | ;SETUP A POINTER TO LAST PARAMETER IN |
| 12641 | 052362 | 063737 | 001656 | 002126 | | ADD IPNU,VIP | ; INPUT TABLE ENTRY. |
| 12642 | 052370 | 013703 | 002126 | | UPTP: | MOV VIP,R3 | |
| 12643 | 052374 | 062703 | 001566 | | | ADD #PTP,R3 | ;R3 POINTS TO PARAMETER TABLE POINTER |
| 12644 | 052400 | 013701 | 002134 | | | MOV INPTP,R1 | |
| 12645 | 052404 | 063701 | 002126 | | | ADD VIP,R1 | |
| 12646 | 052410 | 062701 | 000002 | | | ADD #2,R1 | |
| 12647 | 052414 | 005711 | | | | TST (R1) | ; (R1) POINTS TO 1ST ENTRY IN PARAMETER TABLE |
| 12648 | 052416 | 001464 | | | | BEQ TNXP | |
| 12649 | 052420 | 017137 | 000000 | 002130 | | MOV @(R1),PTW1 | ;PTW1 CONTAINS TYPE,SIZE, AND # OF ENTRIES |
| 12650 | | | | | | | ; IN PARAMETER TABLE |
| 12651 | 052426 | 042737 | 177400 | 002130 | | BIC #177400,PTW1 | ;STRIP OFF TYPE AND ENTRY SIZE |
| 12652 | 052434 | 122737 | 000001 | 002130 | | CMPB #1,PTW1 | ;# OF ENTRIES = 1? |
| 12653 | 052442 | 001452 | | | | BEQ TNXP | ;YES - NO UPDATING. TRY NEXT PARAMETER |
| 12654 | 052444 | 005737 | 002130 | | | TST PTW1 | ;# OF ENTRIES = 0? |
| 12655 | 052450 | 001447 | | | | BEQ TNXP | ;YES - NO UPDATING. TRY NEXT PARAMETER |
| 12656 | 052452 | 017102 | 000000 | | | MOV @(R1),R2 | ;PARAMETER TABLE CONTAINS MORE THAN 1 ENTRY. |
| 12657 | 052456 | 042702 | 037777 | | | BIC #037777,R2 | ;LOOK AT ENTRY TYPE |
| 12658 | 052462 | 005702 | | | | TST R2 | |
| 12659 | 052464 | 001401 | | | | BEQ FLE | ;FIXED LENGTH ENTRIES |
| 12660 | 052466 | 000447 | | | | BR VLE | ;VARIABLE LENGTH ENTRIES |
| 12661 | | | | | | | |
| 12662 | 052470 | | | | FLE: | | ;PARAMETER TABLE CONTAINS FIXED |
| 12663 | | | | | | | ; LENGTH ENTRIES. |
| 12664 | 052470 | 017137 | 000000 | 002130 | | MOV @(R1),PTW1 | |
| 12665 | 052476 | 000337 | 002130 | | | SWAB PTW1 | |
| 12666 | 052502 | 113704 | 002130 | | | MOVB PTW1,R4 | ;LOAD R4 WITH ENTRY SIZE IN WORDS |
| 12667 | 052506 | 006304 | | | | ASL R4 | ;CONVERT SIZE IN WORDS TO BYTES |
| 12668 | 052510 | 005002 | | | | CLR R2 | |
| 12669 | 052512 | 000337 | 002130 | | | SWAB PTW1 | |
| 12670 | 052516 | 042737 | 177400 | 002130 | | BIC #177400,PTW1 | ;PTW1 CONTAINS # OF ENTRIES IN |
| 12671 | | | | | | | ; PARAMETER TABLE. |
| 12672 | 052524 | 005337 | 002130 | | 1\$: | DEC PTW1 | ;CALCULATE SIZE OF PARAMETER TABLE |
| 12673 | 052530 | 005737 | 002130 | | | TST PTW1 | ; SIZE = ENTRY SIZE X # OF ENTRIES |
| 12674 | 052534 | 001402 | | | | BEQ GPTESZ | ;CALCULATION COMPLETE - R2 CONTAINS SIZE |
| 12675 | | | | | | | ; OF PARAMETER TABLE (MINUS 1 ENTRY) |
| 12676 | 052536 | 060402 | | | | ADD R4,R2 | |
| 12677 | 052540 | 000771 | | | | BR 1\$ | |
| 12678 | | | | | | | |
| 12679 | 052542 | | | | GPTESZ: | | |
| 12680 | 052542 | 061102 | | | | ADD (R1),R2 | ;UPDATE R2 TO CONTAIN ADDRESS OF |
| 12681 | 052544 | 062702 | 000002 | | | ADD #2,R2 | ; LAST ENTRY IN PARAMETER TABLE |
| 12682 | 052550 | 021302 | | | | CMP (R3),R2 | |
| 12683 | 052552 | 001403 | | | | BEQ RESPTP | ;CURRENT PTP POINTS TO LAST ENTRY IN |
| 12684 | | | | | | | ; PARAMETER TABLE. RESET TO 1ST ENTRY |
| 12685 | 052554 | 060413 | | | | ADD R4,(R3) | ;STILL MORE ENTRIES TO TRY IN PARAMETER |
| 12686 | | | | | | | ; TABLE. UPDATE PTP TO NEXT ENTRY. |

| | | | | | | | |
|-------|--------|--------|--------|--------|------------------|--|--|
| 12687 | 052556 | 000137 | 041064 | | JMP NTC | | ;TRY NEXT TEST CONDITON FOR SAME CIS INST. |
| 12688 | | | | | | | |
| 12689 | 052562 | | | | RESPTP: | | ;RESET PTP BACK TO 1ST ENTRY IN TABLE |
| 12690 | 052562 | 011113 | | | MOV (R1),(R3) | | |
| 12691 | 052564 | 062713 | 000002 | | ADD #2,(R3) | | |
| 12692 | 052570 | 162737 | 000002 | 002126 | TNXP: SUB #2,VIP | | ;BACKUP TO NEXT PARAMETER TABLE POINTER ; AND UPDATE IT. |
| 12693 | | | | | | | |
| 12694 | 052576 | 005737 | 002126 | | TST VIP | | |
| 12695 | 052602 | 001421 | | | BEQ RFNITE | | ;ALL TEST CONDITIONS ASSOCIATED WITH CURRENT ; INPUT TABLE ENTRY EXERCISED. |
| 12696 | | | | | | | |
| 12697 | 052604 | 000671 | | | BR UPTP | | |
| 12698 | | | | | | | |
| 12699 | 052606 | | | | VLE: | | ;PARAMETER TABLE CONTAINS VARIABLE LENGTH ENTRIES |
| 12700 | 052606 | 113704 | 002130 | | MOVB PTW1,R4 | | ;R4 CONTAINS # OF ENTRIES IN PARAMETER TABLE |
| 12701 | 052612 | 005304 | | | DEC R4 | | ;DETERMINE ADDRESS OF LAST ENTRY IN PARAMETER TABLE |
| 12702 | 052614 | 011102 | | | MOV (R1),R2 | | ;R2 POINTS TO 1ST WORD IN PARAMETER TABLE |
| 12703 | 052616 | 005722 | | | 1\$: TST (R2)+ | | ;SEARCH FOR 0 WORDS |
| 12704 | 052620 | 001376 | | | BNE 1\$ | | |
| 12705 | 052622 | 005304 | | | DEC R4 | | ;AT LAST ENTRY YET? |
| 12706 | 052624 | 001374 | | | BNE 1\$ | | ;NO |
| 12707 | 052626 | 021302 | | | CMP (R3),R2 | | ;YES - IS CURRENT PTP POINTING TO LAST ENTRY? |
| 12708 | 052630 | 001754 | | | BEQ RESPTP | | ;YES - RESET PTP TO 1ST ENTRY |
| 12709 | 052632 | 011304 | | | MOV (R3),R4 | | ;NO - UPDATE PTP TO NEXT ENTRY. |
| 12710 | 052634 | 005724 | | | 2\$: TST (R4)+ | | |
| 12711 | 052636 | 001376 | | | BNE 2\$ | | |
| 12712 | 052640 | 010413 | | | MOV R4,(R3) | | |
| 12713 | 052642 | 000137 | 041064 | | JMP NTC | | ;TRY NEXT TEST CONDITION FOR SAME INPUT ; TABLE ENTRY. |
| 12714 | | | | | | | |
| 12715 | | | | | | | |


```
12717 052646          RFNITE:          ;READY FOR NEXT INPUT TABLE ENTRY.
12718 052646 005737 001760          TST RANDOM          ;RANDOM EXERCISE MODE?
12719 052652 001402          BEQ 2$              ;BRANCH IF NO
12720 052654 000137 040446          JMP NITE
12721 052660 013701 002134          2$: MOV INPTP,R1      ;SAVE INPUT TABLE POINTER
12722 052664 063737 001656 002134  3$: ADD IPNU,INPTP    ;UPDATE INPTP TO 1ST WORD OF NEXT ENTRY.
12723 052672 063737 001656 002134    ADD IPNU,INPTP
12724 052700 062737 000004 002134    ADD #4,INPTP
12725 052706 005777 127222          TST @INPTP          ;NOTE: TO ELIMINATE AN INPUT TABLE FROM
12726                               ; A TEST RUN SET THE FIRST WORD IN THE
12727                               ; INPUT TABLE TO 177777
12728 052712 100764          BMI 3$              ;IF MINUS SKIP THIS TABLE
12729 052714 005777 127214          TST @INPTP
12731 052720 001440          BEQ 5$
12736 052722 005737 002206          TST QVMODE          ;GO ON TO PROBEAHEAD ABORT TESTS
12737 052726 001406          BEQ 4$              ;RUNNING IN QV MODE?
12738 052730 013700 002134          MOV INPTP,R0        ;BRANCH IF NO
12739 052734 033760 002212 000002    BIT PTQV,2(R0)      ;LOOK AT ENTRY TYPE WORD
12740                               ; IS QV MODE BIT ASSOCIATED WITH PROCESSOR
12741 052742 001750          BEQ 3$              ; TYPE UNDER TEST SET IN ENTRY TYPE WORD?
12742 052744 021177 127164          4$: CMP (R1),@INPTP  ;BRANCH IF NO (SKIP TABLE)
12743 052750 001422          BEQ 1$              ;NEXT INST FOR TESTING = CURRENT INST?
12744 052752 005737 002464          TST ONEINS          ;BRANCH IF YES
12745 052756 001023          BNE DONE            ;SINGLE INSTRUCTION TESTING?
12746 052760 005737 002044          TST PROGD           ;BRANCH IF YES
12747 052764 001014          BNE 1$              ;PROGRESS DISPLAY?
12748 052766 017701 127142          MOV @INPTP,R1       ;BRANCH IF NO
12749 052772 006301          ASL R1              ;SETUP PTR TO PROPER MESSAGE HEADER
12750 052774 062701 067730          ADD #INEM,R1
12751 053000 011137 002142          MOV (R1),EMPTR
12753 053004 042737 000007 047464    BIC #7,TINST
12754                               ;CLEAR REGISTER FIELD OF L3D INST OP CODE
12755 053012 004737 063370          JSR PC,IDINST       ;(FOR PROGRESS DISPLAY ONLY)
12757 053016 000137 040446          1$: JMP NITE         ;IDENTIFY NEXT INSTRUCTION TO BE TESTED
12759 053022 000137 110722          5$: JMP PROBAH
12761 053026          DONE:
12762 053026 005037 001420          CLR TOTTC
12763 053032 005037 001416          CLR TOTTC
12764 053036 005237 001122          INC $PASS
12765 053042 005037 001120          CLR $TESTN
12766 053046 005737 002206          TST QVMODE          ;INCREMENT PASS COUNTER IN APT MAILBOX
12767 053052 001403          BEQ 1$              ;IF IN QV MODE TYPE END OF QV PASS INDICATOR
12768 053054 104400          TYPE
12769 053056 015611          ENDP
12770 053060 000402          BR 2$
12772 053062 104400          1$: TYPE
12773 053064 015521          ENDP
12778 053066 013700 000042          2$: MOV @#42,R0
12780 053072 001407          BEQ HERE
12785 053074 000005          RESET
12786 053076 004710          ENDAD: JSR PC,(R0)
12787 053100 000240          NOP
12788 053102 000240          NOP
12789 053104 000240          NOP
```

```

12795 053106 000137 036514          JMP START
12797 053112          HERE:
12799 053112 005737 002074          TST FSRUN          ;NORMAL FIELD SERVICE RUN?
12800 053116 001445          BEQ NFS           ;BRANCH IF NO
12801 053120 005737 002206          TST QVMODE        ;RUNNING IN QV MODE
12802 053124 001411          BEQ 1$           ;BRANCH IF NO
12803 053126 005737 001750          TST N200M        ;PROG STARTED AT LOC 200?
12804 053132 001404          BEQ 2$           ;BRANCH IF NO
12805 053134 005037 002206          CLR QVMODE
12806 053140 000137 037054          JMP COMST
12807 053144 000137 036522          2$: JMP QVST
12808 053150 104400          1$: TYPE
12809 053152 016157          FSHDR2           ;PRINT ENTERING RANDOM MODE HEADER
12810 053154 104400          TYPE
12811 053156 016213          FSHDR3
12812 053160 104400          TYPE
12813 053162 016274          FSHDR4
12814 053164 104400          TYPE
12815 053166 016360          FSHDR5
12817 053170 012737 000001 001660          MOV #1,INCSQ1     ;INITIALIZE BUFFER INITIALIZATION CONSTANTS
12818 053176 012737 000002 001662          MOV #2,INCSQ2
12819 053204 012737 177777 001760          MOV #177777,RANDOM ;SET RANDOM MODE FLAG
12820 053212 012737 072022 072076          MOV #IDUM,INPTBL ;POINT TO DUMMY INPUT TABLE
12821 053220 013737 072076 002134          MOV INPTBL,INPTP
12822 053226 000137 040446          JMP NITE
12823 053232 000137 036710          NFS: JMP DVTST
12824
12825 053236          FC00:           ;FLOW COMMAND = 00 -IGNORE ENTRY.
12826 053236 000137 042010          JMP FCRTN
12827
12828
12829          ;SUBROUTINES TO EXTRACT VARIOUS DATA FIELDS FROM FLOW TABLE ENTRY.
12830
12831 053242          PF1:           ;LOAD PTPTR WITH CONTANTS OF PTP
12832          ; SPECIFIED IN THE PF1 FIELD OF FLOW
12833          ; TABLE ENTRY.
12834 053242 017701 126474          MOV @FLOPTR,R1   ;FLOPTR POINTS TO FLOW TABLE ENTRY
12835 053246 032701 004000          BIT #004000,R1  ;SAVE STRING INSERTED FOR POSSIBLE ERROR PRINTOUT?
12836 053252 001405          BEQ 1$
12837 053254 042701 004000          BIC #004000,R1  ;YES
12838 053260 012737 177777 002472          MOV #177777,SAVSRF ;SET SAVE STRING FLAG
12839 053266 042701 170077          1$: BIC #170077,R1 ;STRIP OFF ALL BUT DESIRED FIELD
12840 053272 006301          ASL R1
12841 053274 006301          ASL R1
12842 053276 000301          SWAB R1         ;ADJUST DESIRED FIELD TO BIT 0
12843 053300 006301          ASL R1
12844 053302 062701 001566          ADD #PTP,R1     ;ADD # IN FIELD TO TOP OF PTP TABLE
12845 053306 011137 002136          MOV (R1),PTPTR ;SAVE CONTENTS OF DESIRED PTP IN PTPTR
12846 053312 000207          RTS PC
12847
12848          PF2:           ;SAME AS PF1 SUBROUTINE EXCEPT FOR FIELD.
12849 053314          MOV @FLOPTR,R1
12850 053320 042701 177700          BIC #177700,R1
12851 053324 006301          ASL R1

```

| | | | | | |
|-------|--------|--------|--------|----------------|---|
| 12852 | 053326 | 062701 | 001566 | ADD #PTP,R1 | |
| 12853 | 053332 | 011137 | 002136 | MOV (R1),PTPTR | |
| 12854 | 053336 | 000207 | | RTS PC | |
| 12855 | | | | | |
| 12856 | 053340 | | | RF1: | ;LOAD R1 WITH ADDRESS OF TEST |
| 12857 | | | | | ; OPERAND SPECIFIED. |
| 12858 | 053340 | 004737 | 053352 | JSR PC,RF1X | ;GET FIELD ONE CONTENTS |
| 12859 | 053344 | 062701 | 003624 | ADD #TRN,R1 | ;ADD IN ADDRESS OF TEST OPERAND TABLE |
| 12860 | 053350 | 000207 | | RTS PC | |
| 12861 | | | | | |
| 12862 | 053352 | | | RF1X: | ;GET FIELD ONE CONTENTS FROM FLOW TABLE ENTRY |
| 12863 | 053352 | 017701 | 126364 | MOV @FLOPTR,R1 | ;FLOPTR POINTS TO FLOW TABLE ENTRY |
| 12864 | 053356 | 042701 | 170777 | BIC #170777,R1 | ;STRIP OFF ALL BUT DESIRED FIELD |
| 12865 | 053362 | 006201 | | ASR R1 | |
| 12866 | 053364 | 000301 | | SWAB R1 | ;RIGHT ADJUST FIELD |
| 12867 | 053366 | 006301 | | ASL R1 | |
| 12868 | 053370 | 000207 | | RTS PC | |
| 12869 | | | | | |
| 12870 | 053372 | | | RF2: | ;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD. |
| 12871 | 053372 | 004737 | 053404 | JSR PC,RF2X | |
| 12872 | 053376 | 062701 | 003624 | ADD #TRN,R1 | |
| 12873 | 053402 | 000207 | | RTS PC | |
| 12874 | | | | | |
| 12875 | 053404 | | | RF2X: | ;SAME AS RF2X SUBROUTINE EXCEPT FOR FIELD. |
| 12876 | 053404 | 017701 | 126332 | MOV @FLOPTR,R1 | |
| 12877 | 053410 | 042701 | 177077 | BIC #177077,R1 | |
| 12878 | 053414 | 006301 | | ASL R1 | |
| 12879 | 053416 | 006301 | | ASL R1 | |
| 12880 | 053420 | 000301 | | SWAB R1 | |
| 12881 | 053422 | 006301 | | ASL R1 | |
| 12882 | 053424 | 000207 | | RTS PC | |
| 12883 | | | | | |
| 12884 | 053426 | | | RF3: | ;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD |
| 12885 | 053426 | 004737 | 053440 | JSR PC,RF3X | |
| 12886 | 053432 | 062701 | 003624 | ADD #TRN,R1 | |
| 12887 | 053436 | 000207 | | RTS PC | |
| 12888 | | | | | |
| 12889 | 053440 | | | RF3X: | ;SAME AS RF1X SUBROUTINE EXCEPT FOR FIELD |
| 12890 | 053440 | 017701 | 126276 | MOV @FLOPTR,R1 | |
| 12891 | 053444 | 042701 | 177707 | BIC #177707,R1 | |
| 12892 | 053450 | 006201 | | ASR R1 | |
| 12893 | 053452 | 006201 | | ASR R1 | |
| 12894 | 053454 | 000207 | | RTS PC | |
| 12895 | | | | | |
| 12896 | 053456 | | | RF4: | ;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD |
| 12897 | 053456 | 004737 | 053470 | JSR PC,RF4X | |
| 12898 | 053462 | 062701 | 003624 | ADD #TRN,R1 | |
| 12899 | 053466 | 000207 | | RTS PC | |
| 12900 | | | | | |
| 12901 | 053470 | | | RF4X: | ;SAME AS RF1X SUBROUTINE EXCEPT FOR FIELD |
| 12902 | 053470 | 017701 | 126246 | MOV @FLOPTR,R1 | |
| 12903 | 053474 | 042701 | 177770 | BIC #177770,R1 | |
| 12904 | 053500 | 006301 | | ASL R1 | |
| 12905 | 053502 | 000207 | | RTS PC | |

| | | | | | | | | | | |
|-------|--------|--------|--------|--------|-------|--|--|--|--|--|
| 12906 | | | | | | | | | | |
| 12907 | 053504 | | | | | | | | | |
| 12908 | | | | | | | | | | |
| 12909 | | | | | | | | | | |
| 12910 | 053504 | 005237 | 002226 | | | | | | | |
| 12911 | 053510 | 022737 | 000001 | 002226 | | | | | | |
| 12912 | 053516 | 001003 | | | | | | | | |
| 12913 | 053520 | 004737 | 053352 | | | | | | | |
| 12914 | 053524 | 000426 | | | | | | | | |
| 12915 | 053526 | 022737 | 000002 | 002226 | 1\$: | | | | | |
| 12916 | 053534 | 001003 | | | | | | | | |
| 12917 | 053536 | 004737 | 053404 | | | | | | | |
| 12918 | 053542 | 000417 | | | | | | | | |
| 12919 | 053544 | 022737 | 000003 | 002226 | 2\$: | | | | | |
| 12920 | 053552 | 001003 | | | | | | | | |
| 12921 | 053554 | 004737 | 053440 | | | | | | | |
| 12922 | 053560 | 000410 | | | | | | | | |
| 12923 | 053562 | 022737 | 000004 | 002226 | 3\$: | | | | | |
| 12924 | 053570 | 001003 | | | | | | | | |
| 12925 | 053572 | 004737 | 053470 | | | | | | | |
| 12926 | 053576 | 000401 | | | | | | | | |
| 12927 | 053600 | 005001 | | | 9\$: | | | | | |
| 12928 | 053602 | 000207 | | | 10\$: | | | | | |
| 12929 | | | | | | | | | | |
| 12930 | 053604 | | | | | | | | | |
| 12931 | | | | | | | | | | |
| 12932 | | | | | | | | | | |
| 12933 | | | | | | | | | | |
| 12934 | 053604 | 005237 | 002226 | | | | | | | |
| 12935 | 053610 | 022737 | 000001 | 002226 | | | | | | |
| 12936 | 053616 | 001003 | | | | | | | | |
| 12937 | 053620 | 004737 | 053340 | | | | | | | |
| 12938 | 053624 | 000427 | | | | | | | | |
| 12939 | 053626 | 022737 | 000002 | 002226 | 1\$: | | | | | |
| 12940 | 053634 | 001003 | | | | | | | | |
| 12941 | 053636 | 004737 | 053372 | | | | | | | |
| 12942 | 053642 | 000420 | | | | | | | | |
| 12943 | 053644 | 022737 | 000003 | 002226 | 2\$: | | | | | |
| 12944 | 053652 | 001003 | | | | | | | | |
| 12945 | 053654 | 004737 | 053426 | | | | | | | |
| 12946 | 053660 | 000411 | | | | | | | | |
| 12947 | 053662 | 022737 | 000004 | 002226 | 3\$: | | | | | |
| 12948 | 053670 | 001003 | | | | | | | | |
| 12949 | 053672 | 004737 | 053456 | | | | | | | |
| 12950 | 053676 | 000402 | | | | | | | | |
| 12951 | 053700 | 012701 | 003624 | | 9\$: | | | | | |
| 12952 | 053704 | 000207 | | | 10\$: | | | | | |
| 12953 | | | | | | | | | | |
| 12954 | 053706 | | | | | | | | | |
| 12955 | | | | | | | | | | |
| 12956 | | | | | | | | | | |
| 12957 | | | | | | | | | | |
| 12958 | | | | | | | | | | |
| 12959 | 053706 | 004737 | 053242 | | | | | | | |

RFNX:

; SUBROUTINE TO RETURN CONTENTS OF FLOW
; TABLE ENTRY FIELD POINTED TO BY THE
; CONTENTS OF NXFLD+1.

; FIELD REQUESTED = 1?
; BRANCH IF NO
; GET CONTENTS OF FIELD ONE.
; EXIT
; FIELD REQUESTED=2?
; BRANCH IF NO
; GET CONTENTS OF FIELD 2
; EXIT
; FIELD REQUESTED = 3?
; BRANCH IF NO
; GET CONTENTS OF FIELD 3
; EXIT
; FIELD REQUESTED = 4?
; BRANCH IF NO
; GET CONTENTS OF FIELD 4
; EXIT
; FIELD REQUESTED INVALID - RETURN 0.

RFN:

; SUBROUTINE TO RETURN CONTENTS OF FLOW
; TABLE ENTRY FIELD PLUS TEST OPERAND TABLE
; OFFSET POINTED TO BY THE
; CONTENTS OF NXFLD+1.

; FIELD REQUESTED = 1?
; BRANCH IF NO
; GET CONTENTS OF FIELD ONE PLUS TRN OFFSET.
; EXIT
; FIELD REQUESTED=2?
; BRANCH IF NO
; GET CONTENTS OF FIELD 2 PLUS TRN OFFSET.
; EXIT
; FIELD REQUESTED = 3?
; BRANCH IF NO
; GET CONTENTS OF FIELD 3 PLUS TRN OFFSET.
; EXIT
; FIELD REQUESTED = 4?
; BRANCH IF NO
; GET CONTENTS OF FIELD 4 PLUS TRN OFFSET.
; EXIT
; FIELD REQUESTED INVALID - RETURN #TRN.

FC15:

; FLOW COMMAND = 15 - SETUP SPECIAL HANDLING WORD
; BIT 0 = 1 MEANS SKIP 04 FLOW COMMANDS
; BIT 1 = 1 MEANS 07 FLOW COMMAND IGNORE BIT 15
; OF STRING LENGTH
; BIT 2 = 1 MEANS DON'T CHECK BUFFER RESULTS
; FORM PARAMETER TABLE POINTER TO SPECIAL HANDLING REQUES

| | | | | | | |
|-------|--------|--------|--------|--------|-------------------|--|
| 12960 | 053712 | 017737 | 126220 | 002140 | MOV @PTPTR,SPHAND | ;COPY SPECIAL HANDLING REQUEST INTO SPECIAL HANDLING WOR |
| 12961 | 053720 | 000137 | 042010 | | JMP FCRTN | |
| 12962 | | | | | | |

```

12964          .SBTTL          SETUP MACHINE DEPENDENT CONSTANTS
12965          :
12966          :SIZE FOR PROCESSOR TYPE - SETUP MACHINE DEPENDENT CCNSTANTS BASED UPON RESULT.
12967          :
12968 053724      SIZEPT:
12969 053724 012737 054154 000010      MOV #3$,@#RESVEC          ;SETUP RESERVED INST TRAP VECTOR
12970          :
12971 053732 005037 000012      CLR @#RESVEC+2          ; IN CASE MACHINE UNDER TEST DOES NOT HAVE MFPT
12972 053736 000007      MFPT          ;IS THIS AN 11/74?
12973 053740 123700 001706      CMPB EL74,R0
12974 053744 001020      BNE 1$          ;BRANCH IF NOT AN 11/74
12975 053746 012737 177777 002156      MOV #177777,MMFLG      ;SET MEM MANAGEMENT FLAG SIGNALING THAT SYS
12976          :
12977 053754 012737 000003 002164      MOV #3,NMODES          ; UNDER TEST HAS 11/70 TYPE MEM MGMT
12978 053762 012737 177777 002060      MOV #177777,TWOSET    ;PROC UNDER TEST HAS 3 MODES (K,S,U)
12979 053770 012737 000002 002212      MOV #2,PTQV           ;PROCESSOR UNDER TEST HAS 2 REGISTER SETS
12980          :
12981          :
12982 053776 012737 000003 003032      MOV #3,LTCDLY         ;INPUT TABLE ENTRY TYPE WORD BIT 1
12983          :
12984          : IDENTIFIES TABLE RUN IN QV MODE FOR 11/74
12985          : (1=RUN,0=SKIP)
12986          :
12987 054004 000543          BR 4$          ;INITIALIZE COUNTER USED BY LINE TIME CLOCK
12988 054006 123700 001710      1$: CMPB EL44,R0          ; ROUTINES - COUNT IS A MEASURE OF
12989 054012 001017          BNE 2$          ; TIME REMAINING BEFORE INTERRUPT FROM
12990 054014 012737 177777 002156      MOV #177777,MMFLG    ; LTC WHEN STARTING EXECUTION OF CIS
12991 054022 012737 000003 002164      MOV #3,NMODES          ; INSTRUCTION UNDER TEST.
12992 054030 005037 002060      11$: CLR TWOSET          ;IS THIS AN 11/44?
12993 054034 012737 000002 002212      MOV #2,PTQV           ;BRANCH IF NO
12994          :
12995 054042 012737 000013 003032      MOV #13,LTCDLY        ;SET MEM MGMT FLAG
12996 054050 000521          BR 4$          ;PROC UNDER TEST HAS 3 MODES
12997 054052 005037 002156      2$: CLR MMFLG          ;INITIALIZE FOR SINGLE REGISTER SET
12998 054056 012737 000240 060110      MOV #NOP,PCIS2        ;INPUT TABLE ENTRY TYPE WORD BIT 2
12999 054064 012737 000240 060112      MOV #NOP,PCIS2+2      ; IDENTIFIES TABLE RUN IN QV MODE FOR 11/44
13000 054072 012737 000240 060134      MOV #NOP,PCIS1        ;INITIALIZE COUNTER USED FOR LTC
13001 054100 012737 000240 060136      MOV #NOP,PCIS1+2
13002 054106 012737 000406 062044      MOV #406,LTCIS
13003 054114 012737 000240 056566      MOV #NOP,ILLSER
13004 054122 012737 000240 056570      MOV #NOP,ILLSER+2
13005 054130 012737 000240 056446      MOV #NOP,HLTSER
13006 054136 012737 000240 056450      MOV #NOP,HLTSER+2
13007 054144 012737 000002 002164      MOV #2,NMODES
13008 054152 000726          BR 11$
13009          :
13010          :MACHINE TYPE UNKNOWN - DEFUAULT TO SINGLE
13011          : REG SET ,NO MEM MGMT, 2 PROC MODES (K & U)
13012 054154 005726          3$: TST (SP)+
13013 054156 005726          TST (SP)+
13014 054160 012737 054226 000004      MOV #33$,@#ERRVEC    ; AND A LTC DELAY OF 13
13015 054166 005737 172340          TST @#KIFARO          ;(NEW MACHINES MAY REQUIRE CHANGES HERE)
13016 054172 005737 177640          TST @#UIPARO          ;FIX UP STACK
13017 054176 012737 054234 000004      MOV #34$,@#ERRVEC    ;IS THIS AN 11/34?
          :
          : IF SO NO TRAP HERE
          : IF SO NO TRAP HERE

```

| | | | | | | | | |
|-------|--------|--------|--------|--------|-------|----------------------|--|--|
| 13018 | 054204 | 005737 | 172240 | | | TST @#SIPARO | | ;IF SO, TRAP HERE |
| 13019 | 054210 | 000720 | | | | BR 2\$ | | |
| 13020 | 054212 | 012737 | 054242 | 000004 | 5\$: | MOV #35\$,@#ERRVEC | | ;IF SO, TRAP HERE |
| 13021 | 054220 | 005737 | 172360 | | | TST @#KDPARO | | |
| 13022 | 054224 | 000712 | | | | BR 2\$ | | |
| 13023 | 054226 | 005726 | | | 33\$: | TST (SP)+ | | ;FIX UP STACK |
| 13024 | 054230 | 005726 | | | | TST (SP)+ | | |
| 13025 | 054232 | 000707 | | | | BR 2\$ | | |
| 13026 | 054234 | 005726 | | | 34\$: | TST (SP)+ | | ;FIX UP STACK |
| 13027 | 054236 | 005726 | | | | TST (SP)+ | | |
| 13028 | 054240 | 000764 | | | | BR 5\$ | | |
| 13029 | 054242 | 005726 | | | 35\$: | TST (SP)+ | | ;FIX UP STACK |
| 13030 | 054244 | 005726 | | | | TST (SP)+ | | |
| 13031 | 054246 | 104400 | | | | TYPE | | ;TYPE '11/34 TYPE MEMORY MANAGEMENT ON SYS UNDER TEST' |
| 13032 | 054250 | 016625 | | | | MPT34 | | |
| 13033 | 054252 | 012737 | 000002 | 002164 | | MOV #2,NMODES | | ;PROC = 11/34 TYPE |
| 13034 | 054260 | 012737 | 177777 | 002156 | | MOV #177777,MMFLG | | |
| 13035 | 054266 | 005037 | 002060 | | | CLR TWOSET | | |
| 13036 | 054272 | 012737 | 000002 | 002212 | | MOV #2,PTQV | | |
| 13037 | 054300 | 012737 | 177777 | 002154 | | MOV #177777,PT34 | | |
| 13038 | 054306 | 012737 | 000013 | 003032 | | MOV #13,LTCDLY | | |
| 13039 | 054314 | 012737 | 056566 | 000010 | 4\$: | MOV #ILLSER,@#RESVEC | | ;RESTORE RESERVED INST TRAP CATCHER |
| 13040 | 054322 | 012737 | 056446 | 000004 | | MOV #HLTSER,@#ERRVEC | | ;RESTORE HALT SERVICE TRAP CATCHER |
| 13041 | 054330 | 000207 | | | | RTS PC | | |

```

13043      .SBTTL          REGISTER SET SELECTION ROUTINES
13044
13045      ;
13046      ;IF PROCESSOR UNDER TEST HAS 2 REGISTER SETS
13047      ;SELECT REGISTER SET TO BE USED BASED ON LEAST SIGNIFICANT BIT OF
13048      ;TEST # (TOTTC), AND LOAD REGISTER SET WHICH WAS NOT
13049      ;SELECTED WITH A FIXED PATTERN <NOTREG>.
13050      ;
13051
13052      SELREG:
13053      054332 005737 002060      TST TWOSET          ;DOES MACHINE UNDER TEST HAVE 2 REG SETS?
13054      054336 001506          BEQ 1$              ;BRANCH IF NO
13055      054340 032737 000001 001420  BIT #1,TOTTC        ;PROCESSOR IS AN 11/74 - HAS 2 REG SETS;USE
13056      ; LEAST SIGNIF. BIT OF TEST COUNT TO LOAD
13057      ; PSW BIT 11 (REG SET BIT).
13058      054346 001041          BNE 2$              ;BRANCH TO USE REGISTER SET 1.
13059      054350 052777 004000 125306  BIS #4000,@TPSW     ;CIS INST WILL BE TESTED USING GPR SET 0.
13060      054356 042737 001000 002132  BIC #1000,FATAL    ;INDICATE GPR SET 0 IN FATAL ERROR WORD
13061      ;LOAD SET 1 WITH PATTERN IN NOTREG.
13062      054364 013700 001704      MOV NOTREG,R0
13063      054370 013701 001704      MOV NOTREG,R1
13064      054374 013702 001704      MOV NOTREG,R2
13065      054400 013703 001704      MOV NOTREG,R3
13066      054404 013704 001704      MOV NOTREG,R4
13067      054410 013705 001704      MOV NOTREG,R5
13068      054414 042777 004000 125242  BIC #4000,@TPSW     ;SET REGISTER SET TO 0
13069      054422 042777 004000 126062  BIC #4000,@PCLK1P   ;ASSURE CORRECT REGISTER USAGE ON INTERRUPT
13070      054430 042777 004000 126362  BIC #4000,@LTCIP
13071      054436 042737 004000 000006  BIC #4000,@#6
13072      054444 005037 002052      CLR REGSET
13073      054450 000441          BR 1$              ;SET REG SET INDICATOR TO 0
13074      054452 042777 004000 125204  2$: BIC #4000,@TPSW     ;BRANCH TO RETURN
13075      054460 052737 001000 002132  BIS #1000,FATAL    ;CIS INST WILL BE TESTED USING GPR SET 1.
13076      ;INDICATE GPR SET 1 IN FATAL ERROR WORD
13077      ;LOAD SET 0 WITH PATTERN IN NOTREG.
13077      054466 013700 001704      MOV NOTREG,R0
13078      054472 013701 001704      MOV NOTREG,R1
13079      054476 013702 001704      MOV NOTREG,R2
13080      054502 013703 001704      MOV NOTREG,R3
13081      054506 013704 001704      MOV NOTREG,R4
13082      054512 013705 001704      MOV NOTREG,R5
13083      054516 052777 004000 125140  BIS #4000,@TPSW     ;SET REGISTER SET TO 1
13084      054524 052777 004000 125760  BIS #4000,@PCLK1P   ;ASSURE CORRECT REGISTER USAGE ON INTERRUPT
13085      054532 052777 004000 126260  BIS #4000,@LTCIP
13086      054540 052737 004000 000006  BIS #4000,@#6
13087      054546 012737 000001 002052  MOV #1,REGSET      ;SET REG SET INDICATOR TO 1
13088      054554 000207      1$: RTS PC
13089
13090      ;
13091      ;ROUTINE TO VERIFY THAT REGISTER SET WHICH WAS NOT SELECTED
13092      ;(PROVIDED PROCESSOR UNDER TEST HAS 2) DID NOT GET CHANGED, AND
13093      ;SWITCH TO REGISTER SET 0.
13094      ;
13095
13096      054556      CKUREG:

```



```

13097 054556 005737 002060          TST TWOSET          ;PROCESSOR UNDER TEST HAVE TWO REGISTER SETS?
13098 054562 001435          BEQ 1$              ;BRANCH IF NO
13099 054564 005737 002052          TST REGSET          ;DETERMINE WHICH REG SET WAS IN USE
13100 054570 001426          BEQ 2$              ;BRANCH IF REG SET 0 WAS USED
13101 054572 042777 004000 125064    BIC #4000,@TPSW     ;VERIFY CONTENTS OF REG SET 0
13102 054600 020037 001704          CMP R0,NOTREG
13103 054604 001030          BNE 4$
13104 054606 020137 001704          CMP R1,NOTREG
13105 054612 001025          BNE 4$
13106 054614 020237 001704          CMP R2,NOTREG
13107 054620 001022          BNE 4$
13108 054622 020337 001704          CMP R3,NOTREG
13109 054626 001017          BNE 4$
13110 054630 020437 001704          CMP R4,NOTREG
13111 054634 001014          BNE 4$
13112 054636 020537 001704          CMP R5,NOTREG
13113 054642 001011          BNE 4$
13114 054644 000404          BR 1$
13115 054646 052777 004000 125010 2$:  BIS #4000,@TPSW     ;VERIFY CONTENTS OF REG SET 1
13116 054654 000751          BR 3$
13117 054656 042777 004000 125000 1$:  BIC #4000,@TPSW     ;SET REG SET TO ZERO
13118 054664 000207          RTS PC              ;RETURN
13119 054666          4$: PRINTB #HLTMSG
      (6) 054666 012746 012163          MOV #HLTMSG,-(SP)
      (3) 054672 010600          MOV SP,R0
      (4) 054674 004737 065304          JSR PC,FPRINT
13120 054700 012737 054762 002170    MOV #100$,HLTLOC
13121 054706 004737 054766          JSR PC,IDINFO      ;IDENTIFY FAILING INST
13122 054712          PRINTB #FORM38,REGSET
      (7) 054712 013746 002052          MOV REGSET,-(SP)
      (6) 054716 012746 014341          MOV #FORM38,-(SP)
      (3) 054722 010600          MOV SP,R0
      (4) 054724 004737 065304          JSR PC,FPRINT
13123 054730          PRINTB #FORM39,NOTREG,R0,R1,R2,R3,R4,R5
      (13) 054730 010546          MOV R5,-(SP)
      (12) 054732 010446          MOV R4,-(SP)
      (11) 054734 010346          MOV R3,-(SP)
      (10) 054736 010246          MOV R2,-(SP)
      (9) 054740 010146          MOV R1,-(SP)
      (8) 054742 010046          MOV R0,-(SP)
      (7) 054744 013746 001704          MOV NOTREG,-(SP)
      (6) 054750 012746 014420          MOV #FORM39,-(SP)
      (3) 054754 010600          MOV SP,R0
      (4) 054756 004737 065304          JSR PC,FPRINT
13124 054762 000000          100$: HALT          ;REGISTER SET ERROR;PRESS CONTINUE SWITCH TO CONTINUE
13125          ; TESTING
13126 054764 000734          BR 1$
13127
13128 054766          IDINFO: PRINTB #TRPINF,HLTLOC,MODE,DEN,TINST,TOTTCH,TOTTC
      (12) 054766 013746 001420          MOV TOTTC,-(SP)
      (11) 054772 013746 001416          MOV TOTTCH,-(SP)
      (10) 054776 013746 047464          MOV TINST,-(SP)
      (9) 055002 013746 002162          MOV DEN,-(SP)
      (8) 055006 013746 002160          MOV MODE,-(SP)

```

REGISTER SET SELECTION ROUTINES

| | | | | | |
|-------|--------|--------|--------|-----|---------------|
| (7) | 055012 | 013746 | 002170 | MOV | HLTLOC,-(SP) |
| (6) | 055016 | 012746 | 012300 | MOV | #TRPINF,-(SP) |
| (3) | 055022 | 010600 | | MOV | SP,R0 |
| (4) | 055024 | 004737 | 065304 | JSR | PC,FPRINT |
| 13129 | 055030 | 000207 | | RTS | PC |
| 13130 | | | | | |

```
13132          .SBTTL          MEMORY MANAGEMENT SUBROUTINES
13133
13134          :MEMORY MANAGEMENT SUBROUTINES
13135          :
13136          :
13137          :
13138          :
13139          :SETUP PAR'S
13140          SETPAR:          ;SETUP PAR'S FOR USER,SUPV, AND KERNEL I & D SPACES
13141          055032 005737 002156          TST MMFLG          ;DOES SYSTEM UNDER TEST HAVE MEMORY MANAGEMENT?
13142          055036 001002          BNE 2$          ;BRANCH IF YES
13143          055040 000137 055516          JMP 1$
13144          055044 005737 002154          2$: TST PT34          ;IS THIS AN 11/34 TYPE PROCESSOR
13145          ; (I.E. K, U MODES AND 18 BIT MEM MGMT)
13146          055050 001142          BNE 3$          ;BRANCH IF YES
13147          055052 005037 172516          CLR @MMR3          ;CLEAR OUT D-SPACE ENABLES
13148          055056 012737 000000 177660          MOV #0,@UDPAR0          ;SETUP USER D PAR'S
13149          055064 012737 000200 177662          MOV #200,@UDPAR1
13150          055072 012737 000400 177664          MOV #400,@UDPAR2
13151          055100 012737 000600 177666          MOV #600,@UDPAR3
13152          055106 012737 001000 177670          MOV #1000,@UDPAR4
13153          055114 012737 001200 177672          MOV #1200,@UDPAR5
13154          055122 012737 001400 177674          MOV #1400,@UDPAR6
13155          055130 012737 177600 177676          MOV #177600,@UDPAR7
13156
13157          055136 012737 000000 172240          MOV #0,@SIPAR0          ;SETUP SUPERVISOR I PAR'S
13158          055144 012737 000200 172242          MOV #200,@SIPAR1
13159          055152 012737 000400 172244          MOV #400,@SIPAR2
13160          055160 012737 000600 172246          MOV #600,@SIPAR3
13161          055166 012737 001000 172250          MOV #1000,@SIPAR4
13162          055174 012737 001200 172252          MOV #1200,@SIPAR5
13163          055202 012737 001400 172254          MOV #1400,@SIPAR6
13164          055210 012737 177600 172256          MOV #177600,@SIPAR7
13165
13166          055216 012737 000000 172260          MOV #0,@SDPAR0          ;SETUP SUPERVISOR D PAR'S
13167          055224 012737 000200 172262          MOV #200,@SDPAR1
13168          055232 012737 000400 172264          MOV #400,@SDPAR2
13169          055240 012737 000600 172266          MOV #600,@SDPAR3
13170          055246 012737 001000 172270          MOV #1000,@SDPAR4
13171          055254 012737 001200 172272          MOV #1200,@SDPAR5
13172          055262 012737 001400 172274          MOV #1400,@SDPAR6
13173          055270 012737 177600 172276          MOV #177600,@SDPAR7
13174
13175          055276 012737 000000 172360          MOV #0,@KDPAR0          ;SETUP KERNEL D PAR'S
13176          055304 012737 000200 172362          MOV #200,@KDPAR1
13177          055312 012737 000400 172364          MOV #400,@KDPAR2
13178          055320 012737 000600 172366          MOV #600,@KDPAR3
13179          055326 012737 001000 172370          MOV #1000,@KDPAR4
13180          055334 012737 001200 172372          MOV #1200,@KDPAR5
13181          055342 012737 001400 172374          MOV #1400,@KDPAR6
13182          055350 012737 177600 172376          MOV #177600,@KDPAR7
13183
13184          055356 012737 000000 177640          3$: MOV #0,@UIPAR0          ;SETUP USER I PAGE ADDRESS REGISTERS
13185          055364 012737 000200 177642          MOV #200,@UIPAR1
```

```

13186 055372 012737 000400 177644      MOV #400,@#UIPAR2
13187 055400 012737 000600 177646      MOV #600,@#UIPAR3
13188 055406 012737 001000 177650      MOV #1000,@#UIPAR4
13189 055414 012737 001200 177652      MOV #1200,@#UIPAR5
13190 055422 012737 001400 177654      MOV #1400,@#UIPAR6
13191 055430 012737 177600 177656      MOV #177600,@#UIPAR7
13192
13193 055436 012737 000000 172340      MOV #0,@#KIPAR0          ;SETUP KERNEL I PAR'S
13194 055444 012737 000200 172342      MOV #200,@#KIPAR1
13195 055452 012737 000400 172344      MOV #400,@#KIPAR2
13196 055460 012737 000600 172346      MOV #600,@#KIPAR3
13197 055466 012737 001000 172350      MOV #1000,@#KIPAR4
13198 055474 012737 001200 172352      MOV #1200,@#KIPAR5
13199 055502 012737 001400 172354      MOV #1400,@#KIPAR6
13200 055510 012737 177600 172356      MOV #177600,@#KIPAR7
13201
13202 055516 000207      1$:      RTS PC
13203
13204      ;
13205      ;SELECT MODE AND D-SPACE ENABLE/DISABLE
13206      ;
13207      ;BOTH MODE AND D-SPACE ENABLE ARE SELECTED RANDOMLY FOR EACH TEST
13208      ;IF EXECUTING IN RANDOM MODE OR IN A NORMAL FIELD SERVICE TYPE RUN.
13209      ;IF IN A DESIGN VERIFICATION TYPE RUN (ST @ 204) AND NOT RANDOM MODE
13210      ;THEN THESE VARIABLES WERE SET VIA OPERATOR DIAGLOG ABOVE.
13211      ;REGARDLESS OF THE RUN TYPE, IF MEMORY MANAGEMENT IS AVAILABLE
13212      ;(MMFLG=NONZERO) THEN MMR3 IS LOADED TO PROPER D-SPACE STATE.
13213      ;
13214      SELMD:
13215 055520 005737 002074      TST FSRUN                ;NORMAL FFIELD SERVICE TYPE RUN?
13216 055524 001003      BNE 10$                  ;BRANCH IF YES
13217 055526 005737 001760      TST RANDOM                ;RANDOM TESTING?
13218 055532 001443      BEQ 11$                  ;BRANCH IF NO
13219 055534 022737 000001 002164 10$:      CMP #1,NMODES            ;DOES SYSTEM UNDER TEST HAVE MORE THAN ONE
13220      ; PROCESSOR MODE?
13221 055542 001003      BNE 2$                   ;BRANCH IF YES
13222 055544 005037 002160      CLR MODE
13223 055550 000422      BR 4$
13224 055552 004737 063464      2$:      JSR PC,RN                ;GENERATE A RANDOM #
13225 055556 042700 177774      BIC #177774,R0           ;USE BITS 1 & 0 TO SELECT MODE
13226      ; (I.E 00=KERNEL,01=SUPV,11=USER)
13227 055562 022700 000002      CMP #2,R0                ;ILLEGAL MODE (10)?
13228 055566 001771      BEQ 2$                   ;YES - TRY AGAIN
13229 055570 022737 000002 002164      CMP #2,NMODES            ;DOES SYSTEM UNDER TEST HAVE ONLY 2 PROC MODES?
13230 055576 001005      BNE 6$                   ;BRANCH IF NO
13231 055600 022700 000001      CMP #1,R0                ;ON 2 MODE MACHINE MODES ASSUMED TO BE
13232      ; KERNEL & USER
13233 055604 001002      BNE 6$                   ;BRANCH IF MODE IS LEGAL
13234 055606 052700 000002      BIS #2,R0                ;IF RANDOM MODE = SUPV THEN SWITCH IT TO USER
13235 055612 010037 002160      6$:      MOV R0,MODE              ;MODE VALID - SET INTO MODE WORD
13236 055616 005037 002162      4$:      CLR DEN
13237 055622 004737 063464      JSR PC,RN                ;GENERATE A RANDOM #
13238 055626 032700 000001      BIT #1,R0                ;USE BIT 0 OF RANDOM NUMBER TO SELECT D ENABLE
13239 055632 001403      BEQ 11$                  ;BRANCH TO DISABLE D SPACE

```

```

13240 055634 012737 177777 002162      MOV #177777,DEN      ;SET D ENABLE FLAG
13241 055642 005737 002156      11$:  TST MMFLG          ;MEMORY MGMT AVAILABLE?
13242 055646 001435              BEQ 1$              ;NO - EXIT
13243 055650 005737 002154      TST PT34           ;11/34 TYPE MEM MGMT?
13244 055654 001403              BEQ 7$              ;BRANCH IF NO
13245 055656 005037 002162      CLR DEN
13246 055662 000427              BR 1$
13247 055664 005037 172516      7$:  CLR @MMR3        ;DISABLE ALL D-SPACES
13248 055670 005737 002162      TST DEN           ;ENABLE D SPACE IN MMR3?
13249 055674 001422              BEQ 1$              ;BRANCH IF NO
13250 055676 005737 002160      13$:  TST MODE         ;MODE SELECTED = KERNEL?
13251 055702 001004              BNE 3$              ;BRANCH IF NO
13252 055704 052737 000004 172516  BIS #4,@MMR3        ;ENABLE KERNEL D SPACE
13253 055712 000413              BR 1$
13254 055714 022737 000003 002160  3$:  CMP #3,MODE        ;MODE SELECTED = USER
13255 055722 001004              BNE 5$              ;BRANCH IF NO
13256 055724 052737 000001 172516  BIS #1,@MMR3        ;ENABLE USER D SPACE
13257 055732 000403              BR 1$
13258 055734 052737 000002 172516  5$:  BIS #2,@MMR3        ;MODE SLECTED = SUPERVISOR
13259 055742 000207              1$:  RTS PC           ;ENABLE SUPERVISOR D SPACE
13260
13261
13262      ; SETUP PDR'S
13263      ;
13264 055744      ;SETPDR:
13265 055744 005737 002154      TST PT34           ;11/34 TYPE MEM MGMT?
13266 055750 001022              BNE 8$              ;BRANCH IF YES
13267
13268      ;FIRST CLEAR ALL PDR'S (ABORT ALL ACCESSES)
13269 055752 012700 172200      2$:  MOV #SIPDR0,R0
13270 055756 005020              CLR (R0)+           ;CLEAR SUPERVISOR PDR'S
13271 055760 020027 172236      CMP R0,#SDPDR7
13272 055764 101774              BLOS 2$
13273
13274 055766 012700 177620      1$:  MOV #UDPDR0,R0
13275 055772 005020              CLR (R0)+           ;CLEAR USER D-PDR'S
13276 055774 020027 177636      CMP R0,#UDPDR7
13277 056000 101774              BLOS 1$
13278
13279 056002 012700 172320      99$:  MOV #KDPDR0,R0
13280 056006 005020              CLR (R0)+           ;CLEAR KERNEL D-PDR'S
13281 056010 020027 172336      CMP R0,#KDPDR7
13282 056014 101774              BLOS 99$
13283
13284 056016 012700 177600      8$:  MOV #UIPDR0,R0
13285 056022 005020              CLR (R0)+           ;CLEAR USER I-PDR'S
13286 056024 020027 177616      9$:  CMP R0,#UIPDR7
13287 056030 101774              BLOS 9$
13288
13289 056032 012700 172300      3$:  MOV #KIPDR0,R0
13290 056036 005020              CLR (R0)+           ;CLEAR KERNEL PDR'S
13291 056040 020027 172316      CMP R0,#KIPDR7
13292 056044 101774              BLOS 3$
13293

```

```

13294                                     ;SETUP KERNEL MODE PDR'S
13295
13296 056046 012737 077406 172316      MOV #77406,@#KIPDR7      ;ALLOW R/W ACCESS OF I/O PAGE
13298 056054 012737 077402 172310      MOV #77402,@#KIPDR4      ; AND R ONLY ACCESS OF PHYSICAL ADDRESS
13299 056062 012737 077402 172306      MOV #77402,@#KIPDR3      ; 20 TO 120K
13301 056070 012737 077402 172304      MOV #77402,@#KIPDR2
13302 056076 012737 077402 172302      MOV #77402,@#KIPDR1
13303 056104 005737 002160              TST MODE                ;IS MODE = KERNEL?
13304 056110 001404                      BEQ 31$                  ;BRANCH IF YES
13305 056112 012737 077406 172300      MOV #77406,@#KIPDR0
13306 056120 000403                      BR 32$
13307 056122 012737 077402 172300 31$:  MOV #77402,@#KIPDR0
13308 056130 005737 002160 32$:      TST MODE                ;IF MODE= KERNEL & D-SPACE IS DISABLED
13309 056134 001021                      BNE 5$                  ; THEN ALLOW R/W OF STACK & TEST BUFFER AREA
13310 056136 005737 002162              TST DEN
13311 056142 001004                      BNE 4$
13313 056144 012737 077406 172312      MOV #77406,@#KIPDR5
13318 056152 000455                      BR 11$
13319 056154 012737 077406 172336 4$:  MOV #77406,@#KDPDR7      ;IF MODE IS KERNEL & D-SPACE IS ENABLED
13321 056162 012737 077406 172332      MOV #77406,@#KDPDR5      ; THEN SETUP KERNEL D-SPACE PDRS
13326 056170 012737 077402 172320      MOV #77402,@#KDPDR0      ;ALLOW R/W ACCESS OF I/O PAGE AND TEST BUFFER
13327                                     ; AREA; R-ONLY ACCESS OF PHYS 0-20K
13331 056176 000443                      BR 11$
13332 056200 022737 000001 002160 5$:  CMP #1,MODE              ;SETUP SUPERVISOR MODE PDR'S
13333 056206 001016                      BNE 7$                  ;BRANCH IF TEST MODE IS NOT SUPERVISOR
13334 056210 012737 077402 172204      MOV #77402,@#SIPDR2      ;ALLOW R ONLY ACCESS OF SPACE INCLUDING
13335                                     ; CIS INST TO BE EXECUTED
13339 056216 005737 002162              TST DEN                ;IF D-SPACE IS NOT ENABLED ALLOW
13340 056222 001004                      BNE 6$                  ; R/W ACCESS OF TEST BUFFER AREA IN SUPERVISOR
13341                                     ; I-SPACE
13343 056224 012737 077406 172212      MOV #77406,@#SIPDR5
13348 056232 000425                      BR 11$
13350 056234 012737 077406 172232 6$:  MOV #77406,@#SDPDR5      ;D-SPACE IS ENABLED; ALLOW R/W ACCESS
13355 056242 000421                      BR 11$                  ; TO TEST BUFFER AREA IN SUPERVISOR D-SPACE
13356 056244 022737 000003 002160 7$:  CMP #3,MODE              ;SETUP USER MODE PDR'S
13357 056252 001015                      BNE 11$                 ;BRANCH IF TEST MODE IS NOT USER
13358 056254 012737 077402 177604      MOV #77402,@#UIPDR2      ;ALLOW R ONLY ACCESS OF SPACE INCLUDING
13359                                     ; CIS INST TO BE EXECUTED
13363 056262 005737 002162              TST DEN                ;IF D-SPACE IS NOT ENABLED ALLOW R/W
13364 056266 001004                      BNE 10$                 ; ACCESS OF TEST BUFFER AREA IN USER
13366 056270 012737 077406 177612      MOV #77406,@#UIPDR5      ; I SPACE
13371 056276 000403                      BR 11$
13373 056300 012737 077406 177632 10$: MOV #77406,@#UDPDR5      ;D-SPACE IS ENABLED; ALLOW R/W ACCESS
13378                                     ; TO TEST BUFFER AREA IN USER D-SPACE
13379 056306 005737 001760 11$:      TST RANDOM              ;ASSURE THAT XLATION BUFFER (MOVTC) IS IN READABLE SPACE
13380                                     ;IN RANDOM MODE?
13381 056312 001454                      BEQ 12$                 ;BRANCH IF NO
13382 056314 022737 076032 047464      CMP #76032,TINST         ;IS INST UNDER TEST = MOVTC?
13383 056322 001404                      BEQ 20$                 ;BRANCH IF YES
13384 056324 022737 076132 047464      CMP #76132,TINST
13385 056332 001044                      BNE 12$
13386 056334 005737 002162 20$:      TST DEN                ;IS D-SPACE ENABLED?
13387 056340 001017                      BNE 13$                 ;BRANCH IF YES
13388 056342 005737 002160              TST MODE                ;IS MODE = KERNEL?

```

```

13389 056346 001436          BEQ 12$          ;BRANCH IF YES - NO PDR CHANGE REQUIRED FOR XLATION BUFF
13390 056350 022737 000001 002160    CMP #1,MODE     ;IS MODE = SUPERVISOR?
13391 056356 001004          BNE 14$          ;BRANCH IF NO
13392 056360 012737 077402 172202    MOV #77402,@#SIPDR1 ;ALLOW R-ONLY ACCESS TO XLATION BUFFER
13393 056366 000426          BR 12$
13394 056370 012737 077402 177602 14$: MOV #77402,@#UIPDR1 ;USER MODE- ALLOW R-ONLY ACCESS TO XLATION BUFFER
13395 056376 000422          BR 12$
13396 056400 005737 002160          13$: TST MODE
13397 056404 001004          BNE 15$
13398 056406 012737 077402 172322          MOV #77402,@#KDPDR1 ;KERNEL MODE,D-EN - ALLOW R-ONLY ACCESS TO XLATI
13399 056414 000413          BR 12$
13400 056416 022737 000001 002160 15$: CMP #1,MODE
13401 056424 001004          BNE 16$
13402 056426 012737 077402 172222          MOV #77402,@#SDPDR1 ;SUPERVISOR MODE,D-EN - ALLOW READ OF XLATION BU
13403 056434 000403          BR 12$
13404 056436 012737 077402 177622 16$: MOV #77402,@#UDPDR1 ;USER MODE,D-EN - ALLOW R-ONLY ACCESS OF XLATION
13405 056444 000207          12$: RTS PC
13406
13407
13408
13409
13410          ;HALT SERVICE ROUTINE (USED FOR TRAPS CAUSED BY HALTS IN SUPERVISOR OR USER MODE
13411          ;
13412 056446 005037 177572          HLTSER: CLR @#MMRO ;TURN OFF MEM MGMT - OVERWRITTEN WITH NOPS
13413          ; IF NO MEM MGMT ON SYSTEM (REF SIZEPT ROUTINE)
13414 056452 021627 047504          CMP (SP),#TINRET+2 ;WAS HALT AT CIS INST RETURN LOC?
13415 056456 001434          BEQ 1$          ;BRANCH IF YES
13416 056460 011637 002166          MOV (SP),TRPLOC ;GET TRAP LOCATION
13417 056464 162737 000002 002166    SUB #2,TRPLOC
13418 056472          PRINTB #TRAP4 ;PRINT - TRAP TO LOC 4
(6) 056472 012746 012126          MOV #TRAP4,-(SP)
(3) 056476 010600          MOV SP,R0
(4) 056500 004737 065304          JSR PC,FPRINT
13419 056504          PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTC,TOTTC ;PRINT - ADDITIONAL TRAP
(12) 056504 013746 001420          MOV TOTTC,-(SP)
(11) 056510 013746 001416          MOV TOTTC,-(SP)
(10) 056514 013746 047464          MOV TINST,-(SP)
(9) 056520 013746 002162          MOV DEN,-(SP)
(8) 056524 013746 002160          MOV MODE,-(SP)
(7) 056530 013746 002166          MOV TRPLOC,-(SP)
(6) 056534 012746 012300          MOV #TRPINF,-(SP)
(3) 056540 010600          MOV SP,R0
(4) 056542 004737 065304          JSR PC,FPRINT
13420 056546 000000          HALT
13421 056550 016637 000002 003662 1$: MOV 2(SP),TCCR ;SAVE CIS INST RETURN CONDITION CODES
13422 056556 005726          TST (SP)+
13423 056560 005726          TST (SP)+
13424 056562 000137 047526          JMP SUHRET
13425
13426
13427          ;
13428          ;ILLEGAL INSTRUCTION TRAP SERVICE ROUTINE
13429          ;
13430 056566 005037 177572          ILLSER: CLR @#MMRO ;TURN OFF MEM MGMT - OVERWRITTEN WITH NOPS

```

```

13431                                     ; IF NO MEM MGMT ON SYSTEM (REF SIZEPT ROUTINE)
13432 056572 005737 002154             TST PT34                       ;11/34 TYPE PROCESSOR?
13433 056576 001403                     BEQ 2$                          ;BRANCH IF NO
13434                                     ;NOTE:ON 11/34 HALT IN 'USER MODE TRAPS TO 10 (NOT 4)
13435 056600 021627 047504             CMP (SP),#TINRET+2             ;WAS HALT AT CIS INST RETURN LOC?
13436 056604 001452                     BEQ 3$                          ;BRANCH IF YES
13437 056606 022737 000001 001420 2$:  CMP #1,TOTTC                    ;WAS ILLEGAL INST TRAP DURING TEST #1 ?
13438 056614 001011                     BNE 1$                          ;BRANCH IF NO
13439 056616 021627 047466             CMP (SP),#TINST+2             ;WAS ILLEGAL INST THE CIS INST?
13440 056622 001006                     BNE 1$                          ;BRANCH IF NO
13441 056624                             PRINTB #CISQ                    ;PRINT CISP PRESENT? WARNING
    (6) 056624 012746 012003             MOV #CISQ,-(SP)
    (3) 056630 010600                     MOV SP,R0
    (4) 056632 004737 065304             JSR PC,FPRINT
13442 056636 000000                     HALT
13443 056640 011637 002166             MOV (SP),TRPLOC                ;GET LOCATION THAT CAUSED TRAP
13444 056644 162737 000002 002166     SUB #2,TRPLOC
13445 056652                             PRINTB #TRAP10                  ;PRINT TRAP TO 10
    (6) 056652 012746 012144             MOV #TRAP10,-(SP)
    (3) 056656 010600                     MOV SP,R0
    (4) 056660 004737 065304             JSR PC,FPRINT
13446 056664                             PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTC ;PRINT ADDITIONAL TRAP I
    (12) 056664 013746 001420            MOV TOTTC,-(SP)
    (11) 056670 013746 001416            MOV TOTTC,-(SP)
    (10) 056674 013746 047464            MOV TINST,-(SP)
    (9) 056700 013746 002162            MOV DEN,-(SP)
    (8) 056704 013746 002160            MOV MODE,-(SP)
    (7) 056710 013746 002166            MOV TRPLOC,-(SP)
    (6) 056714 012746 012300            MOV #TRPINF,-(SP)
    (3) 056720 010600                     MOV SP,R0
    (4) 056722 004737 065304             JSR PC,FPRINT
13447 056726 000000                     HALT
13448 056730 000002                     RTI
13449 056732 016637 000002 003662 3$:  MOV 2(SP),TCCR                  ;SAVE CIS INST RETURN CONDITON CODES
13450 056740 005726                     TST (SP)+
13451 056742 005726                     TST (SP)+
13452 056744 000137 047526             JMP SUHRET
13453
13454
13455                                     ;
13456                                     ;MEMORY MANAGEMENT VIOLATION TRAP SERVICE ROUTINE
13457                                     ;
13458 056750 005037 177572             MMVIOL: CLR @MMR0                ;TURN OFF MEM MGMT
13459 056754 011637 002166             MOV (SP),TRPLOC                ;GET LOCATION WHICH CAUSED TRAP
13460 056760 162737 000002 002166     SUB #2,TRPLOC
13461 056766                             PRINTB #MMVMSG                  ;PRINT MEMORY MANAGEMENT VIOL
    (6) 056766 012746 012106             MOV #MMVMSG,-(SP)
    (3) 056772 010600                     MOV SP,R0
    (4) 056774 004737 065304             JSR PC,FPRINT
13462 057000                             PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTC
    (12) 057000 013746 001420            MOV TOTTC,-(SP)
    (11) 057004 013746 001416            MOV TOTTC,-(SP)
    (10) 057010 013746 047464            MOV TINST,-(SP)
    (9) 057014 013746 002162            MOV DEN,-(SP)

```


| | | | | | | | |
|-------|--------|--------|--------|--------|------|------------------|-------------------------------------|
| (8) | 057020 | 013746 | 002160 | | MOV | MODE,-(SP) | |
| (7) | 057024 | 013746 | 002166 | | MOV | TRPLOC,-(SP) | |
| (6) | 057030 | 012746 | 012300 | | MOV | #TRPINF,-(SP) | |
| (3) | 057034 | 010600 | | | MOV | SP,R0 | |
| (4) | 057036 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 13463 | 057042 | 000000 | | | HALT | | |
| 13464 | 057044 | 000002 | | | RTI | | |
| 13465 | | | | | | | |
| 13466 | | | | | | | |
| 13467 | | | | | | | |
| 13468 | | | | | | | |
| 13469 | 057046 | | | | | | |
| 13470 | 057046 | 105777 | 122616 | | 1\$: | TSTB @TKS | :WAIT FOR A CHARACTER |
| 13471 | 057052 | 100375 | | | | BPL 1\$ | |
| 13472 | 057054 | 117737 | 122632 | 064672 | | MOVB @TKB,RCHAR | :READ & SAVE CHAR |
| 13473 | 057062 | 042737 | 177600 | 064672 | | BIC #^C177,RCHAR | :GET RID OF JUNK IF ANY |
| 13474 | 057070 | 004737 | 064654 | | | JSR PC,ECHAR | :ECHO CHARACTER |
| 13475 | 057074 | 123727 | 064672 | 000113 | | CMPB RCHAR,#113 | :IS CHAR A 'K' |
| 13476 | 057102 | 001411 | | | | BEQ 6\$ | :BRANCH IF YES |
| 13477 | 057104 | 123727 | 064672 | 000123 | | CMPB RCHAR,#123 | :IS CHAR AN 'S' |
| 13478 | 057112 | 001410 | | | | BEQ 3\$ | :BRANCH IF YES |
| 13479 | 057114 | 123727 | 064672 | 000125 | | CMPB RCHAR,#125 | :IS CHAR A 'U' ? |
| 13480 | 057122 | 001410 | | | | BEQ 4\$ | :BRANCH IF YES |
| 13481 | 057124 | 000414 | | | | BR 5\$ | :CHAR IS ILLEGAL :RETURN TO CALL +2 |
| 13482 | 057126 | 005037 | 002160 | | 6\$: | CLR MODE | :SET MODE TO KERNEL (0) |
| 13483 | 057132 | 000407 | | | | BR 2\$ | |
| 13484 | 057134 | 012737 | 000001 | 002160 | 3\$: | MOV #1,MODE | :SET MODE WORD TO SUPERVISOR (1) |
| 13485 | 057142 | 000403 | | | | BR 2\$ | |
| 13486 | 057144 | 012737 | 000003 | 002160 | 4\$: | MOV #3,MODE | :SET MODE WORD TO USER (3) |
| 13487 | 057152 | 062716 | 000002 | | 2\$: | ADD #2,(SP) | |
| 13488 | 057156 | 000207 | | | 5\$: | RTS PC | |
| 13489 | | | | | | | |
| 13490 | | | | | | | |

```

13492          .SBTTL          PROGRAMMABLE CLOCK SERVICE ROUTINE
13494          ;NOTE: LOCATIONS 57670-60070 ARE RESERVED FOR STACK USAGE DURING
13495          ;          EXECUTION OF CIS STACK PROBEAHEAD MEMORY MGMT ABORT TESTS.
13496
13497          060100          . =60100
13499
13500          ; KW11-P (PROGRAMMABLE CLOCK) INTERRUPT SERVICE ROUTINES (ALWAYS ENTERED IN KERNEL MODE)
13501          ;
13502 060100      PCIS3:          ;P CLK SERVICE ROUTINE USED FOR INTR DURING
13503          ;          EXECUTION OF 'STATE DISTURBING' DIVPI
13504          ;          INSTRUCTION BELOW.
13505 060100 042777 000001 122406      BIC #1,@PC1CSR          ;TURN OFF PCLK1
13506 060106 000002          RTI
13507
13508 060110 005037 177572      PCIS2: CLR @#MMR0          ;P CLK INTR SERVICE ROUTINE USED WHEN
13509          ;          TESTING LATENCY
13510          ;THIS CLR INSTRUCTION TURNS OFF MEMORY MANAGEMENT
13511          ; THE CLR IS OVERWRITTEN WITH NOPS IF NOT 11/44
13512          ; (SEE SIZEPT ROUTINE).
13513 060114 017737 122410 002536      MOV @PC2CTR,LATCT          ;SAVE P-CLK2 COUNTER FOR LATENCY CALCULATION
13514 060122 042777 000001 122374      BIC #1,@PC2CSR          ;TURN OFF PCLK2
13515 060130 005077 122372          CLR @PC2CSB          ;CLEAR PCLK2 COUNTER
13516
13517 060134 005037 177572      PCIS1: CLR @#MMR0          ;NORMAL P CLK INTERRUPT SERVICE ROUTINE
13518          ;THIS CLR TURNS OFF MEM MGMT.
13519          ;THIS CLR GETS OVERWRITTEN WITH NOPS IF NOT 11/44
13520 060140 042777 000001 122346      BIC #1,@PC1CSR          ;TURN OFF P-CLK1
13521 060146 004737 061206      JSR PC,SGPRO6          ;SAVE GENERAL PURPOSE REGS 0-6
13522 060152 021627 047464          CMP (SP),#TINST          ;INTERRUPTED THE TEST INST???
13523 060156 001154          BNE 1$          ;BRANCH IF NO
13524 060160 032766 000400 000002      BIT #400.2(SP)          ;IS PSW BIT 8 SET?
13525 060166 001407          BEQ 4$          ;BRANCH IF NO
13526 060170 005237 002542          INC INTCT          ;UPDATE INTERRUPT COUNT
13527 060174 052737 040000 002132      BIS #40000,FATAL          ;SET INTERRUPT INDICATOR IN FATAL ERROR WORD
13528 060202 004737 061040          JSR PC,RECLAT          ;RECORD LATENCY
13529 060206 004737 061434          JSR PC,STATCG          ;HAS THE 'STATE' OF CIS INST CHANGED?
13530 060212 000504          BR 5$          ;NO RETURN
13531 060214 005237 002534          INC PROGCT          ;YES RETURN - UPDATE PROGRESS COUNT
13532 060220 032766 000400 000002      BIT #400.2(SP)          ;IS PSW BIT 8 SET?
13533 060226 001025          BNE 6$          ;BRANCH IF YES
13534 060230          PRINTB #HLTMSG
13534 (6) 060230 012746 012163          MOV #HLTMSG,-(SP)
13534 (3) 060234 010600          MOV SP,R0
13534 (4) 060236 004737 065304          JSR PC,FPRINT
13535 060242 012737 060300 002170      MOV #100$,HLTLOC
13536 060250 004737 054766          JSR PC,IDINFO          ;IDENTIFY FAILING INST
13537 060254          PRINTB #FORM42          ;MSG: CIS INST WAS SUSPENDED TO SERVICE INTR
13537 (6) 060254 012746 014671          MOV #FORM42,-(SP)
13537 (3) 060260 010600          MOV SP,R0
13537 (4) 060262 004737 065304          JSR PC,FPRINT
13538 060266          PRINTB #FORM43          ;MSG: PSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT
13538 (6) 060266 012746 014760          MOV #FORM43,-(SP)
13538 (3) 060272 010600          MOV SP,R0
13538 (4) 060274 004737 065304          JSR PC,FPRINT

```

```
13539 060300 000000          100$: HALT          ;BIT 8 OF PSW SHOULD HAVE BEEN SET!!!
13540                                     ;PRESS CONTINUE TO PROCEED WITH TESTING
13541
13542 060302          6$:          ;INST IS IN A NEW PART OF ITS OPERATION
13543 060302 012737 000001 002552 MOV #1,INTRVL ;SET P-CLK INTERVAL TO MINIMUM = 1
13544 060310 004737 061316 JSR PC,SAVST  ;SAVE CIS INST STATE
13545 060314 004737 061152          11$: JSR PC,DIC    ;ALLOW INTERRUPT DURING THIS DIVPI IF REQUESTED
13546 060320 076175 DIVPI        ;DISTURB INTERNAL CISP STATE BY
13547 060322 003146 DIVDS          ; EXECUTING A DIVP IN-LINE INST.
13548 060324 003146 DIVDS
13549 060326 003152 DIVDD
13550 060330 042777 000001 122156 BIC #1,@PC1CSR ;TURN OFF PCLK1
13551 060336 013777 002036 122144 MOV SPCV,@PCLK1V ;RESTORE PCLK VECTOR
13552 060344 013737 002040 120400 MOV TPRECS,PRECSK ;RESTORE 65TH STACK WORD
13553 060352 004737 061264 JSR PC,RGPRO6 ;RESTORE GENERAL PURPOSE REGISTERS 0-6
13554 060356 013777 002552 122132 MOV INTRVL,@PC1CSB ;SET INTERVAL
13555 060364 005737 002540 TST LATEN    ;LATENCY TESTING?
13556 060370 001403 BEQ 7$       ;BRANCH IF NO
13557 060372 052777 000001 122124 BIS #1,@PC2CSR ;TURN ON P-CLK2
13558 060400 005737 002156          7$: TST MMFLG   ;TESTING WITH MEM MGMT ON?
13559 060404 001403 BEQ 8$       ;BRANCH IF NO
13560 060406 052737 000001 177572 BIS #1,@MMRO  ;TURN ON MEM MGMT
13561 060414 052777 000001 122072 8$: BIS #1,@PC1CSR ;TURN ON P-CLK1
13562 060422 000002          14$: RTI          ;RETURN FROM SERVICE
13563
13564 060424 005237 002552          5$: INC INTRVL  ;INCREASE P-CLK1 INTERVAL
13565 060430 023737 002552 002556 CMP INTRVL,MAXIVL ;IS INTERVAL GREATER THAN SOME PRESET
13566                                     ; MAXIMUM ALLOWED?
13567 060436 103420 BLO 12$     ;BRANCH IF NO
13568 060440 PRINTB #HLTMSG
(6) 060440 012746 012163 MOV #HLTMSG,-(SP)
(3) 060444 010600 MOV SP,R0
(4) 060446 004737 065304 JSR PC,FPRINT
13569 060452 012737 060476 002170 MOV #101$,HLTLOC
13570 060460 004737 054766 JSR PC,IDINFO ;IDENTIFY FAILING INST
13571 060464 PRINTB #NOPROG
(6) 060464 012746 012202 MOV #NOPROG,-(SP)
(3) 060470 010600 MOV SP,R0
(4) 060472 004737 065304 JSR PC,FPRINT
13572 060476 000000          101$: HALT      ;CIS INST DID NOT MAKE PROGRESS -
13573                                     ; INTERVAL TILL INTERRUPT EXCEEDS USER
13574                                     ; DEFINED MAXIMUM ALLOWED.
13575 060500 005737 002534          12$: TST PROGCT ;HAS PROGRESS BEEN MADE ON THIS INST PREVIOUSLY?
13576 060504 001445 BEQ 13$     ;BRANCH IF NO
13577 060506 000702 BR 11$      ;DISTURB INTERNAL STATE; THEN RETURN
13578                                     ; FOR MORE PROGRESS
13579 060510 101054          1$: BHI 15$    ;BRANCH IF INST ALREADY HAS COMPLETED (PC>TINST)
13580 060512 032766 000400 000002 BIT #400,2(SP) ;DID NOT REACH INST YET.
13581                                     ;VERIFY THAT BIT 8 OF PSW IS NOT SET.
13582 060520 001435 BEQ 17$    ;BRANCH IF BIT 8 = 0.
13583 060522 PRINTB #HLTMSG
(6) 060522 012746 012163 MOV #HLTMSG,-(SP)
(3) 060526 010600 MOV SP,R0
(4) 060530 004737 065304 JSR PC,FPRINT
```

| | | | | | | | |
|-------|--------|--------|--------|--------|--------|--------------------|---|
| 13584 | 060534 | 012737 | 000102 | 002170 | | MOV #102,HLTLOC | |
| 13585 | 060542 | 004737 | 054766 | | | JSR PC,IDINFO | ;IDENTIFY FAILING INST |
| 13586 | 060546 | | | | | PRINTB #FORM44 | ;MSG: BIT 8 OF PSW SET WITH PC < CIS INST PC |
| (6) | 060546 | 012746 | 015041 | | | MOV #FORM44,-(SP) | |
| (3) | 060552 | 010600 | | | | MOV SP,R0 | |
| (4) | 060554 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13587 | 060560 | | | | | PRINTB #FORM45 | ;MSG: SUSPECT THAT CIS INST BACKED UP TOO FAR |
| (6) | 060560 | 012746 | 015114 | | | MOV #FORM45,-(SP) | |
| (3) | 060564 | 010600 | | | | MOV SP,R0 | |
| (4) | 060566 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13588 | 060572 | | | | | PRINTB #FORM46 | ;MSG: WHEN SERVICING INTERRUPT |
| (6) | 060572 | 012746 | 015173 | | | MOV #FORM46,-(SP) | |
| (3) | 060576 | 010600 | | | | MOV SP,R0 | |
| (4) | 060600 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13589 | 060604 | 000000 | | | 102\$: | HALT | ;BIT 8 OF PSW SET WITH PC < CIS INST PC. |
| 13590 | 060606 | 042766 | 000400 | 000002 | | BIC #400,2(SP) | ;SUSPECT THAT CIS INST BACKED UP PC TOO FAR |
| 13591 | | | | | | | ; WHEN SERVICING INTERRUPT. |
| 13592 | | | | | | | ;PRESS CONTINUE TO PROCEED WITH TESTING |
| 13593 | 060614 | 005237 | 002552 | | 17\$: | INC INTRVL | |
| 13594 | 060620 | 013777 | 002552 | 121670 | 13\$: | MOV INTRVL,@PC1CSB | ;INCREASE INTERVAL |
| 13595 | 060626 | 004737 | 061264 | | | JSR PC,RGPR06 | ;RESTORE REGISTERS |
| 13596 | 060632 | 005726 | | | | TST (SP)+ | ;FIX UP STACK POINTER |
| 13597 | 060634 | 005726 | | | | TST (SP)+ | |
| 13598 | 060636 | 000137 | 047424 | | | JMP TOPC2 | ;RETURN TO TURN ON PCLK-1 POINT |
| 13599 | | | | | | | |
| 13600 | 060642 | 004737 | 061040 | | 15\$: | JSR PC,RECLAT | ;RECORD LATENCY |
| 13601 | 060646 | 032737 | 000100 | 047464 | | BIT #100,TINST | ;IS INST UNDER TEST AN IN-LINE INST |
| 13602 | 060654 | 001430 | | | | BEQ 20\$ | ;BRANCH IF NO |
| 13603 | 060656 | 021637 | 002214 | | | CMP (SP),ICOMPC | ;VERIFY THAT PC HAS BEEN ADJUSTED TO POINT |
| 13604 | | | | | | | ; TO NEXT INST |
| 13605 | 060662 | 103025 | | | | BHIS 20\$ | ;BRANCH IF PC IS OK |
| 13606 | 060664 | | | | | PRINTB #HLTMSG | |
| (6) | 060664 | 012746 | 012163 | | | MOV #HLTMSG,-(SP) | |
| (3) | 060670 | 010600 | | | | MOV SP,R0 | |
| (4) | 060672 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13607 | 060676 | 012737 | 060734 | 002170 | | MOV #103\$,HLTLOC | |
| 13608 | 060704 | 004737 | 054766 | | | JSR PC,IDINFO | ;IDENTIFY FAILING INST |
| 13609 | 060710 | | | | | PRINTB #FORM48 | ;MSG: IN-LINE CIS INST COMPLETED WITH PC |
| (6) | 060710 | 012746 | 015325 | | | MOV #FORM48,-(SP) | |
| (3) | 060714 | 010600 | | | | MOV SP,R0 | |
| (4) | 060716 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13610 | 060722 | | | | | PRINTB #FORM49 | ;MSG: POINTING AT IN-LINE OPERANDS RATHER |
| (6) | 060722 | 012746 | 015403 | | | MOV #FORM49,-(SP) | |
| (3) | 060726 | 010600 | | | | MOV SP,R0 | |
| (4) | 060730 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 13611 | | | | | | | ; THAN NEXT INST. |
| 13612 | 060734 | 000000 | | | 103\$: | HALT | ;PRESS CONTINUE TO PROCEED WITH TESTING |
| 13613 | 060736 | 004737 | 061264 | | 20\$: | JSR PC,RGPR06 | ;RESTORE REGISTERS |
| 13614 | 060742 | 032766 | 000400 | 000002 | | BIT #400,2(SP) | ;CIS INST COMPLETE - VERIFY THAT PSW BIT 8 |
| 13615 | | | | | | | ; IS NOT SET |
| 13616 | 060750 | 001420 | | | | BEQ 16\$ | |
| 13617 | 060752 | | | | | PRINTB #HLTMSG | |
| (6) | 060752 | 012746 | 012163 | | | MOV #HLTMSG,-(SP) | |
| (3) | 060756 | 010600 | | | | MOV SP,R0 | |

```

(4) 060760 004737 065304      JSR PC,FPRINT
13618 060764 012737 061010 002170  MOV #104$,HLTLOC
13619 060772 004737 054766      JSR PC,IDINFO      ;IDENTIFY FAILING INST
13620 060776      PRINTB #FORM47    ;MSG: CIS INST COMPLETED BUT PSW BIT 8 STILL SET
(6) 060776 012746 015244      MOV #FORM47,-(SP)
(3) 061002 010600      MOV SP,R0
(4) 061004 004737 065304      JSR PC,FPRINT
13621 061010 000000      104$: HALT      ;CIS INST COMPLETED BUT PSW BIT 8 STILL SET
13622      ;PRESS CONTINUE TO PROCEED WITH TESTING
13623 061012 042766 000400 000002 16$: BIC #400,2(SP)
13624 061020 012777 000001 121470  MOV #1,@PC1CSB    ;SET UP FOR NEXT PASS
13625 061026 012737 000001 002552  MOV #1,INTRVL
13626 061034 000137 060422      JMP 14$
13627
13628      ;KW11-P INTERRUPT SERVICE ROUTINE SUBROUTINES
13629      ;
13630      ;ROUTINE TO RECORD INTERRUPT LATENCY
13631      ;
13632 061040 163737 002552 002536 RECLAT: SUB INTRVL,LATCT    ;CALCULATE LATENCY
13633 061046 005737 002554      TST STOPLA        ;STOP ON EXCESSIVE LATENCY?
13634 061052 001424      BEQ 2$           ;BRANCH IF NO
13635 061054 023737 002554 002536  CMP STOPLA,LATCT  ;IS LATENCY EXCESSIVE?
13636 061062 101020      BHI 2$          ;BRANCH IF NO
13637 061064      PRINTB #HLTMSG
(6) 061064 012746 012163      MOV #HLTMSG,-(SP)
(3) 061070 010600      MOV SP,R0
(4) 061072 004737 065304      JSR PC,FPRINT
13638 061076 012737 000100 002170  MOV #100,HLTLOC
13639 061104 004737 054766      JSR PC,IDINFO    ;IDENTIFY FAILING INST
13640 061110      PRINTB #LATEXC
(6) 061110 012746 012250      MOV #LATEXC,-(SP)
(3) 061114 010600      MOV SP,R0
(4) 061116 004737 065304      JSR PC,FPRINT
13641 061122 000000      100$: HALT      ;LATENCY EXCEEDED USER DEFINED
13642      ; 'MAXIMUM ALLOWABLE'
13643
13644 061124 013701 002272      2$: MOV OCTIC,R1    ;RECORD LATENCY
13645 061130 006301      ASL R1
13646 061132 062701 004102      ADD #ILATEN,R1   ;FORM POINTER INTO INST LATENCY TABLE
13647 061136 021137 002536      CMP (R1),LATCT  ;IS LATENCY BIGGER THAN THAT ALREADY
13648      ; RECORDED FOR INST?
13649 061142 101002      BHI 1$          ;BRANCH IF NO
13650 061144 013711 002536      MOV LATCT,(R1)  ;SAVE NEW LATENCY VALUE
13651 061150 000207      1$: RTS PC
13652
13653      ;ROUTINE TO TURN ON P-CLK DURING DIVPI 'STATE DISTURBING' INST
13654      ;
13655 061152 017737 121332 002036 DIC: MOV @PCLK1V,SPCV    ;SAVE CONTENTS OF P CLK INTERR VECTOR
13656 061160 013737 120400 002040  MOV PRECSK,TPRECS ;SAVE CONTENTS OF 65TH STACK WORD
13657 061166 012777 060100 121314  MOV #PCIS3,@PCLK1V ;SETUP INTR VECTOR
13658 061174 000207      DI: RTS PC      ;OVERWRITTEN WITH A NOP IF USER REQUESTS
13659 061176 052777 000001 121310  BIS #1,@PC1CSR  ;TURN ON P-CLK1 - ENABLE INTR DURING
13660      ; SUBSEQUENT (STATE DISTURBING) CIS INST
13661 061204 000207      RTS PC

```

```

13662
13663 ;ROUTINE TO SAVE GENERAL PURPOSE REGISTERS 0 THROUGH 6.
13664 ;
13665 061206 SGPR06:
13666 061206 010037 003000 MOV R0,SGPRO
13667 061212 010137 003002 MOV R1,SGPR1
13668 061216 010237 003004 MOV R2,SGPR2
13669 061222 010337 003006 MOV R3,SGPR3
13670 061226 010437 003010 MOV R4,SGPR4
13671 061232 010537 003012 MOV R5,SGPR5
13672 061236 032737 030000 177776 BIT #30000,PSW ;WAS PREVIOUS MODE USER OR SUPV?
13673 061244 001404 BEQ 1$ ;BRANCH IF NO
13674 061246 006506 MFPI SP ;GET PREVIOUS MODE SP
13675 061250 012637 003014 MOV (SP)+,SGPR6 ;STORE PREVIOUS MODE SP IN SGPR6
13676 061254 000402 BR 2$
13677 061256 010637 003014 1$: MOV SP,SGPR6
13678 061262 000207 2$: RTS PC
13679
13680 ;
13681 ;ROUTINE TO RESTORE GENERAL PURPOSE REGISTERS 0 THROUGH 5
13682 ;
13683 061264 RGPR06:
13684 061264 013700 003000 MOV SGPRO,R0
13685 061270 013701 003002 MOV SGPR1,R1
13686 061274 013702 003004 MOV SGPR2,R2
13687 061300 013703 003006 MOV SGPR3,R3
13688 061304 013704 003010 MOV SGPR4,R4
13689 061310 013705 003012 MOV SGPR5,R5
13690 ;NOTE NO NEED TO RESTORE R6 BECAUSE IT HAS NOT CHANGED.
13691 061314 000207 RTS PC
13692
13693 ;
13694 ;ROUTINE TO SAVE STATE OF CISP - STATE=STACK POINTER,GENERAL PURPOSE REGISTER
13695 ; CONTENTS & STACK CONTENTS.
13696 ;
13697 061316 SAVST:
13698 061316 016637 000004 002560 MOV 4(SP),STATPS ;SAVE PSW STATE
13699 061324 013737 003000 002562 MOV SGPRO,STATR0 ;SAVE STATE OF GENERAL PURPOSE REGS 0-6
13700 061332 013737 003002 002564 MOV SGPR1,STATR1
13701 061340 013737 003004 002566 MOV SGPR2,STATR2
13702 061346 013737 003006 002570 MOV SGPR3,STATR3
13703 061354 013737 003010 002572 MOV SGPR4,STATR4
13704 061362 013737 003012 002574 MOV SGPR5,STATR5
13705 061370 013737 003014 002576 MOV SGPR6,STATR6
13706 061376 013700 003014 MOV SGPR6,R0 ;SAVE STACK CONTENTS
13707 061402 062700 000006 ADD #6,R0

```

```
13709 061406 020027 120602          CMP R0,#CSTACK          ;DID CIS INST PUSH ANYTHING ONTO STACK?
13710 061412 103007                   BHS 1$                  ;BRANCH IF NO
13711 061414 012702 003000          MOV #SCSTK,R2          ;COPY USED PORTION OF STACK INTO A
13712 061420 012701 120602          MOV #CSTACK,R1        ; SAVE AREA.
13713 061424 014142                   2$: MOV -(R1),-(R2)
13714 061426 020100                   CMP R1,R0              ;ALL OF USED PORTION OF STACK COPIED?
13715 061430 103375                   BHS 2$                  ;BRANCH IF NO
13716 061432 000207                   1$: RTS PC
13717
13718
13719
13720
; ROUTINE TO CHECK FOR A CISP STATE CHANGE
; RETURNS TO CALL ON NO CHANGE; CALL+2 ON CHANGE
```

13722
13723 061434
13724 061434 026637 000004 002560 :STATCG: CMP 4(SP),STATPS ;DID PSW CHANGE?

| | | | | | | |
|-------|--------|--------|--------|--------|------------------|--|
| 13726 | 061442 | 001054 | | | BNE STHASC | :BRANCH IF YES |
| 13727 | 061444 | 023737 | 003014 | 002576 | CMP SGPR6,STATR6 | :DID STACK POINTER CHANGE? |
| 13728 | 061452 | 001050 | | | BNE STHASC | :BRANCH IF YES |
| 13729 | 061454 | 023737 | 003000 | 002562 | CMP SGPRO,STATR0 | :DID ANY OF THE GENERAL PURPOSE REGISTER |
| 13730 | | | | | | :CONTENTS CHANGE? |
| 13731 | 061462 | 001044 | | | BNE STHASC | :BRANCH IF R0 CHANGED |

```
13733 061464 023737 003002 002564      CMP SGPR1,STATR1
13734 061472 001040                      BNE STHASC      ;BRANCH IF R1 HAS CHANGED
13735 061474 023737 003004 002566      CMP SGPR2,STATR2
13736 061502 001034                      BNE STHASC      ;BRANCH IF R2 HAS CHANGED
13737 061504 023737 003006 002570      CMP SGPR3,STATR3
13738 061512 001030                      BNE STHASC      ;BRANCH IF R3 HAS CHANGED
13739 061514 023737 003010 002572      CMP SGPR4,STATR4
13740 061522 001024                      BNE STHASC      ;BRANCH IF R4 HAS CHANGED
13741 061524 023737 003012 002574      CMP SGPR5,STATR5
13742 061532 001020                      BNE STHASC      ;BRANCH IF R5 HAS CHANGED
13743 061534 013700 003014              MOV SGPR6,R0    ;DID THE STACK CONTENTS CHANGE
13744 061540 062700 000006              ADD #6,R0
13745 061544 020027 120602              CMP R0,#CSTACK ;DID ANYTHING GET PUSHED ONTO THE STACK?
13746 061550 103010                      BHS 1$          ;BRANCH IF NO
13747 061552 012702 003000              MOV #SCSTK,R2
13748 061556 012701 120602              MOV #CSTACK,R1
13749 061562 024142                      2$: CMP -(R1),-(R2) ;DID ANY OF THE INFORMATION ON THE STACK GET CHANGED?
13750 061564 001003                      BNE STHASC      ;BRANCH IF YES
13751 061566 020100                      CMP R1,R0       ;ALL OF STACK CHECKED?
13752 061570 103374                      BHS 2$          ;BRANCH IF NO
13753 061572 000402                      1$: BR NOSCHG
13754 061574 062716 000002              STHASC: ADD #2,(SP)
13755 061600 000207                      NOSCHG: RTS PC
13756
13757
13758 061602      ;ROUTINE TO CHECK FOR AND SETUP P-CLK 1
PC1CK:      ;TEST FOR P-CLKS PRESENT
```

```
13760 061602 013701 002532          MOV TIMEOUT,R1          ;SAVE TIME OUT VECTOR CONTENTS
13761 061606 011146          MOV (R1),-(SP)
13762 061610 012721 061676          MOV #1$(R1)+
13763 061614 011146          MOV (R1),-(SP)
13764 061616 005011          CLR (R1)
13765
13766 061620 005777 120670          TST @PC1CSR            ;ATTEMPT ACCESS P-CLK1 CSR
13767                                     ;PCLK1 IS RESPONDING
13768 061624 012777 060134 120656          MOV #PCIS1,@PCLK1V    ;SET UP P-CLK1 INTERRUPT SERVICE VECTOR
13769 061632 005077 120656          CLR @PC1CSR           ;CLEAR P-CLK1 CSR
13770 061636 012777 000000 120650          MOV #000,@PC1CSR     ;SET P-CLK1 FOR SINGLE INT,COUNT DOWN,100K HZ
13771                                     ;CLOCK, INT ENABLE
13772 061644 012777 000000 120644          MOV #1,@PC1CSB       ;SET COUNTER TO 1 (1 MICRO SEC INTERVAL
13773                                     ;WITH 1 MHZ EXTERNAL CLOCK.
13774 061652 012737 000001 002552          MOV #1,INTRVL        ;SAVE INTERVAL SETTING
13775 061660 013737 002544 047434          MOV KNOPI,TOPCI      ;OVERWRITE BRANCH TO ALLOW TURNING ON OF
13776                                     ;PCLK1 PRIOR TO CIS INST EXECUTION
13777 061666 062766 000004 000004          ADD #4,4(SP)         ;ADJUST STACK FOR 'EXISTS RETURN'
13778 061674 000402          BR 2$
13779 061676 005726          1$: TST (SP)+           ;FIX UP STACK POINTER
13780 061700 005726          TST (SP)+
13781 061702 013701 002532          2$: MOV TIMEOUT,R1    ;RESTORE TIME OUT INTR VECTOR TO ORIGINAL STATE
13782 061706 012661 000002          MOV (SP)+,2(R1)
13783 061712 012611          MOV (SP)+,(R1)
13784 061714 000207          RTS PC
13785
13786                                     ;ROUTINE TO CHECK FOR AND SETUP P-CLK2
13787 061716 013701 002532          PC2CK: MOV TIMEOUT,R1 ;SAVE TIME OUT VECTOR CONTENTS
13788 061722 011146          MOV (R1),-(SP)
13789 061724 012721 062024          MOV #1$(R1)+
13790 061730 011146          MOV (R1),-(SP)
13791 061732 005011          CLR (R1)
13792 061734 005777 120564          TST @PC2CSR            ;ATTEMPT ACCESS OF PCLK 2 CSR
13793 061740 012737 177777 002540          MOV #177777,LATEN    ;SET LATENCY TESTING FLAG
13794 061746 012700 004102          MOV #ILATEN,R0       ;CLEAR INTERRUPT LATENCY TABLE
13795 061752 005020          12$: CLR (R0)+
13796 061754 020027 004172          CMP R0,#LATEND+2
13797 061760 001374          BNE 12$
13798 061762 012777 060110 120520          MOV #PCIS2,@PCLK1V
13799 061770 005077 120530          CLR @PC2CSR          ;CLEAR P-CLK2 CSR
```

| | | | | | | |
|-------|--------|--------|--------|--------|---------------------|--|
| 13801 | 061774 | 012777 | 000020 | 120522 | MOV #20,@PC2CSR | ;SET P-CLK2 FOR INT. DISABLE, COUNT UP |
| 13802 | | | | | | ; 100KHZ CLOCK |
| 13803 | 062002 | 005077 | 120520 | | CLR @PC2CSB | ;SET COUNTER TO 0 |
| 13804 | 062006 | 013737 | 002546 | 047424 | MOV KNOP2, TOPC2 | ;OVERWRITE BRANCH TO ALLOW TURNING ON OF |
| 13805 | | | | | | ; PCLK2 PRIOR TO CIS INST EXECUTION |
| 13806 | 062014 | 062766 | 000004 | 000004 | ADD #4,4(SP) | ;ADJUST STACK FOR 'EXISTS RETURN' |
| 13807 | 062022 | 000402 | | | BP 2\$ | |
| 13808 | 062024 | 005726 | | | 1\$: TST (SP)+ | ;FIX UP STACK POINTER |
| 13809 | 062026 | 005726 | | | TST (SP)+ | |
| 13810 | 062030 | 013701 | 002532 | | 2\$: MOV TIMEOUT,R1 | ;RESTORE TIME OUT INTERRUPT VECTOR TO |
| 13811 | 062034 | 012661 | 000002 | | MOV (SP)+,2(R1) | ; ORIGINAL STATE |
| 13812 | 062040 | 012611 | | | MOV (SP)+,(R1) | |
| 13813 | 062042 | 000207 | | | RTS PC | |

| | | | | | | |
|-------|--------|--------|--------|--------|-----------------------|---|
| 13831 | 062072 | 021627 | 047464 | | CMP (SP),#TINST | : INTERRUPTED THE CIS INST UNDER TEST? |
| 13832 | 062076 | 001045 | | | BNE EXLTCS | : NO - EXIT LTC SERVICE |
| 13833 | 062100 | 004737 | 061434 | | JSR PC,STATCG | : HAS THE STATE OF CIS INST CHANGED? |
| 13834 | 062104 | 000442 | | | BR EXLTCS | : NO RETURN - EXIT LTC SERVICE |
| 13835 | 062106 | 032766 | 000400 | 000002 | BIT #400,2(SP) | : YES RETURN - IS PSW BIT 8 SET? |
| 13836 | 062114 | 001025 | | | BNE 1\$ | : BRANCH IF YES |
| 13837 | 062116 | | | | PRINTB #HLTMSG | |
| (6) | 062116 | 012746 | 012163 | | MOV #HLTMSG,-(SP) | |
| (3) | 062122 | 010600 | | | MOV SP,R0 | |
| (4) | 062124 | 004737 | 065304 | | JSR PC,FPRINT | |
| 13838 | 062130 | 012737 | 062166 | 002170 | MOV #100\$,HLTLOC | |
| 13839 | 062136 | 004737 | 054766 | | JSR PC,IDINFO | : IDENTIFY FAILING INST |
| 13840 | 062142 | | | | PRINTB #FORM42 | : MSG: CIS INST WAS SUSPENDED TO SERVICE INTR |
| (6) | 062142 | 012746 | 014671 | | MOV #FORM42,-(SP) | |
| (3) | 062146 | 010600 | | | MOV SP,R0 | |
| (4) | 062150 | 004737 | 065304 | | JSR PC,FPRINT | |
| 13841 | 062154 | | | | PRINTB #FORM43 | : MSG: PSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT |
| (6) | 062154 | 012746 | 014760 | | MOV #FORM43,-(SP) | |
| (3) | 062160 | 010600 | | | MOV SP,R0 | |
| (4) | 062162 | 004737 | 065304 | | JSR PC,FPRINT | |
| 13842 | 062166 | 000000 | | | 100\$: HALT | |
| 13843 | | | | | | |
| 13844 | 062170 | 005237 | 002542 | | 1\$: INC INTCT | : UPDATE INTERRUPT COUNT |
| 13845 | 062174 | 052737 | 040000 | 002132 | BIS #40000,FATAL | : SET INTR INDICATION IN FATAL ERROR WORD |
| 13846 | 062202 | 076175 | | | DIVPI | : DISTURB INTERNAL CISP STATE BY |
| 13847 | 062204 | 003146 | | | DIVDS | : EXECUTING A DIVP IN-LINE INST. |
| 13848 | 062206 | 003146 | | | DIVDS | |
| 13849 | 062210 | 003152 | | | DIVDD | |
| 13850 | 062212 | 004737 | 061264 | | EXLTCS: JSR PC,RGPR06 | : RESTORE GENERAL PURPOSE REGS 0-6 |
| 13851 | 062216 | 005737 | 002156 | | TST MMFLG | : TESTING WITH MEM MGMT ON ? |
| 13852 | 062222 | 001403 | | | BEQ 1\$ | : BRANCH IF NO |
| 13853 | 062224 | 013737 | 002210 | 177572 | MOV MMSTAT,@#MMR0 | : TURN ON MEM MGMT |
| 13854 | 062232 | 000002 | | | 1\$: RTI | : RETURN FROM SERVICE |

```
13856  
13857  
13858           ;LTC SYNC UP ROUTINE  
13859           ;  
13860           ;LTCSUP:  
13861 062234 005077 120562           CLR @LKS           ;DISABLE INTERRUPTS; CLEAR MONITOR  
13862  
13863 062240 032777 000200 120554 1$: BIT #200,@LKS       ;WAIT FOR CLOCK SIGNAL  
13864 062246 001774  
13865 062250 005077 120546           CLR @LKS  
13866 062254 000207           RTS PC  
13867
```

```
13869
13870      ;LTC - DETERMINE COUNT PER CLOCK TICK
13871      :
13872      :LTCINT:
13873      062256 052777 000100 120536      BIS #100,@LKS      ;ENABLE INTERRUPTS
13874      062264 005237 003024      1$: INC LCNT      ;COUNT TILL LTC INTERRUPTS
13875      062270 000775      BR 1$
13876      062272 005077 120524      LTCINT: CLR @LKS      ;DISABLE LTC INTERRUPTS
13877      062276 163737 003032 003024      SUB LTCDLY,LCNT    ;INTERRUPT RETURN
13878      062304 005726      TST (SP)+
13879      062306 005726      TST (SP)+      ;FIX UP STACK
13880      062310 000207      RTS PC
13881
13882
13883      ; LTC - ROUTINE TO CHECK FOR LINE TIME CLOCK ON SYSTEM
13884      :
13885      :LTC:
13886      062312 013701 002532      MOV TIMEOUT,R1      ;SAVE TIME OUT VECTOR
13887      062316 011146      MOV (R1),-(SP)      ;SETUP INTERRUPT VECTOR
13888      062320 012721 062364      MOV #1$, (R1)+
13889      062324 011146      MOV (R1),-(SP)
13890      062326 005011      CLR (R1)
13891      062330 005777 120466      TST @LKS      ;ATTEMPT ACCESS OF LTC
13892      062334 005077 120462      CLR @LKS      ;CLEAR LTC CSR
13893      062340 012777 062272 120450      MOV #LTCINT,@LTCIV ;LTC IS RESPONDING - SETUP LTC INTR VECTOR
```



```
13895 062346 013737 003030 047372      MOV KNOP4,TOLTC      ;OVERWRITE BRANCH TO ALLOW TURNING ON OF
13896                                     ; OF LTC PRIOR TO CIS INST EXECUTION.
13897 062354 062766 000004 000004      ADD #4,4(SP)        ;ADJUST RETURN TO CALL + 4
13898 062362 000402                                     BR 2$
13899 062364 005726      1$: TST (SP)+          ;FIX UP STACK POINTER
13900 062366 005726      TST (SP)+
13901 062370 013701 002532      2$: MOV TIMEOUT,R1   ;RESTORE TIME OUT INTR VECTOR TO
13902 062374 012661 000002      MOV (SP)+,2(R1)    ; ORIGINAL STATE
13903 062400 012611      MOV (SP)+,(R1)
13904 062402 000207      RTS PC
13905                                     ;
13906                                     ;RANDOM EXERCISE MODE SUBROUTINES
13907                                     ;
13908
13909 062404      SRNGST:           ;SUBROUTINE TO SAVE RANDOM # GEN STATE
13910                                     ;AT START OF EACH TEST
13911 062404 013737 063544 001776      MOV RNCON,STRNC
13912 062412 013737 063546 002000      MOV RP1,STRP1
13913 062420 013737 063550 002002      MOV RP2,STRP2
13914 062426 000207      RTS PC
13915
13916 062430      SRNGSX:           ;SUBROUTINE TO SAVE RANDOM # GEN. STATE X.
13917 062430 013737 063544 002004      MOV RNCON,SXRNC
13918 062436 013737 063546 002006      MOV RP1,SXRP1
13919 062444 013737 063550 002010      MOV RP2,SXRP2
13920 062452 000207      RTS PC
13921
13922 062454      RRNGSX:           ;SUBROUTINE TO RESTORE RANDOM # GEN STATE X.
13923 062454 013737 002004 063544      MOV SXRNC,RNCON
```

| | | | | | |
|-------|--------|--------|--------|--------|-------------------------|
| 13925 | 062462 | 013737 | 002006 | 063546 | MOV SGRP1,RP1 |
| 13926 | 062470 | 013737 | 002010 | 063550 | MOV SGRP2,RP2 |
| 13927 | 062476 | 000207 | | | RTS PC |
| 13928 | | | | | |
| 13929 | 062500 | | | | |
| 13930 | 062500 | 013737 | 063544 | 002012 | SRNGSY: MOV RNCON,SYRNC |

;SUBROUTINE TO SAVE RANDOM # GEN STATE Y.

| | | | | | | |
|-------|--------|--------|--------|--------|-----------------|--|
| 13932 | 062506 | 013737 | 063546 | 002014 | MOV RP1,SYRP1 | |
| 13933 | 062514 | 013737 | 063550 | 002016 | MOV RP2,SYRP2 | |
| 13934 | 062522 | 000207 | | | RTS PC | |
| 13935 | | | | | | |
| 13936 | 062524 | | | | RRNGSY: | ;SUBROUTINE TO RESTORE RANDOM # GEN STATE Y. |
| 13937 | 062524 | 013737 | 002012 | 063544 | MOV SYRNC,RNCON | |
| 13938 | 062532 | 013737 | 002014 | 063546 | MOV SYRP1,RP1 | |
| 13939 | 062540 | 013737 | 002016 | 063550 | MOV SYRP2,RP2 | |
| 13940 | 062546 | 000207 | | | RTS PC | |
| 13941 | | | | | | |
| 13942 | 062550 | | | | SRNGSW: | ;SUBROUTINE TO SAVE RANDOM # GEN STATE W. |
| 13943 | 062550 | 013737 | 063544 | 002020 | MOV RNCON,SWRNC | |
| 13944 | 062556 | 013737 | 063546 | 002022 | MOV RP1,SWRP1 | |
| 13945 | 062564 | 013737 | 063550 | 002024 | MOV RP2,SWRP2 | |
| 13946 | 062572 | 000207 | | | RTS PC | |
| 13947 | | | | | | |
| 13948 | 062574 | | | | RRNGSW: | ;SUBROUTINE TO RESTORE RANDOM # GEN STATE W. |
| 13949 | 062574 | 013737 | 002020 | 063544 | MOV SWRNC,RNCON | |
| 13950 | 062602 | 013737 | 002022 | 063546 | MOV SWRP1,RP1 | |
| 13951 | 062610 | 013737 | 002024 | 063550 | MOV SWRP2,RP2 | |
| 13952 | 062616 | 000207 | | | RTS PC | |
| 13953 | | | | | | |
| 13954 | 062620 | | | | SRNGSV: | ;SUBROUTINE TO SAVE RANDOM # GEN STATE V. |
| 13955 | 062620 | 013737 | 063544 | 002026 | MOV RNCON,SVRNC | |
| 13956 | 062626 | 013737 | 063546 | 002030 | MOV RP1,SVRP1 | |
| 13957 | 062634 | 013737 | 063550 | 002032 | MOV RP2,SVRP2 | |
| 13958 | 062642 | 000207 | | | RTS PC | |
| 13959 | | | | | | |
| 13960 | 062644 | | | | RRNGSV: | ;SUBROUTINE TO RESTORE RANDOM # GEN STATE V. |
| 13961 | 062644 | 013737 | 002026 | 063544 | MOV SVRNC,RNCON | |
| 13962 | 062652 | 013737 | 002030 | 063546 | MOV SVRP1,RP1 | |

13964 062660 013737 002032 063550 MOV SVRP2,RP2

13966 062666 000207
13967

RTS PC

```
13969      .SBTTL          RANDOM MODE SUBROUTINES
13970      :
13971      :SUBROUTINE TO GENERATE A RANDOM CIS INST AND LOAD ITS IDENTIFIER
13972      : (REFERENCE THE OINST TABLE) INTO THE FIRST WORD OF THE DUMMY
13973      : INPUT TABLE. THIS ROUTINE ONLY GENERATES CIS INSTS WHICH HAVE A NON-ZERO
13974      : ENCODING IN THE OINST TABLE. ZERO OINST TABLE ENTRIES ASSOC WITH CIS INST
13975      : NOT TO BE INCLUDED IN THE RANDOM EXERCISING.
13976      :
13977      :GENRI:
13977 062670      JSR PC,RN          ;GET A RANDOM #
13978 062670 004737 063464      BIC #BS128,R0      ;MASK OFF ALL BUT LEAST SIGNIF 5 BITS
13979 062674 042700 177740      INC R0
13980 062700 005200              ;VALID IDENTIFIERS = 1 TO 31
13981 062702 020027 000031      CMP R0,#31
13982 062706 101370              BHI GENRI          ;BRANCH IF IDENTIFIER IS INVALID
```

```

13984 062710 010037 072022      MOV R0, IDUM      ;LOAD INST IDENTIFIER INTO DUMMY INPUT TABLE
13985 062714 006300              ASL R0
13986 062716 062700 003722      ADD #OINST, R0   ;VERIFY THAT GENERATED INST IS
13987 062722 005710              TST (R0)         ;A MEMBER OF THE SET OF CIS INSTS
13988 062724 001761              BEQ GENRI        ;TO BE RANDOMLY EXERCISED.
13989 062726 000207              RTS PC
13990
13991      ;ROUTINE TO LOAD UP DUMMY INPUT TABLE USING RANDOM NUMBER GENERATOR.
13992      ;ROUTINE USES THE RANDOM EXERCISE MASK TABLES TO LIMIT OPERANDS
13993      ;(LENGTHS, ADDRESSES, ETC) TO THE PROPER RANGE.
13994
13995 062730 012702 072026      LDINPT: MOV #IDUM+4, R2      ;SETUP POINTER INTO DUMMY TABLE
13996 062734 013701 072022      MOV IDUM, R1
13997 062740 006301              ASL R1
13998 062742 062701 004172      ADD #MINST, R1
13999 062746 011101              MOV (R1), R1
14000 062750 012137 001764      1$: MOV (R1)+, PMASK      ;GET MASK FOR GIVEN INPUT PARAMETER
14001 062754 022737 125252 001764  CMP #EOT, PMASK  ;IS MASK=END OF MASK TABLE (EOT)
14002 062762 001431              BEQ IDFLD        ;BRANCH IF YES
14003 062764 022737 152525 001764  CMP #DSCPTR, PMASK ;DOES MASK INDICATE THAT INPUT
14004                                ;PARAMETER IS A DESCRIPTOR POINTER?
14005 062772 001003              BNE 2$          ;BRANCH IF NO
14006 062774 012722 002504      MOV #RANDSC, (R2)+ ;FILL TABLE ENTRY WITH A RANDOM
14007 063000 000763              BR 1$          ;DESCRIPTOR POINTER.
14008 063002 032737 100000 001764  2$: BIT #100000, PMASK ;MASK AND OFFSET?
14009 063010 001010              BNE 3$          ;BRANCH IF NO(MASK ONLY).
14010 063012 004737 063464      JSR PC, RN      ;GENERATE A RANDOM #
14012 063016 042700 176000      BIC #BS4, R0    ;MASK WITH 176000
14017 063022 063700 001764      ADD PMASK, R0  ;ADD IN OFFSET
14018 063026 010022              MOV R0, (R2)+  ;STORE INPUT PARAMETER
14019 063030 000747              BR 1$
14020 063032 004737 063464      3$: JSR PC, RN      ;GENERATE A RANDOM NUMBER
14021 063036 043700 001764      BIC PMASK, R0  ;MASK TO VALID RANGE
14022 063042 010022              MOV R0, (R2)+  ;STORE INPUT PARAMETER IN DUMMY TABLE
14023 063044 000747              BR 1$
14024 063046 005022      IDFLD: CLR (R2)+      ;CLEAR REMAINDER OF DUMMY INPUT TABLE
14025 063050 020227 072074      CMP R2, #IDUME
14026 063054 001374              BNE IDFLD
14027 063056 022737 000020 072022  CMP #20, IDUM   ;IS RANDOM MODE INST = ASHP OR ASHN?
14028 063064 001404              BEQ 1$          ;BRANCH IF YES
14029 063066 022737 000030 072022  CMP #30, IDUM
14030 063074 001007              BNE 2$          ;BRANCH IF NO
14031 063076 123727 072033 000011  1$: CMPB IDUM+11, #11 ;YES - LIMIT ROUND DIGIT TO 0 - 9
14032 063104 101403              BLOS 2$
14033 063106 142737 000010 072033  BICB #10, IDUM+11 ;CONVERT INVALID DIGIT TO A VALID ONE
14034 063114 022737 000003 072022  2$: CMP #3, IDUM   ;IS RANDOM INST = MOVTC
14035 063122 001006              BNE 3$          ;BRANCH IF NO
14036 063124 013737 072056 002172  MOV IDUM+34, IRXLT ;ADJUST IP15 FOR PROPER LEVEL OF INDIRECTING
14037 063132 012737 002172 072056  MOV #IRXLT, IDUM+34
14038 063140 000207              3$: RTS PC
14039
14040      ;ROUTINE TO LOAD MISCELLANEOUS CONSTANTS USING RANDOM NUMBER GENERATOR
14041
14042 063142      LDCON:

```

```

14043 063142 004737 063464      JSR PC,RN
14044 063146 010037 001660      MOV RO,INCSQ1      ;LOAD TEST BUFFER INCREMENTING SEQUENCE
14045 063152 004737 063464      JSR PC,RN          ; SEED WITH A RANDOM #
14046 063156 010037 001662      MOV RO,INCSQ2
14047 063162 000207      RTS PC
14048
14049
14050
14051      ;ROUTINE TO ACKNOWLEDGE OPERATOR REQUESTS
14052      CNTL T - DISPLAY CURRENT TEST #(DECIMAL) THEN RETURN TO CALL+6
14053      CNTL C - RETURN TO CALL+2
14054      CNTL D - SET DISPLAY AND NO QUERY SWITCH. THEN RETURN TO CALL + 6
14055      CNTL E - SET DISPLAY SWITCH. THEN RETURN TO CALL + 6
14056      CNTL N - CLEAR DISPLAY SWITCH. THEN RETURN TO CALL + 6
14057      CNTL O - TOGGLE PROGRESS DISPLAY SWITCH. THEN RETURN TO CALL +6
14058      OTHER - RETURN TO CALL+6
14059      EXTBK:
14060 063164 005737 110520      TST XOCHAR          ;DID TYPE ROUTINE XON/XOFF CHECK FIND A CHAR
14061 063170 001406      BEQ 7$             ;BRANCH IF NO
14062 063172 013737 110520 064672  MOV XOCHAR,RCHAR
14063 063200 005037 110520      CLR XOCHAR
14064 063204 000406      BR 8$
14065 063206 105777 116456      7$: TSTB @TKS          ;CHAR THERE?
14066 063212 100063      BPL 1$            ;NO - EXIT ROUTINE
14067 063214 117737 116472 064672  MOVB @TKB,RCHAR    ;READ AND SAVE TTY CHAR
14068 063222 042737 177600 064672  8$: BIC #^C177,RCHAR ;GET RID OF JUNK IF ANY
14069 063230 023727 064672 000003  CMP RCHAR,#003     ;IS CHAR A CNTL C?
14070 063236 001453      BEQ 2$            ;BRANCH IF YES
14071 063240 023727 064672 000024  CMP RCHAR,#024     ;IS CHAR A CNTL T?
14072 063246 001003      BNE 3$
14073 063250 004737 063370      JSR PC,IDINST
14074 063254 000442      BR 1$
14075 063256 023727 064672 000004  3$: CMP RCHAR,#004   ;IS CHAR A CNTL D?
14076 063264 001007      BNE 5$
14077 063266 012737 177777 002042  MOV #177777,NOERDS ;SET DISPLAY SWITCH
14078 063274 012737 177777 002204  MOV #177777,QRFLG  ;SET QUERY FOR DISPLAY BUFFER FLAG
14079 063302 000427      BR 1$
14080 063304 023727 064672 000005  5$: CMP RCHAR,#005   ;IS CHAR A CNTL E?
14081 063312 001004      BNE 6$            ;BRANCH IF NO
14082 063314 012737 177777 002042  MOV #177777,NOERDS ;SET DISPLAY SWITCH
14083 063322 000417      BR 1$
14084 063324 023727 064672 000017  6$: CMP RCHAR,#017   ;IS CHAR A CNTL O?
14085 063332 001003      BNE 4$            ;BRANCH IF NO
14086 063334 005137 002044      COM PROGD          ;TOGGLE PROGRESS DISPLAY SWITCH
14087 063340 000410      BR 1$
14088 063342 023727 064672 000016  4$: CMP RCHAR,#016   ;IS CHAR A CNTL N?
14089 063350 001004      BNE 1$
14090 063352 005037 002042      CLR NOERDS         ;YES - CLEAR NO ERROR DISPLAY SWITCH
14091 063356 005037 002204      CLR QRFLG          ;CLEAR QUERY SWITCH
14092 063362 062716 000004      1$: ADD #4,(SP)     ;RETURN TO CALL+6
14093 063366 000207      2$: RTS PC
14094
14095      ;ROUTINE TO DISPLAY CURRENT INST AND TEST #
14096

```



```

14097 063370 IDINST:
14098 063370 012737 177777 001762 MOV #177777,CTACT ;SET CONTROL T ACTIVE FLAG
14099 063376 PRINTB #FORM21 ;PRINT A CRLF
(6) 063376 012746 014020 MOV #FORM21,-(SP)
(3) 063402 010600 MOV SP,R0
(4) 063404 004737 065304 JSR PC,FPRINT
14100 063410 004777 116526 JSR PC,@EMPTR ;PRINT INST & TEST #
14101 063414 000207 RTS PC
14102
14103
14104 ;ROUTINE TO RANDOMIZE PACKED STRING DATA TYPE
14105
14106 RPTYPE:
14107 063416 004737 063464 JSR PC,RN ;GET A RANDOM #
14108 063422 032700 000001 BIT #1,R0 ;USE BIT 0 OF THE RANDOM # TO SELECT BETWEEN
14109 063426 001403 BEQ 1$ ; THE TWO TYPES FOR PACKED STRINGS (6,7).
14110 063430 012700 000007 MOV #7,R0
14111 063434 000402 BR 10$
14112 063436 012700 000006 1$: MOV #6,R0
14113 063442 000207 10$: RTS PC
14114
14115 ;ROUTINE TO RANDOMIZE ZONED STRING DATA TYPES
14116
14117 RZTYPE:
14118 063444 004737 063464 JSR PC,RN ;GET A RANDOM #
14119 063450 042700 177770 BIC #177770,R0 ;USE BITS 0,1 & 2 TO SELECT BETWEEN
14120 063454 020027 000005 CMP R0,#5 ; THE 6 TYPES FOR ZONED STRINGS
14121 063460 101371 BHI RZTYPE
14122 063462 000207 RTS PC
14123
14124 ;ROUTINE TO GENERATE A PSEUDO RANDOM NUMBER
14125
14126 :INPUTS: NONE
14127 :OUTPUTS: PSEUDO RANDOM VALUE IN R0
14128
14129
14130 RN:
14131 063464 013700 063546 MOV RP1,R0
14132 063470 000241 CLC
14133 063472 005337 063544 DEC RNCON
14134 063476 006100 ROL R0
14135 063500 006100 ROL R0
14136 063502 063700 063544 ADD RNCON,R0
14137 063506 063700 063550 ADD RP2,R0
14138 063512 010037 063546 MOV R0,RP1
14139 063516 006100 ROL R0
14140 063520 006100 ROL R0
14141 063522 063700 063550 ADD RP2,R0
14142 063526 006100 ROL R0
14143 063530 006100 ROL R0
14144 063532 010037 063550 MOV R0,RP2
14145 063536 013700 063546 MOV RP1,R0
14146 063542 000207 RTS PC
14147 063544 000000 RNCON: .WORD 0 ;RANDOM # GENERATOR SEEDS

```

14148 063546 001233
14149 063550 007622
14150 063552 000000
14151 063554 001233
14152 063556 007622
14153

RP1: .WORD 1233
RP2: .WORD 7622
KRNCON: .WORD 0
KRP1: .WORD 1233
KRP2: .WORD 7622

```

14155      .SBTTL      MESSAGE PRINT ROUTINES
14156      :+++++
14157
14158      :ERROR MESSAGE PRINT ROUTINES
14159      :           RETURNS TO CALL +2 FOR REPEAT TEST
14160      :           RETURNS TO CALL +6 FOR NORMAL RETURN
14161      :
14162      :-----
14163      :INSERR:
14164      063560 004777 116356      JSR    PC,@EMPTR      ;PRINT ERROR MESSAGE HEADER
14165      063564 032737 000100 047464  BIT    #100,TINST    ;INST UNDER TEST TYPE?
14166      063572 001406          BEQ    11$           ;BRANCH IF REGISTER TYPE
14167      063574          PRINTB #INMEM      ;IN-LINE TYPE
14168      (6) 063574 012746 011114  MOV    #INMEM,-(SP)
14169      (3) 063600 010600          MOV    SP,R0
14170      (4) 063602 004737 065304  JSR    PC,FPRINT
14171      063606 000405          BR     12$
14172      (6) 063610 012746 011066 11$:  PRINTB #INREG
14173      (3) 063614 010600          MOV    #INREG,-(SP)
14174      (4) 063616 004737 065304  MOV    SP,R0
14175      063622          JSR    PC,FPRINT
14176      (14) 063622 005046 12$:  PRINTB #FORM13,ER0,ER1,ER2,ER3,ER4,ER5,TR6,<B,TCC>
14177      (14) 063624 153716 003642  CLR    -(SP)
14178      (13) 063630 013746 003640  BISB  TCC,(SP)
14179      (12) 063634 013746 003676  MOV    TR6,-(SP)
14180      (11) 063640 013746 003674  MOV    ER5,-(SP)
14181      (10) 063644 013746 003672  MOV    ER4,-(SP)
14182      (9) 063650 013746 003670  MOV    ER3,-(SP)
14183      (8) 063654 013746 003666  MOV    ER2,-(SP)
14184      (7) 063660 013746 003664  MOV    ER1,-(SP)
14185      (6) 063664 012746 011170  MOV    ER0,-(SP)
14186      (3) 063670 010600          MOV    #FORM13,-(SP)
14187      (4) 063672 004737 065304  MOV    SP,R0
14188      063676          JSR    PC,FPRINT
14189      (6) 063676 012746 011142  PRINTB #EMOUT
14190      (3) 063702 010600          MOV    #EMOUT,-(SP)
14191      (4) 063704 004737 065304  MOV    SP,R0
14192      063710          JSR    PC,FPRINT
14193      (14) 063710 005046  PRINTB #FORM14,ER0R,ER1R,ER2R,ER3R,ER4R,ER5R,ER6R,<B,ECCR>
14194      (14) 063712 153716 003720  CLR    -(SP)
14195      (13) 063716 013746 003716  BISB  ECCR,(SP)
14196      (12) 063722 013746 003714  MOV    ER6R,-(SP)
14197      (11) 063726 013746 003712  MOV    ER5R,-(SP)
14198      (10) 063732 013746 003710  MOV    ER4R,-(SP)
14199      (9) 063736 013746 003706  MOV    ER3R,-(SP)
14200      (8) 063742 013746 003704  MOV    ER2R,-(SP)
14201      (7) 063746 013746 003702  MOV    ER1R,-(SP)
14202      (6) 063752 012746 011250  MOV    ER0R,-(SP)
14203      (3) 063756 010600          MOV    #FORM14,-(SP)
14204      (4) 063760 004737 065304  MOV    SP,R0
14205      063764          JSR    PC,FPRINT
14206      063770 005737 002146  TST   ERRREG      ;WAS THERE A REGISTER ERROR?
14207      063772 001440          BEQ    1$           ;YES - PRINT OUT DISCREPANCIES
14208      PRINTB #ACOUT

```

| | | | | | | |
|-------|--------|--------|--------|--------|--|--|
| (6) | 063772 | 012746 | 011330 | MOV | #ACOUT,-(SP) | |
| (3) | 063776 | 010600 | | MOV | SP,R0 | |
| (4) | 064000 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14176 | 064004 | 012701 | 003644 | MOV | #TROR,R1 | |
| 14177 | 064010 | 012702 | 003702 | MOV | #EROR,R2 | |
| 14178 | 064014 | 021122 | | 5\$: | CMP | (R1),(R2)+ ;COMPARE ACTUAL WITH EMULATOR REGS. |
| 14179 | 064016 | 001412 | | | BEQ | 2\$ |
| 14180 | 064020 | 011137 | 002264 | MOV | (R1),TERR ;NOT EQUAL - PRINT ACTUAL | |
| 14181 | 064024 | | | PRINTB | #FORM15,TERR | |
| (7) | 064024 | 013746 | 002264 | MOV | TERR,-(SP) | |
| (6) | 064030 | 012746 | 011356 | MOV | #FORM15,-(SP) | |
| (3) | 064034 | 010600 | | MOV | SP,R0 | |
| (4) | 064036 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14182 | 064042 | 000405 | | BR | 3\$ | |
| 14183 | 064044 | | | 2\$: | PRINTB | #FORM16 ;EQUAL - PRINT SPACES |
| (6) | 064044 | 012746 | 011365 | MOV | #FORM16,-(SP) | |
| (3) | 064050 | 010600 | | MOV | SP,R0 | |
| (4) | 064052 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14184 | 064056 | 020127 | 003660 | 3\$: | CMP | R1,#TR6R ;ALL REGISTERS COMPARED? |
| 14185 | 064062 | 001424 | | | BEQ | 4\$;BRANCH IF YES |
| 14186 | 064064 | 062701 | 000002 | | ADD | #2,R1 |
| 14187 | 064070 | 000751 | | | BR | 5\$;LOOK AT NEXT REGISTER |
| 14188 | 064072 | 005737 | 002144 | 1\$: | TST | ERRCC ;WAS THERE A CONDITION CODE ERROR? |
| 14189 | 064076 | 001431 | | | BEQ | 6\$ |
| 14190 | 064100 | | | | PRINTB | #ACOUT ;YES - PRINT ACTUAL COND. CODES |
| (6) | 064100 | 012746 | 011330 | MOV | #ACOUT,-(SP) | |
| (3) | 064104 | 010600 | | MOV | SP,R0 | |
| (4) | 064106 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14191 | 064112 | | | PRINTB | #FORM17,<B,TCCR> | |
| (7) | 064112 | 005046 | | CLR | -(SP) | |
| (7) | 064114 | 153716 | 003662 | BISB | TCCR,(SP) | |
| (6) | 064120 | 012746 | 011371 | MOV | #FORM17,-(SP) | |
| (3) | 064124 | 010600 | | MOV | SP,R0 | |
| (4) | 064126 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14192 | 064132 | 000413 | | BR | 6\$ | |
| 14193 | 064134 | 005737 | 002144 | 4\$: | TST | ERRCC |
| 14194 | 064140 | 001410 | | | BEQ | 6\$ |
| 14195 | 064142 | | | | PRINTB | #FORM18,<B,TCCR> |
| (7) | 064142 | 005046 | | CLR | -(SP) | |
| (7) | 064144 | 153716 | 003662 | BISB | TCCR,(SP) | |
| (6) | 064150 | 012746 | 011401 | MOV | #FORM18,-(SP) | |
| (3) | 064154 | 010600 | | MOV | SP,R0 | |
| (4) | 064156 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14196 | 064162 | 004737 | 066242 | 6\$: | JSR | PC,PRNIB ;GO CHECK FOR POSSIBLE NIBBLE PRINTOUT. |
| 14197 | 064166 | 005037 | 002270 | | CLR | FILLS2 |
| 14198 | 064172 | 005737 | 002150 | | TST | ERRBUF ;WAS THERE A BUFFER ERROR? |
| 14199 | 064176 | 001422 | | | BEQ | LODT |
| 14200 | 064200 | | | | PRINTB | #EBUFO,EMADR,EMDTA ;YES PRINT FIRST BUFFER |
| (8) | 064200 | 013746 | 002202 | MOV | EMDTA,-(SP) | |
| (7) | 064204 | 013746 | 002200 | MOV | EMADR,-(SP) | |
| (6) | 064210 | 012746 | 011405 | MOV | #EBUFO,-(SP) | |
| (3) | 064214 | 010600 | | MOV | SP,R0 | |
| (4) | 064216 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14201 | 064222 | | | PRINTB | #ABUFO,AEADR,AEDTA ; BYTE DISCREPANCY. | |

| | | | | | | | |
|-------|--------|--------|--------|---------|---|------------------------------------|--------------------|
| (8) | 064222 | 013746 | 002176 | MOV | AEDTA,-(SP) | | |
| (7) | 064226 | 013746 | 002174 | MOV | AEADR,-(SP) | | |
| (6) | 064232 | 012746 | 011442 | MOV | #ABUFO,-(SP) | | |
| (3) | 064236 | 010600 | | MOV | SP,R0 | | |
| (4) | 064240 | 004737 | 065304 | JSR | PC,FPRINT | | |
| 14202 | 064244 | | | LODT: | | | |
| 14203 | 064244 | 005737 | 002204 | TST | QRYFLG | :INHIBIT BUFFER QUERY? | |
| 14204 | 064250 | 001100 | | BNE | ERMDON | :YES | |
| 14205 | 064252 | 104400 | | TYPE | | | |
| 14206 | 064254 | 011473 | | QDISP | | | |
| 14207 | 064256 | 004737 | 064460 | JSR | PC,YORN | :DISPLAY BUFFER? | |
| 14208 | 064262 | 000137 | 064452 | JMP | ERMDON | :CONTINUE (C) RETURN | |
| 14209 | 064266 | 000137 | 064302 | JMP | 1\$ | :DISPLAY MEMORY (D) RETURN | |
| 14210 | 064272 | 000137 | 064456 | JMP | LERMD | :REPEAT TEST (R) RETURN | |
| 14211 | 064276 | 000137 | 064446 | JMP | RTSUPV | :RESTART (S) RETURN | |
| 14212 | 064302 | 104400 | | 1\$: | TYPE | :PRINT 'ADDR(S)?' | |
| 14213 | 064304 | 013627 | | AST | | | |
| 14214 | 064306 | 004737 | 064674 | JSR | PC,RANGE | :GET RANGE OF LOCATIONS TO DISPLAY | |
| 14215 | 064312 | 000754 | | BR | LODT | :NO MORE DISPLAY REQUESTED-RETURN | |
| 14216 | 064314 | | | PRINTB | #ADDHDR | :PRINT BYTE HEADER | |
| (6) | 064314 | 012746 | 013701 | MOV | #ADDHDR,-(SP) | | |
| (3) | 064320 | 010600 | | MOV | SP,R0 | | |
| (4) | 064322 | 004737 | 065304 | JSR | PC,FPRINT | | |
| 14217 | 064326 | 004737 | 065230 | 2\$: | JSR | PC,FILLPB | :FILL PRINT BUFFER |
| 14218 | 064332 | 000763 | | BR | 1\$ | :RANGE EXHAUSTED RETURN | |
| 14219 | 064334 | | | PRINTB | #FORM19,BAD,<B,PB0>,<B,PB1>,<B,PB2>,<B,PB3> | | |
| (11) | 064334 | 005046 | | CLR | -(SP) | | |
| (11) | 064336 | 153716 | 003141 | BISB | PB3,(SP) | | |
| (10) | 064342 | 005046 | | CLR | -(SP) | | |
| (10) | 064344 | 153716 | 003140 | BISB | PB2,(SP) | | |
| (9) | 064350 | 005046 | | CLR | -(SP) | | |
| (9) | 064352 | 153716 | 003137 | BISB | PB1,(SP) | | |
| (8) | 064356 | 005046 | | CLR | -(SP) | | |
| (8) | 064360 | 153716 | 003136 | BISB | PB0,(SP) | | |
| (7) | 064364 | 013746 | 002222 | MOV | BAD,-(SP) | | |
| (6) | 064370 | 012746 | 013642 | MOV | #FORM19,-(SP) | | |
| (3) | 064374 | 010600 | | MOV | SP,R0 | | |
| (4) | 064376 | 004737 | 065304 | JSR | PC,FPRINT | | |
| 14220 | 064402 | | | PRINTB | #FORM20,<B,PB4>,<B,PB5>,<B,PB6>,<B,PB7> | | |
| (10) | 064402 | 005046 | | CLR | -(SP) | | |
| (10) | 064404 | 153716 | 003145 | BISB | PB7,(SP) | | |
| (9) | 064410 | 005046 | | CLR | -(SP) | | |
| (9) | 064412 | 153716 | 003144 | BISB | PB6,(SP) | | |
| (8) | 064416 | 005046 | | CLR | -(SP) | | |
| (8) | 064420 | 153716 | 003143 | BISB | PB5,(SP) | | |
| (7) | 064424 | 005046 | | CLR | -(SP) | | |
| (7) | 064426 | 153716 | 003142 | BISB | PB4,(SP) | | |
| (6) | 064432 | 012746 | 013765 | MOV | #FORM20,-(SP) | | |
| (3) | 064436 | 010600 | | MOV | SP,R0 | | |
| (4) | 064440 | 004737 | 065304 | JSR | PC,FPRINT | | |
| 14221 | 064444 | 000730 | | BR | 2\$ | | |
| 14222 | 064446 | 062716 | 000004 | RTSUPV: | ADD #4,(SP) | :RETURN TO RESTART AT LOC 'START' | |
| 14223 | 064452 | 062716 | 000004 | ERMDON: | ADD #4,(SP) | :NORMAL RETURN TO CALL +6 | |
| 14224 | 064456 | 000207 | | LERMD: | RTS PC | :REPEAT TEST RETURN TO CALL +2 | |

14225
14226
14227
14228
14229 064460
14230 064460 105777 115204
14231 064464 100375
14232 064466 117737 115220 064672
14233 064474 042737 177600 064672
14234 064502 023727 064672 000123
14235 064510 001450
14236 064512 023727 064672 000110
14237 064520 001003
14238 064522 005037 047444
14239
14240 064526 000407

:
:SUBROUTINE TO ACCEPT Y,N,C,R,S,D OR H RESPONSE FROM TTY. RETURNS TO CALL +2
: ON N OR C RESPONSE; CALL +4 ON A Y OR D RESPONSE; CALL +6 ON AN R OR H RESPONSE;
: AND CALL +10 ON AN S RESPONSE.

YORN:

1\$: TSTB @TKS ;WAIT FOR A CHARACTER
BPL 1\$
MOVB @TKB,RCHAR ;READ & SAVE CHAR
BIC #^C177,RCHAR ;GET RID OF JUNK IF ANY
CMP RCHAR,#123 ;IS CHAR AN S ?
BEQ 5\$;BRANCH IF YES
CMP RCHAR,#110 ;IS CHAR A H
BNE 6\$;BRANCH IF NO
CLR PREINS ;INSERT A HALT IMMEDIATELY BEFORE
; THE CIS INST UNDER TEST. THEN REPEAT TEST.
BR 10\$

| | | | | | | | | |
|-------|--------|--------|--------|--------|-------|-----|----------------|---|
| 14242 | 064530 | 023727 | 064672 | 000122 | 6\$: | CMP | RCHAR,#122 | :IS CHAR R |
| 14243 | 064536 | 001011 | | | | BNE | 7\$ | :BRANCH IF NO |
| 14244 | 064540 | 013737 | 001672 | 047444 | | MOV | KNOP,PREINS | :RESTORE NOP TO INST IMMED BEFOR CIS INST UNDER TEST |
| 14245 | 064546 | 012737 | 177777 | 002266 | 10\$: | MOV | #177777,RPTFLG | :SET REPEAT TEST FLAG |
| 14246 | 064554 | 005337 | 002054 | | | DEC | ERRCT | :DECREMENT ERROR COUNT SO THAT ERROR COUNT |
| 14247 | | | | | | | | : DOESN'T ADVANCE ON REPEAT OF TEST |
| 14248 | 064560 | 000426 | | | | BR | 4\$ | |
| 14249 | 064562 | 023727 | 064672 | 000131 | 7\$: | CMP | RCHAR,#131 | :IS CHAR = Y |
| 14250 | 064570 | 001424 | | | | BEQ | 2\$ | |
| 14251 | 064572 | 023727 | 064672 | 000104 | | CMP | RCHAR,#104 | :IS CHAR = D? |
| 14252 | 064600 | 001420 | | | | BEQ | 2\$ | |
| 14253 | 064602 | 023727 | 064672 | 000116 | | CMP | RCHAR,#116 | :NO - IS CHAR = N |
| 14254 | 064610 | 001404 | | | | BEQ | 11\$ | |
| 14255 | 064612 | 023727 | 064672 | 000103 | | CMP | RCHAR,#103 | :IS CHAR = C? |
| 14256 | 064620 | 001317 | | | | BNE | 1\$ | |
| 14257 | 064622 | 013737 | 001672 | 047444 | 11\$: | MOV | KNOP,PREINS | :RESTORE NOP TO INST IMMED BEFORE CIS INST UNDER TEST |
| 14258 | 064630 | 000406 | | | | BR | 3\$ | :YES - RETURN - CALL +2 |
| 14259 | 064632 | 062716 | 000004 | | 5\$: | ADD | #4,(SP) | |
| 14260 | 064636 | 062716 | 000004 | | 4\$: | ADD | #4,(SP) | |
| 14261 | 064642 | 062716 | 000004 | | 2\$: | ADD | #4,(SP) | :CHAR = Y OR D SETUP RETURN - CALL +4 |
| 14262 | 064646 | 004737 | 064654 | | 3\$: | JSR | PC,ECHAR | :ECHO CHARACTER - WAIT FOR |

| | | | | | | | |
|-------|--------|--------|--------|--------|------|------------|--------------------------------------|
| 14264 | 064652 | 000207 | | RTS | PC | : | PRINTER READY |
| 14265 | | | | | | : | LOAD CHAR TO BE TYPED INTO DATA REG. |
| 14266 | 064654 | 105777 | 115034 | ECHAR: | TSTB | @TPS | SUBROUTINE TO PRINT CHAR IN 'RCHAR' |
| 14267 | 064660 | 100375 | | | BPL | ECHAR | WAIT UNTIL PRINTER IS READY |
| 14268 | 064662 | 113777 | 064672 | 115026 | MOVB | RCHAR,@TPB | LOAD CHAR INTO DATA REG |
| 14269 | 064670 | 000207 | | | RTS | PC | |


```

14271
14272 064672 000000          RCHAR: .WORD 0
14273
14274          :SUBROUTINE TO GET RANGE OF LOCATIONS TO DISPLAY.
14275          : RETURNS TO CALL +2 ON NO MORE DISPLAY REQUESTED - USER
14276          : RESPONDED WITH 'C'.
14277          : NORMAL RETURN IS TO CALL +4 WITH LOWER DISPLAY LIMIT
14278          : IN 'RLL' AND UPPER DISPLAY LIMIT IN 'RUL'.
14279 064674          RANGE:          :NORMAL RETURN = CALL +4.
14280 064674 004737 064776          JSR      PC,ACCOCT          :GET RANGE LOWER LIMIT
14281 064700 000207          RTS      PC          :RETURN - EXIT DISPLAY
14282 064702 000411          BR      1$          :RETURN - SINGLE LIMIT SPECIFIED
14283 064704 012637 002216          MOV      (SP)+,RLL          :NORMAL RETURN - SAVE LOWER LIMIT
14284 064710 004737 064776          JSR      PC,ACCOCT          :GET RANGE UPPER LIMIT
14285 064714 000207          RTS      PC          :RETURN - EXIT DISPLAY
14286 064716 000411          BR      2$          :NORMAL RETURN - SAVE UPPER LIMIT
14287 064720 104400          TYPE          :RETURN - TYPE? <CR><LF>*
14288 064722 015476          QUES
14289 064724 000763          BR      RANGE          :TRY AGAIN
14290 064726 012637 002216 1$: MOV      (SP)+,RLL          :SINGLE LIMIT SPECIFIED - SAVE AS
14291 064732 013737 002216 002220 MOV      RLL,RUL          : BOTH LOWER & UPPER LIMIT
14292 064740 000402          BR      3$          :EXIT
14293 064742 012637 002220 2$: MOV      (SP)+,RUL          :SAVE UPPER LIMIT
14294 064746 042737 000007 002216 3$: BIC      #7,RLL          :ROUND OFF RANGE TO GROUP OF
14295 064754 042737 000007 002220 BIC      #7,RUL          :TEN BYTES.
14296 064762 062737 000010 002220 ADD      #10,RUL
14297 064770 062716 000002 ADD      #2,(SP)          :EXIT TO CALL +4
14298 064774 000207          RTS      PC
14299
14300          :SUBROUTINE TO ACCEPT OCTAL # FROM TTY. RETURNS TO CALL +2 ON INITIAL CR.
14301          : RETURNS TO CALL +4 ON <CR> OR / WITH LIMIT ON STACK. RETURNS TO CALL +6 ON <-> WITH
14302          : LIMIT ON STACK.
14303 064776          ACCOCT:
14304 064776 005046          CLR      -(SP)          :CLEAR STORAGE FOR OCTAL #
14305 065000 105777 114664 1$: TSTB    @TKS          :CHAR THERE?
14306 065004 100375          BPL      1$          :NO - WAIT
14307 065006 117746 114700          MOVB    @TKB,-(SP)          :SAVE THE CHAR
14308 065012 042716 177600          BIC      #^C177,(SP)          :STRIP-OFF THE ASCII
14309 065016 022726 000015          CMP     #15,(SP)+          :IS IT A 'CR'?
14310 065022 001005          BNE     2$
14311 065024 104400          TYPE          :YES - ECHO CR & LF
14312 065026 015512          XCRLF
14313 065030 062706 000002          ADD     #2,SP          :RETURN TO CALL +2
14314 065034 000207          RTS     PC
14315 065036 024627 000055 2$: CMP     -(SP),#55          :IS CHAR = '-'
14316 065042 001462          BEQ     6$          :BRANCH IF YES
14317 065044 021627 000057 3$: CMP     (SP),#57          :IS CHAR A / ?
14318 065050 001403          BEQ     31$
14319 065052 021627 000015          CMP     (SP),#15          :IS CHAR A <CR>?
14320 065056 001016          BNE     4$
14321 065060 104400          31$: TYPE          :YES - ECHO/<CR> AND <LF>
14322 065062 015515          SLCRLF
14323 065064 016616 000002 7$: MOV     2(SP),(SP)          :SWAP POSITION OF OCTAL #
14324 065070 016666 000004 000002 MOV     4(SP),2(SP)          :AND RETURN PC ON STACK

```

```

14325 065076 011666 000004      MOV      (SP),4(SP)
14326 065102 062706 000002      ADD      #2,SP
14327 065106 062716 000002      ADD      #2,(SP)      ;UPDATE RETURN POINTER
14328 065112 000207      RTS      PC           ;RETURN WITH OCTAL LIMIT ON STACK
14329 065114 011637 064672      4$:     MOV      (SP),RCHAR ;ECHO CHAR ACCEPTED
14330 065120 004737 064654      JSR      PC,ECHAR
14331
14332 065124 021627 000060      CMP      (SP),#60     ;CHAR <0?
14333 065130 002422      BLT      5$           ;BRANCH IF YES
14334 065132 021627 000067      CMP      (SP),#67     ;CHAR>7?
14335 065136 003017      BGT      5$           ;BRANCH IF YES
14336 065140 042726 000060      BIC      #60,(SP)+    ;STRIP OFF ASCII
14337 065144 006316      ASL      (SP)         ;SHIFT PRESENT DATA OVER TO
14338 065146 006316      ASL      (SP)         ; MAKE ROOM FOR NEW DIGIT
14339 065150 006316
14340 065152 056616 177776      BIS      -2(SP),(SP) ;SET IN NEW DIGIT
14341 065156 105777 114506      10$:    TSTB     @TKS       ;CHAR THERE
14342 065162 100375      BPL      10$         ;NO - WAIT
14343 065164 117746 114522      MOVB     @TKB,-(SP)  ;SAVE CHAR
14344 065170 042726 177600      BIC      #^C177,(SP)+
14345 065174 000720      BR      2$
14346 065176 104400      5$:     TYPE      ;TYPE ?<CR><LF>*
14347 065200 015476      QUES
14348 065202 062706 000004      ADD      #4,SP
14349 065206 000674      BR      1$
14350 065210 011637 064672      6$:     MOV      (SP),RCHAR ;ECHO '-'
14351 065214 004737 064654      JSR      PC,ECHAR
14352 065220 062766 000002 000004      ADD      #2,4(SP)   ;UPDATE RETURN POINTER
14353 065226 000716      BR      7$
14354
14355      ;SUBROUTINE TO FILL BYTE PRINT BUFFER. RETURNS TO CALL +2 WHEN DISPLAY REQUEST IS
14356      ;COMPLETE (RLL=RUL). NORMAL RETURN TO CALL+4 WITH RLL=RLL +10 & PRINT BUFFER FILLED.
14357 065230 023737 002216 002220  FILLPB:  CMP      RLL,RUL
14358 065236 001421      BEQ      1$         ; NORMAL RETURN TO CALL +4
14359 065240 013701 002216      MOV      RLL,R1     ;SETUP POINTER TO DISPLAY LOCS
14360 065244 010137 002222      MOV      R1,BAD     ;SAVE BUFFER ADDRESS FOR PRINTOUT
14361 065250 012137 003136      MOV      (R1)+,PB0  ;TRANSFER 10 BYTES AT DISPLAY
14362 065254 012137 003140      MOV      (R1)+,PB2  ; LOC ADDRESS TO PRINT
14363 065260 012137 003142      MOV      (R1)+,PB4  ; BUFFER.
14364 065264 012137 003144      MOV      (R1)+,PB6
14365 065270 062737 000010 002216      ADD      #10,RLL    ;UPDATE LOWER LIMIT DISPLAY POINTER
14366 065276 062716 000002      ADD      #2,(SP)   ;UPDATE RETURN POINTER
14367 065302 000207      1$:     RTS      PC
14368
14369      .EVEN
14370
14371      ;SUBROUTINE TO TYPE FORMATED 'PRINTB' STATEMENTS
14372      ;
14373 065304      FPRINT:
14374 065304 010537 002062      MOV      R5,FSAVR5  ;SAVE REGISTERS
14375 065310 010437 002064      MOV      R4,FSAVR4
14376 065314 010337 002066      MOV      R3,FSAVR3
14377 065320 010237 002070      MOV      R2,FSAVR2
14378 065324 010137 002072      MOV      R1,FSAVR1
  
```

| | | | | | | | |
|-------|--------|--------|--------|--------|----------------|--|-----------------------------|
| 14379 | 065330 | 012001 | | | MOV (R0)+,R1 | :SETUP R1 AS POINTER INTO FORMAT STATEMENT | |
| 14380 | 065332 | 112102 | | 1\$: | MOVB (R1)+,R2 | :GET NEXT FORMAT BYTE | |
| 14381 | 065334 | 020227 | 000045 | | CMP R2,#'%' | :IS BYTE = % ? | |
| 14382 | 065340 | 001774 | | | BEQ 1\$ | :BRANCH IF YES | |
| 14383 | 065342 | 020227 | 000117 | | CMP R2,#'0' | :IS BYTE = 0 ? (OCTAL) | |
| 14384 | 065346 | 001426 | | | BEQ 2\$ | :BRANCH IF YES | |
| 14385 | 065350 | 020227 | 000101 | | CMP R2,#'A' | :IS BYTE = A ? (ASCII) | |
| 14386 | 065354 | 001432 | | | BEQ 3\$ | :BRANCH IF YES | |
| 14387 | 065356 | 020227 | 000116 | | CMP R2,#'N' | :IS BYTE = N ? (CRLF) | |
| 14388 | 065362 | 001443 | | | BEQ 4\$ | :BRANCH IF YES | |
| 14389 | 065364 | 020227 | 000131 | | CMP R2,#'Y' | :IS BYTE = Y ? (BINARY) | |
| 14390 | 065370 | 001446 | | | BEQ 5\$ | :BRANCH IF YES | |
| 14391 | 065372 | 020227 | 000104 | | CMP R2,#'D' | :IS BYTE = D ? (DECIMAL) | |
| 14392 | 065376 | 001452 | | | BEQ 6\$ | :BRANCH IF YES | |
| 14393 | 065400 | 020227 | 000123 | | CMP R2,#'S' | :IS BYTE = S ? (SPACE) | |
| 14394 | 065404 | 001463 | | | BEQ 7\$ | :BRANCH IF YES | |
| 14395 | 065406 | 020227 | 000000 | | CMP R2,#0 | :IS BYTE = 0 ? (END OF FORMAT STATEMENT) | |
| 14396 | 065412 | 001512 | | | BEQ 10\$ | :BRANCH IF YES | |
| 14397 | 065414 | 020227 | 000102 | | CMP R2,#'B' | :IS BYTE = B ? (BYTE) | |
| 14398 | 065420 | 001524 | | | BEQ 11\$ | :BRANCH IF YES | |
| 14399 | 065422 | 000743 | | | BR 1\$ | :BYTE = NONE OF THE ABOVE - IGNORE IT. | |
| 14400 | | | | | | | |
| 14401 | 065424 | 112102 | | 2\$: | MOVB (R1)+,R2 | :SET R2 = COUNT OF # OF DIGITS TO PRINT | |
| 14402 | 065426 | 042702 | 177770 | | BIC #177770,R2 | | |
| 14403 | 065432 | 012003 | | | MOV (R0)+,R3 | :SET R3 = WORD OF DIGITS TO PRINT | |
| 14404 | 065434 | 004737 | 065714 | | JSR PC,POCT | :CALL ROUTINE TO PRINT OCTAL DIGITS | |
| 14405 | 065440 | 000734 | | | BR 1\$ | | |
| 14406 | 065442 | 112102 | | 3\$: | MOVB (R1)+,R2 | :SET R2 = NEXT ASCII CHAR TO PRINT | |
| 14407 | 065444 | 022702 | 000045 | | CMP #'%,R2 | :IS CHAR = % | |
| 14408 | 065450 | 001730 | | | BEQ 1\$ | :BRANCH IF YES | |
| 14409 | 065452 | 022702 | 000000 | | CMP #0,R2 | :END OF FORMAT BYTES? | |
| 14410 | 065456 | 001470 | | | BEQ 10\$ | :BRANCH IF YES | |
| 14411 | 065460 | 110237 | 066016 | | MOVB R2,TDIG | :YES - PREPARE TO EXIT ROUTINE | |
| 14412 | 065464 | 104400 | 066016 | | TYPE ,TDIG | :CALL ROUTINE TO PRINT ASCII BYTE | |
| 14413 | 065470 | 000764 | | | BR 3\$ | | |
| 14414 | 065472 | 012737 | 000200 | 066016 | 4\$: | MOV #CRLF,TDIG | :CALL ROUTINE TO PRINT CRLF |
| 14415 | 065500 | 104400 | 066016 | | TYPE ,TDIG | | |
| 14416 | 065504 | 000712 | | | BR 1\$ | | |
| 14417 | 065506 | 112102 | | 5\$: | MOVB (R1)+,R2 | :SET R2 = COUNT OF # OF DIGITS TO PRINT | |
| 14418 | 065510 | 042702 | 177770 | | BIC #177770,R2 | | |
| 14419 | 065514 | 012003 | | | MOV (R0)+,R3 | :SET R3 = WORD OF DIGITS TO PRINT | |
| 14420 | 065516 | 004737 | 066020 | | JSR PC,PBIN | :CALL ROUTINE TO PRINT BINARY DIGITS | |
| 14421 | 065522 | 000703 | | | BR 1\$ | | |
| 14422 | 065524 | 112102 | | 6\$: | MOVB (R1)+,R2 | :SET R2 = COUNT OF DIGITS TO PRINT | |
| 14423 | 065526 | 042702 | 177770 | | BIC #177770,R2 | | |
| 14424 | 065532 | 020227 | 000006 | | CMP R2,#6 | :IF REQUEST IS TO PRINT MORE THAN 5 DIGITS | |
| 14425 | 065536 | 103402 | | | BLO 61\$ | : PRINT 5 INSTEAD | |
| 14426 | 065540 | 012702 | 000005 | | MOV #5,R2 | | |
| 14427 | 065544 | 012003 | | 61\$: | MOV (R0)+,R3 | :SET R3 = WORD OF DIGITS TO PRINT | |
| 14428 | 065546 | 004737 | 066112 | | JSR PC,PDEC | :CALL ROUTINE TO CONVERT (R3) TO DECIMAL | |
| 14429 | 065552 | 000667 | | | BR 1\$ | : AND PRINT DECIMAL DIGITS | |
| 14430 | 065554 | 112102 | | 7\$: | MOVB (R1)+,R2 | :GET MOST SIGN DIGIT OF 1 OR 2 DIGIT | |
| 14431 | | | | | | : COUNT OF # OF SPACES TO PRINT | |
| 14432 | 065556 | 042702 | 177770 | | BIC #177770,R2 | | |

```
14433 065562 121127 000045      CMPB (R1),#'%           ;IS NEXT BYTE = % ?
14434 065566 001415      BEQ 12$
14435 065570 121127 000000      CMPB (R1),#0
14436 065574 001412      BEQ 12$
14437 065576 006302      ASL R2
14438 065600 006302      ASL R2
14439 065602 006302      ASL R2
14440 065604 112137 066014      MOVB (R1)+,OCNT
14441 065610 142737 000370 066014      BICB #370,OCNT
14442 065616 153702 066014      BISB OCNT,R2           ;GET LEAST SIGN DIGIT INTO R2
14443 065622 012737 000040 066016 12$:      MOV #' ,TDIG          ;PRINT A SPACE
14444 065630 104400 066016      TYPE ,TDIG
14445 065634 077206      SOB R2,12$
14446 065636 000635      BR 1$
14447 065640 011640      10$:      MOV (SP),-(R0)
14448 065642 010006      MOV R0,SP
14449 065644 013705 002062      MOV FSAVR5,R5
14450 065650 013704 002064      MOV FSAVR4,R4
14451 065654 013703 002066      MOV FSAVR3,R3
14452 065660 013702 002070      MOV FSAVR2,R2
14453 065664 013701 002072      MOV FSAVR1,R1
14454 065670 000207      RTS PC
14455
14456 065672 112102      11$:      MOVB (R1)+,R2           ;SET R2 = COUNT OF # OF DIGITS TO PRINT
14457 065674 042702 177770      BIC #177770,R2
14458 065700 012003      MOV (R0)+,R3           ;SET R3 = BYTE TO PRINT
14459 065702 042703 177400      BIC #177400,R3
14460 065706 004737 065714      JSR PC,POCT
14461 065712 000607      BR 1$
14462
14463      ;SUBROUTINE TO CONVERT A BINARY # TO OCTAL (ASCII) AN TYPE IT
14464      ;ENTER WITH R2 = # OF OCTAL DIGITS TO TYPE
14465      ;          R3 = BINARY #
14466
14467 065714 112737 000005 066014  POCT:      MOVB #5,OCNT           ;SET THE ITERATION COUNT
14468 065722 005402      NEG R2
14469 065724 062702 000006      ADD #6,R2              ;SUBTRACT # OF DIGITS TO TYPE FROM MAX ALLOWED
14470 065730 110237 066015      MOVB R2,OMODE          ;SAVE IT FOR USE
14471 065734 005004      CLR R4
14472 065736 006103      1$:      ROL R3                 ;ROTATE MSB INTO 'C'
14473 065740 000404      BR 3$
14474 065742 006103      2$:      ROL R3                 ;FORM THIS DIGIT
14475 065744 006103      ROL R3
14476 065746 006103      ROL R3
14477 065750 010304      MOV R3,R4
14478 065752 006104      3$:      ROL R4                 ;GET LSB OF THIS DIGIT
14479 065754 105337 066015      DECB OMODE             ;TYPE THIS DIGIT
14480 065760 100010      BPL 7$                 ;BRANCH IF NO
14481 065762 042704 177770      BIC #177770,R4         ;GET RID OF JUNK
14482 065766 052704 000060      BIS #'0,R4             ;MAKE THIS DIGIT ASCII
14483 065772 110437 066016      MOVB R4,TDIG           ;SAVE FOR TYPING
14484 065776 104400 066016      TYPE ,TDIG             ;TYPE THIS DIGIT
14485 066002 105337 066014      7$:      DECB OCNT              ;COUNT BY 1
14486 066006 002401      BLT 6$                 ;BRANCH IF DONE
```

```
14487 066010 000754          BR 2$          ;BRANCH IF MORE TO DO
14488 066012 000207          6$:          RTS PC
14489
14490 066014      000          OCNT:         .BYTE 0
14491 066015      000          OMODE:        .BYTE 0
14492 066016      000          TDIG:         .BYTE 0
14493 066017      000          .BYTE 0
14494
14495          :SUBROUTINE TO CHANGE A BINARY # TO ASCII AND TYPE IT
14496          :ENTER WITH R2 = # OF BINARY DIGITS TO TYPE
14497          :R3 = BINARY #
14498
14499 066020 112737 000017 066014 PBIN:          MOVB #17,OCNT          ;SET THE ITERATION COUNT
14500 066026 005402          NEG R2
14501 066030 062702 000020          ADD #20,R2          ;SUBTRACT # OF DIGITS TO TYPE FROM MAX ALLOWED
14502 066034 110237 066015          MOVB R2,OMODE        ;SAVE IT FOR USE
14503 066040 005004          CLR R4          ;CLEAR THE OUTPUT WORD
14504 066042 006103          1$:          ROL R3
14505 066044 012704 000000          MOV #0,R4
14506 066050 006104          ROL R4          ;GET BINARY DIGIT
14507 066052 105337 066015          DECB OMODE          ;TYPE THIS DIGIT?
14508 066056 100010          BPL 7$          ;BRANCH IF NO
14509 066060 042704 177776          BIC #177776,R4      ;GET RID OF JUNK
14510 066064 052704 000060          BIS #'0,R4          ;MAKE THIS BIT ASCII
14511 066070 010437 066016          MOV R4,TDIG         ;SAVE FOR TYPING
14512 066074 104400 066016          TYPE ,TDIG         ;TYPE THIS DIGIT
14513 066100 105337 066014          7$:          DECB OCNT          ;COUNT BY 1
14514 066104 002401          BLT 6$          ;BRANCH IF DONE
14515 066106 000755          BR 1$          ;BRANCH IF MORE TO DO
14516 066110 000207          6$:          RTS PC
14517
14518          :SUBROUTINE TO CONVERT A BINARY # TO DECIMAL (ASCII) AND TYPE DECIMAL DIGITS
14519          :ENTER WITH R3 = BINARY #
14520          :R2 = # OF DECIMAL DIGITS TO TYPE
14521
14522          PDEC:          MOV R1,-(SP)          ;SAVE R1
14523 066112 010146          MOV R0,-(SP)          ;SAVE R0
14524 066114 010046          MOV #5,R0
14525 066116 012700 000005          SUB R2,R0          ;R0 CONTAINS # OF DIGITS TO SKIP BEFORE PRINTING
14526 066122 160200          CLR R4          ;ZERO CONSTANTS TABLE INDEX
14527 066124 005004          MOV #DBLK,R5        ;SETUP THE OUTPUT POINTER
14528 066126 012705 066232          CLR R2          ;CLEAR THE BCD #
14529 066132 005002          2$:          MOV DTBL(R4),R1     ;GET THE CONSTANT
14530 066134 016401 066222          3$:          SUB R1,R3          ;FORM THIS BCD DIGIT
14531 066140 160103          BLO 4$          ;BRANCH IF DONE
14532 066142 103402          INC R2          ;INCREASE THE BCD DIGIT BY 1
14533 066144 005202          BR 3$
14534 066146 000774          4$:          ADD R1,R3          ;ADD BACK THE CONSTANT
14535 066150 060103          6$:          BIS #'0,R2        ;MAKE THE BCD DIGIT ASCII
14536 066152 052702 000060          TST R0          ;PRINT THIS DIGIT?
14537 066156 005700          BEQ 61$          ;BRANCH IF YES
14538 066160 001402          DEC R0          ;DECREMENT SKIP COUNT
14539 066162 005300          BR 62$
14540 066164 000401
```

```

14541 066166 110225      61$:  MOVB R2,(R5)+      ;PUT THIS CHAR IN THE OUTPUT BUFFER
14542 066170 005724      62$:  TST (R4)+          ;JUST INCREMENTING
14543 066172 020427 000010  CMP R4,#10          ;CHECK THE TABLE INDEX
14544 066176 002755      BLT 2$              ;GO DO THE NEXT DIGIT
14545 066200 003002      BGT 8$              ;GO TO EXIT
14546 066202 010302      MOV R3,R2           ;GET LSD
14547 066204 000762      BR 6$              ;GO CHANGE TO ASCII
14548 066206 105015      8$:  CLRB (R5)         ;SET THE TERMINATOR
14549 066210 104400 066232  TYPE ,DBLK         ;NOW TYPE THE #
14550 066214 012600      MOV (SP)+,R0       ;RESTORE R0
14551 066216 012601      MOV (SP)+,R1       ;RESTORE R1
14552 066220 000207      RTS PC             ;EXIT
14553 066222 023420      DTBL: 10000.
14554 066224 001750      1000.
14555 066226 000144      100.
14556 066230 000012      10.
14557 066232 000004      DBLK: .BLKW 4
14558
14559
14560
14561
14562
14563
14564
14565
14566
14567
14568
14569
14570
14571
14572
14573
14574
14575
14576
14577
14578
14579
14580
14581
14582
14583
14584
14585
14586 066242 005737 035744  PRNIB: TST EZDF      ;IS TEST CONDITION A 'DIVIDE BY ZERO'
14587 066246 001401      BEQ 1$             ;YES - EXIT WITHOUT DISPLAYING ANY BUFFER STRINGS
14588 066250 000207      RTS PC             ;IS SPECIAL HANDLING BIT 2 SET?
14589 066252 032737 000004 002140 1$:  BIT #4,SPHAND
14590 066260 001401      BEQ 2$             ;YES - EXIT WITHOUT DISPLAYING ANY BUFFER STRINGS.
14591 066262 000207      RTS PC             ;ARE ANY STRINGS TO BE DISPLAYED?
14592 066264 005737 002466 2$:  TST PZCODE
14593 066270 001001      BNE 3$
14594 066272 000207      RTS PC             ;NO - EXIT WITHOUT DISPLAY.

```

.....
 : SUBROUTINE TO DISPLAY DECIMAL STRING SOURCES AND RESULTS IN
 : DECIMAL FORM. STRINGS TO BE DISPLAYED ARE IDENTIFIED BY THE
 : CONTENTS OF PZCODE AS FOLLOWS:
 :

| | | | |
|-----------|---------------|---------------|----------------------|
| BIT 0 = 1 | DISPLAY ZONED | SOURCE STRING | |
| BIT 1 = 1 | | SRC1 | .. |
| BIT 2 = 1 | | SRC2 | .. |
| BIT 3 = 1 | | DEST | .. (DESC IN ER4,ER5) |
| BIT 4 = 1 | | DEST | .. (DESC IN ER2,ER3) |
| BIT 5 = 1 | | DEST | .. (DESC IN ER0,ER1) |
| BIT 8 = 1 | .. PACKED | SOURCE | .. |
| BIT 9 = 1 | | SRC1 | .. |
| BIT10 = 1 | | SRC2 | .. |
| BIT11 = 1 | | DEST | .. (DESC IN ER4,ER5) |
| BIT12 = 1 | | DEST | .. (DESC IN ER2,ER3) |
| BIT13 = 1 | | DEST | .. (DESC IN ER0,ER1) |

NOTE: ALL SOURCE STRINGS MUST BE STORED IN THE INPUT
 SOURCE BUFFER DESCRIBED BY THE DESCRIPTOR(S) AT
 INSR1 AND INSR2.

IF THE DIVP BY 0 FLAG IS SET (EZDF) OR BIT 2 OF THE
 SPECIAL HANDLING CODE IS SET (SPHAND) THEN THIS SUBROUTINE
 RETURNS WITHOUT DISPLAYING ANY STRINGS.

| | | | | | | | |
|-------|--------|--------|--------|--------|-------|-------------------|-----------------------------------|
| 14595 | 066274 | 005037 | 026056 | | 3\$: | CLR EPAK | |
| 14596 | 066300 | 032737 | 000001 | 002466 | | BIT #1,PZCODE | ;PRINT A ZONED SRC STRING? |
| 14597 | 066306 | 001402 | | | | BEQ 4\$ | |
| 14598 | 066310 | 004737 | 066462 | | | JSR PC,SN | ;YES |
| 14599 | 066314 | 032737 | 000002 | 002466 | 4\$: | BIT #2,PZCODE | ;PRINT A ZONED SRC1 STRING? |
| 14600 | 066322 | 001402 | | | | BEQ 5\$ | |
| 14601 | 066324 | 004737 | 066512 | | | JSR PC,S1N | ;YES |
| 14602 | 066330 | 032737 | 000004 | 002466 | 5\$: | BIT #4,PZCODE | ;PRINT A ZONED SRC2 STRING? |
| 14603 | 066336 | 001402 | | | | BEQ 6\$ | |
| 14604 | 066340 | 004737 | 066602 | | | JSR PC,S2N | ;YES |
| 14605 | 066344 | 012737 | 177777 | 026056 | 6\$: | MOV #177777,EPAK | |
| 14606 | 066352 | 032737 | 000400 | 002466 | | BIT #400,PZCODE | ;PRINT A PACKED SOURCE STRING? |
| 14607 | 066360 | 001402 | | | | BEQ 7\$ | |
| 14608 | 066362 | 004737 | 066462 | | | JSR PC,SN | ;YES |
| 14609 | 066366 | 032737 | 001000 | 002466 | 7\$: | BIT #1000,PZCODE | ;PRINT A PACKED SRC1 STRING? |
| 14610 | 066374 | 001402 | | | | BEQ 10\$ | |
| 14611 | 066376 | 004737 | 066512 | | | JSR PC,S1N | ;YES |
| 14612 | 066402 | 032737 | 002000 | 002466 | 10\$: | BIT #2000,PZCODE | ;PRINT A PACKED SRC2 STRING? |
| 14613 | 066410 | 001402 | | | | BEQ 11\$ | |
| 14614 | 066412 | 004737 | 066602 | | | JSR PC,S2N | ;YES |
| 14615 | 066416 | 032737 | 000070 | 002466 | 11\$: | BIT #70,PZCODE | ;PRINT A ZONED DEST. STRING? |
| 14616 | 066424 | 001404 | | | | BEQ 12\$ | |
| 14617 | 066426 | 005037 | 026056 | | | CLR EPAK | |
| 14618 | 066432 | 004737 | 066672 | | | JSR PC,DN | ;YES |
| 14619 | 066436 | 032737 | 034000 | 002466 | 12\$: | BIT #34000,PZCODE | ;PRINT A PACKED DEST. STRING? |
| 14620 | 066444 | 001405 | | | | BEQ 13\$ | |
| 14621 | 066446 | 012737 | 177777 | 026056 | | MOV #177777,EPAK | |
| 14622 | 066454 | 004737 | 066672 | | | JSR PC,DN | ;YES |
| 14623 | 066460 | 000207 | | | 13\$: | RTS PC | ;EXIT DECIMAL DISPLAY SUBROUTINE. |
| 14624 | | | | | | | |
| 14625 | | | | | | | |
| 14626 | 066462 | | | | SN: | PRINTB #FORM22 | ;PRINT 'SRC' |
| (6) | 066462 | 012746 | 014023 | | | MOV #FORM22,-(SP) | |
| (3) | 066466 | 010600 | | | | MOV SP,R0 | |
| (4) | 066470 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14627 | 066474 | 013701 | 002474 | | | MOV INSRC1,R1 | ;LOAD R1 WITH STRING LEN |
| 14628 | 066500 | 013700 | 002476 | | | MOV INSRC1+2,R0 | ;LOAD R0 WITH STRING ADD |
| 14629 | 066504 | 004737 | 067060 | | | JSR PC,DECPRT | ;PRINT DECIMAL DIGIT STRING |
| 14630 | 066510 | 000207 | | | | RTS PC | |
| 14631 | 066512 | 013701 | 047464 | | S1N: | MOV TINST,R1 | |
| 14632 | 066516 | 042701 | 177700 | | | BIC #177700,R1 | |
| 14633 | 066522 | 020127 | 000052 | | | CMP R1,#52 | ;IS INST = CMPN? |
| 14634 | 066526 | 001403 | | | | BEQ 1\$ | ;BRANCH IF YES |
| 14635 | 066530 | 020127 | 000072 | | | CMP R1,#72 | ;IS INST = CMPP? |
| 14636 | 066534 | 001006 | | | | BNE 2\$ | ;BRANCH IF NO |
| 14637 | 066536 | | | | 1\$: | PRINTB #FORM24 | ;PRINT 'SRC2' |
| (6) | 066536 | 012746 | 014056 | | | MOV #FORM24,-(SP) | |
| (3) | 066542 | 010600 | | | | MOV SP,R0 | |
| (4) | 066544 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14638 | 066550 | 000405 | | | | BR 3\$ | |
| 14639 | 066552 | | | | 2\$: | PRINTB #FORM23 | ;PRINT 'SRC1' |
| (6) | 066552 | 012746 | 014040 | | | MOV #FORM23,-(SP) | |
| (3) | 066556 | 010600 | | | | MOV SP,R0 | |
| (4) | 066560 | 004737 | 065304 | | | JSR PC,FPRINT | |

| | | | | | | | |
|-------|--------|--------|--------|--------|-------|-------------------|------------------------------|
| 14640 | 066564 | 013701 | 002474 | | 3\$: | MOV INSR1,R1 | :LOAD R1 WITH STRING LEN |
| 14641 | 066570 | 013700 | 002476 | | | MOV INSR1+2,R0 | :LOAD R0 WITH STRING ADD |
| 14642 | 066574 | 004737 | 067060 | | | JSR PC,DECPRT | :PRINT DECIMAL DIGIT STRING |
| 14643 | 066600 | 000207 | | | | RTS PC | |
| 14644 | 066602 | 013701 | 047464 | | 52N: | MOV TINST,R1 | |
| 14645 | 066606 | 042701 | 177700 | | | BIC #177700,R1 | |
| 14646 | 066612 | 020127 | 000052 | | | CMP R1,#52 | :IS INST = CMPN? |
| 14647 | 066616 | 001403 | | | | BEQ 1\$ | :BRANCH IF YES |
| 14648 | 066620 | 020127 | 000072 | | | CMP R1,#72 | :IS INST = CMPP? |
| 14649 | 066624 | 001006 | | | | BNE 2\$ | :BRANCH IF NO |
| 14650 | 066626 | | | | 1\$: | PRINTB #FORM23 | :PRINT 'SRC1' |
| (6) | 066626 | 012746 | 014040 | | | MOV #FORM23,-(SP) | |
| (3) | 066632 | 010600 | | | | MOV SP,R0 | |
| (4) | 066634 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14651 | 066640 | 000405 | | | | BR 3\$ | |
| 14652 | 066642 | | | | 2\$: | PRINTB #FORM24 | :PRINT 'SRC2' |
| (6) | 066642 | 012746 | 014056 | | | MOV #FORM24,-(SP) | |
| (3) | 066646 | 010600 | | | | MOV SP,R0 | |
| (4) | 066650 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14653 | 066654 | 013701 | 002500 | | 3\$: | MOV INSR2,R1 | :LOAD R1 WITH STRING LENGTH |
| 14654 | 066660 | 013700 | 002502 | | | MOV INSR2+2,R0 | :LOAD R0 WITH STRING ADD |
| 14655 | 066664 | 004737 | 067060 | | | JSR PC,DECPRT | :PRINT DECIMAL DIGIT STRING |
| 14656 | 066670 | 000207 | | | | RTS PC | |
| 14657 | 066672 | | | | DN: | PRINTB #FORM25 | :PRINT 'EM RESULT' |
| (6) | 066672 | 012746 | 014074 | | | MOV #FORM25,-(SP) | |
| (3) | 066676 | 010600 | | | | MOV SP,R0 | |
| (4) | 066700 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14658 | 066704 | 032737 | 020040 | 002466 | | BIT #20040,PZCODE | :LOAD R1 WITH STRING LEN |
| 14659 | 066712 | 001405 | | | | BEQ 1\$ | :LOAD R0 WITH STRING ADDRESS |
| 14660 | 066714 | 013701 | 003664 | | | MOV ER0,R1 | |
| 14661 | 066720 | 013700 | 003666 | | | MOV ER1,R0 | |
| 14662 | 066724 | 000415 | | | | BR 4\$ | |
| 14663 | 066726 | 032737 | 010020 | 002466 | 1\$: | BIT #10020,PZCODE | |
| 14664 | 066734 | 001405 | | | | BEQ 2\$ | |
| 14665 | 066736 | 013701 | 003670 | | | MOV ER2,R1 | |
| 14666 | 066742 | 013700 | 003672 | | | MOV ER3,R0 | |
| 14667 | 066746 | 000404 | | | | BR 4\$ | |
| 14668 | 066750 | 013701 | 003674 | | 2\$: | MOV ER4,R1 | |
| 14669 | 066754 | 013700 | 003676 | | | MOV ER5,R0 | |
| 14670 | 066760 | 004737 | 067060 | | 4\$: | JSR PC,DECPRT | :PRINT DECIMAL DIGIT STRING |
| 14671 | 066764 | | | | | PRINTB #FORM26 | :PRINT 'ACT RESULT' |
| (6) | 066764 | 012746 | 014117 | | | MOV #FORM26,-(SP) | |
| (3) | 066770 | 010600 | | | | MOV SP,R0 | |
| (4) | 066772 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14672 | 066776 | 032737 | 020040 | 002466 | | BIT #20040,PZCODE | :LOAD R1 WITH STRING LEN |
| 14673 | 067004 | 001405 | | | | BEQ 11\$ | :LOAD R0 WITH STRING ADDRESS |
| 14674 | 067006 | 013701 | 002230 | | | MOV TTR0,R1 | |
| 14675 | 067012 | 013700 | 002232 | | | MOV TTR1,R0 | |
| 14676 | 067016 | 000415 | | | | BR 44\$ | |
| 14677 | 067020 | 032737 | 010020 | 002466 | 11\$: | BIT #10020,PZCODE | |
| 14678 | 067026 | 001405 | | | | BEQ 22\$ | |
| 14679 | 067030 | 013701 | 002234 | | | MOV TTR2,R1 | |
| 14680 | 067034 | 013700 | 002236 | | | MOV TTR3,R0 | |
| 14681 | 067040 | 000404 | | | | BR 44\$ | |

| | | | | | | | |
|-------|--------|--------|--------|--------|---------|--------------------|---------------------------------------|
| 14682 | 067042 | 013701 | 002240 | | 22\$: | MOV TTR4,R1 | |
| 14683 | 067046 | 013700 | 002242 | | | MOV TTR5,R0 | |
| 14684 | 067052 | 004737 | 067060 | | 44\$: | JSR PC,DECPRT | :PRINT DECIMAL DIGIT STRING |
| 14685 | 067056 | 000207 | | | | RTS PC | |
| 14686 | | | | | | | |
| 14687 | | | | | | | |
| 14688 | | | | | | | |
| 14689 | | | | | | | |
| 14690 | | | | | | | |
| 14691 | | | | | | | |
| 14692 | | | | | | | |
| 14693 | | | | | | | |
| 14694 | | | | | | | |
| 14695 | | | | | | | |
| 14696 | | | | | | | |
| 14697 | 067060 | 012737 | 177777 | 002300 | DECPRT: | MOV #177777,PRTSGN | :SET PRINTING IN PROGRESS FLAG |
| 14698 | 067066 | 010003 | | | | MOV R0,R3 | :SAVE R0 IN R3 |
| 14699 | 067070 | 005037 | 024646 | | | CLR EODD | |
| 14700 | 067074 | 032701 | 000001 | | | BIT #1,R1 | :IS STRING ODD IN LENGTH |
| 14701 | 067100 | 001403 | | | | BEQ 1\$ | |
| 14702 | 067102 | 012737 | 177777 | 024646 | | MOV #177777,EODD | :SET ODD INDICATOR |
| 14703 | 067110 | 010137 | 024664 | | 1\$: | MOV R1,ELSD | |
| 14704 | 067114 | 110137 | 002100 | | | MOVB R1,NBLKS | :DETERMINE # OF BLANK DIGITS TO PRINT |
| 14705 | 067120 | 012702 | 000037 | | | MOV #37,R2 | |
| 14706 | 067124 | 163702 | 002100 | | | SUB NBLKS,R2 | |
| 14707 | 067130 | 001002 | | | | BNE 11\$ | |
| 14708 | 067132 | 012702 | 000001 | | | MOV #1,R2 | |
| 14709 | 067136 | | | | 11\$: | PRINTB #FORM27 | :PRINT THE BLANKS |
| (6) | 067136 | 012746 | 014142 | | | MOV #FORM27,-(SP) | |
| (3) | 067142 | 010600 | | | | MOV SP,R0 | |
| (4) | 067144 | 004737 | 065304 | | | JSR PC,FPRINT | |
| 14710 | 067150 | 005302 | | | | DEC R2 | |
| 14711 | 067152 | 001371 | | | | BNE 11\$ | |
| 14712 | 067154 | 105701 | | | 3\$: | TSTB R1 | :STRING LENGTH = 0? |
| 14713 | 067156 | 001017 | | | | BNE 4\$ | :BRANCH IF NO |
| 14714 | 067160 | 005737 | 026056 | | | TST EPAK | :STRING PACKED? |
| 14715 | 067164 | 001407 | | | | BEQ 5\$ | :BRANCH IF NO |
| 14716 | 067166 | 005201 | | | | INC R1 | :YES - SET LEN = 1 |
| 14717 | 067170 | 005237 | 024664 | | | INC ELSD | |
| 14718 | 067174 | 012737 | 177777 | 024646 | | MOV #177777,EODD | |
| 14719 | 067202 | 000405 | | | | BR 4\$ | |
| 14720 | 067204 | 005002 | | | 5\$: | CLR R2 | :ZONED - ZERO LENGTH |
| 14721 | 067206 | 004537 | 067402 | | | JSR R5,CONVN | :PRINT 0 |
| 14722 | 067212 | 067436 | | | | DIGTBL | |
| 14723 | 067214 | 000413 | | | | BR 12\$ | :EXIT |
| 14724 | 067216 | 010300 | | | 4\$: | MOV R3,R0 | :RESTORE R0 |
| 14725 | 067220 | 105001 | | | | CLRB R1 | |
| 14726 | 067222 | 004737 | 021126 | | 7\$: | JSR PC,ESNK | :GET NEXT DIGIT |
| 14727 | 067226 | 004537 | 067402 | | | JSR R5,CONVN | :CONVERT NIBBLE & PRINT HEX DIGIT |
| 14728 | 067232 | 067436 | | | | DIGTBL | |
| 14729 | 067234 | 105201 | | | | INCB R1 | |
| 14730 | 067236 | 120137 | 024664 | | | CMPB R1,ELSD | |
| 14731 | 067242 | 001367 | | | | BNE 7\$ | |
| 14732 | 067244 | 004737 | 021126 | | 12\$: | JSR PC,ESNK | :CALL ROUTINE TO FIND SIGN |

```

: SUBROUTINE TO PRINT A DECIMAL STRING OF DIGITS; MSD FIRST ....
: LEAST SIGNIFICANT DIGIT, SIGN.
: INPUT: EPAK=0 FOR ZONED STRING;177777 FOR PACKED
:        R0=STRING ADR
:        R1=STRING LEN
:
: NOTE: ROUTINE PRINTS '0 +' FOR ZONED STRINGS OF
: ZERO LENGTH (EXCEPT SEPARATE TYPE).

```

```

14733                                     ;SIGN RETURNED IN ERSNEG (0=+ ,/=0=-)
14734                                     ;SIGN BYTE RETURNED IN 'SGNBYT'
14735 067250 005737 024616               TST ERSNEG
14736 067254 001006                       BNE 2$
14737 067256                               PRINTB #FORM34
   (6) 067256 012746 014226             MOV #FORM34,-(SP)
   (3) 067262 010600                       MOV SP,R0
   (4) 067264 004737 065304             JSR PC,FPRINT
14738 067270 000405                       BR 33$
14739 067272                               PRINTB #FORM35
   (6) 067272 012746 014233             MOV #FORM35,-(SP)
   (3) 067276 010600                       MOV SP,R0
   (4) 067300 004737 065304             JSR PC,FPRINT
14740 067304                               PRINTB #FORM32
   (6) 067304 012746 014202             MOV #FORM32,-(SP)
   (3) 067310 010600                       MOV SP,R0
   (4) 067312 004737 065304             JSR PC,FPRINT
14741 067316 013702 024722             MOV SGNBYT,R2
14742 067322 006202                       ASR R2
14743 067324 006202                       ASR R2
14744 067326 006202                       ASR R2
14745 067330 006202                       ASR R2
14746 067332 042702 177760             BIC #177760,R2
14747 067336 004537 067402             JSR R5,CONVN
14748 067342 067436                       DIGTBL
14749 067344 113702 024722             MOV SGNBYT,R2
14750 067350 042702 177760             BIC #177760,R2
14751 067354 004537 067402             JSR R5,CONVN
14752 067360 067436                       DIGTBL
14753 067362                               PRINTB #FORM33
   (6) 067362 012746 014222             MOV #FORM33,-(SP)
   (3) 067366 010600                       MOV SP,R0
   (4) 067370 004737 065304             JSR PC,FPRINT
14754 067374 005037 002300             CLR PRISGN
14755 067400 000207                       RTS PC
14756
14757
14758
14759                                     ; SUBROUTINE TO CONVERT NIBBLE (IN R2) TO A PRINTABLE CHARACTER
14760                                     ; AND PRINT CHARACTER.
14761                                     ; INPUT PARAMETER FOLLOWS CALL - CONVERSION TABLE ADDRESS
14762
14763 CONVN:
14764 067402                               MOV R0,-(SP)
14765 067404 010046                       MOV (R5)+,R0
14766 067406 060200                       ADD R2,R0
14767 067410 111037 067434             MOV (R0),ANIB+2
14768 067414                               PRINTB #ANIB
   (6) 067414 012746 067432             MOV #ANIB,-(SP)
   (3) 067420 010600                       MOV SP,R0
   (4) 067422 004737 065304             JSR PC,FPRINT
14769 067426 012600                       MOV (SP)+,R0
14770 067430 000205                       RTS R5
14771

```

| | | | | | | |
|-------|--------|-----|---------|-------|-----|-------------|
| 14772 | 067432 | 045 | ANIB: | .BYTE | 045 | ;% |
| 14773 | 067433 | 101 | | .BYTE | 101 | :A |
| 14774 | 067434 | 000 | | .BYTE | 000 | :PRINT CHAR |
| 14775 | 067435 | 000 | | .BYTE | 000 | :ZERO BYTE |
| 14776 | | | | | | |
| 14777 | | | | | | |
| 14778 | 067436 | 060 | DIGTBL: | .BYTE | 060 | :0 |
| 14779 | 067437 | 061 | | .BYTE | 061 | :1 |
| 14780 | 067440 | 062 | | .BYTE | 062 | :2 |
| 14781 | 067441 | 063 | | .BYTE | 063 | :3 |
| 14782 | 067442 | 064 | | .BYTE | 064 | :4 |
| 14783 | 067443 | 065 | | .BYTE | 065 | :5 |
| 14784 | 067444 | 066 | | .BYTE | 066 | :6 |
| 14785 | 067445 | 067 | | .BYTE | 067 | :7 |
| 14786 | 067446 | 070 | | .BYTE | 070 | :8 |
| 14787 | 067447 | 071 | | .BYTE | 071 | :9 |
| 14788 | 067450 | 101 | | .BYTE | 101 | :A |
| 14789 | 067451 | 102 | | .BYTE | 102 | :B |
| 14790 | 067452 | 103 | | .BYTE | 103 | :C |
| 14791 | 067453 | 104 | | .BYTE | 104 | :D |
| 14792 | 067454 | 105 | | .BYTE | 105 | :E |
| 14793 | 067455 | 106 | | .BYTE | 106 | :F |

```
14794
14795      :SUBROUTINE TO SEARCH FOR A MATCH BETWEEN ENTERED INST
14796      :AND TABLED ASCII LIST OF CIS INSTRUCTIONS.
```

```
14797
14798 067456 012701 004444 SFCI:  MOV #ASZINS,R1
14799 067462 005711 1$:      TST (R1)                ;REACHED END OF TABLED ASCII LIST?
14800 067464 001434      BEQ NOMTCH                ;BRANCH IF YES
14801 067466 021137 002456      CMP (R1),ACINST          ;DO 1ST TWO CHARS MATCH TABLED INST?
14802 067472 001403      BEQ 2$                  ;BRANCH IF YES
14803 067474 062701 000010 11$:   ADD #10,R1              ;UPDATE TO NEXT TABLED INST
14804 067500 000770      BR 1$                  ;RETURN TO CONTINUE SEARCH
14805 067502 026137 000002 002460 2$:  CMP 2(R1),ACINST+2      ;DO 2ND GROUP OF 2 CHARS MATCH
14806 067510 001401      BEQ 3$                  ;BRANCH IF YES
14807 067512 000770      BR 11$                 ;
14808 067514 026137 000004 002462 3$:  CMP 4(R1),ACINST+4      ;DO 3RD GROUP OF 2 CHARS MATCH
14809 067522 001364      BNE 11$                 ;BRANCH IF NO
14810 067524 005737 001760      TST RANDOM              ;RANDOM EXERCISE MODE?
14811 067530 001405      BEQ 4$                  ;BRANCH IF NO
14812 067532 016100 000006      MOV 6(R1),R0
14813 067536 011037 072022      MOV (R0),IDUM           ;LOAD OCTAL CODING FOR CIS INST INTO DUMMY INPUT TABLE
14814 067542 000403      BR MTCH
14815 067544 016137 000006 072076 4$:  MOV 6(R1),INPTBL        ;MATCH FOUND - RETURN TO CALL + 4
14816                                     ;SAVE DESIRED INST INPUT TABLE ADDRESS
14817                                     ;IN INPTBL.
14818 067552 062716 000002      MTCH:  ADD #2,(SP)
14819 067556 000207      NOMTCH: RTS PC          ;NO MATCH - RETURN TO CALL + 2
```

```
14820
14821      :
14822      :SUBROUTINE TO ACCEPT ASCII CHARS (6MAX,LESS+CR) FROM TTY.
14823      :STORES ASCII CHARS 2 PER WORD IN ACINST,ACINST+2, AND ACINST+4.
14824
14825 067560 005037 002456 ACASZ: CLR ACINST
```

| | | | | | | |
|-------|--------|--------|--------|--------|------------------|---------------------------------------|
| 14826 | 067564 | 005037 | 002460 | | CLR ACINST+2 | |
| 14827 | 067570 | 005037 | 002462 | | CLR ACINST+4 | ;CLEAR OUT STORAGE AREA |
| 14828 | 067574 | 012701 | 002456 | | MOV #ACINST,R1 | ;SETUP REG POINTER TO STORAGE AREA |
| 14829 | 067600 | 105777 | 112064 | | TSTB @TKS | ;WAIT FOR A CHAR |
| 14830 | 067604 | 100375 | | | BPL 1\$ | |
| 14831 | 067606 | 117737 | 112100 | 064672 | MOVB @TKB,RCHAR | ;READ AND SAVE CHAR |
| 14832 | 067614 | 042737 | 177600 | 064672 | BIC #^C177,RCHAR | ;GET RID OF JUNK IF ANY |
| 14833 | 067622 | 123727 | 064672 | 000015 | CMPB RCHAR,#15 | ;IS CHAR A CR? |
| 14834 | 067630 | 001412 | | | BEQ 2\$ | ;BRANCH IF YES |
| 14835 | 067632 | 113721 | 064672 | | MOVB RCHAR,(R1)+ | ;SAVE CHAR |
| 14836 | 067636 | 004737 | 064654 | | JSR PC,ECHAR | ;ECHO 6TH CHAR |
| 14837 | 067642 | 022701 | 002464 | | CMP #ACINST+6,R1 | ;6 CHARS ENTERED? |
| 14838 | 067646 | 001354 | | | BNE 1\$ | ;BRANCH IF NO TO LISTEN FOR NEXT CHAR |
| 14839 | 067650 | 112737 | 000015 | 064672 | MOVB #15,RCHAR | ;ECHO A CR |
| 14840 | 067656 | 004737 | 064654 | | JSR PC,ECHAR | ;ECHO CR |
| 14841 | 067662 | 000207 | | | RTS PC | |
| 14842 | | | | | | |
| 14843 | | | | | | |
| 14844 | | | | | | |
| 14845 | | | | | | |
| 14846 | 067664 | | | | | |
| 14847 | 067664 | 013701 | 002272 | | | |
| 14848 | 067670 | 006301 | | | | |
| 14849 | 067672 | 062701 | 004012 | | | |
| 14850 | 067676 | 113737 | 003720 | 002035 | | |
| 14851 | 067704 | 153711 | 003720 | | | |
| 14852 | 067710 | 005137 | 002034 | | | |
| 14853 | 067714 | 042737 | 170377 | 002034 | | |
| 14854 | 067722 | 053711 | 002034 | | | |
| 14855 | 067726 | 000207 | | | | |


```

;SUBROUTINE TO RECORD WHICH CONDITION CODE STATES ARE EXERCISED
;FOR EACH INSTRUCTION.
RECCC:
MOV OCTIC,R1          ;FORM POINTER INTO TABLE OF COND. CODE USAGE
ASL R1
ADD #CCREC,R1
MOVB ECCR,ZCCR+1
BISB ECCR,(R1)        ;LOG CC '1' STATES EXERCISED
COM ZCCR
BIC #170377,ZCCR
BIS ZCCR,(R1)        ;LOG CC '0' STATES EXERCISED
RTS PC
    
```

```
14857          :POINTERS TO CIS INSTRUCTION ERROR MESSAGE HEADER ROUTINE
14858          :
14859 067730      :INEM:
14860 067730      .WORD 0
14861 067732      .WORD YMOVC
14862 067734      .WORD YMOVRC
14863 067736      .WORD YMOVTC
14864 067740      .WORD YLOCC
14865 067742      .WORD YSKPC
14866 067744      .WORD YSCANC
14867 067746      .WORD YSPANC
14868 067750      .WORD YCMPC
14869 067752      .WORD YMATCHC
14870 067754      .WORD YADDN
14871 067756      .WORD YSUBN
14872 067760      .WORD YCMPN
14873 067762      .WORD YCVTNL
14874 067764      .WORD YCVTPN
14875 067766      .WORD YCVTNP
14876 067770      .WORD YASHN
14877 067772      .WORD YCVTLN
14878 067774      .WORD YADDP
14879 067776      .WORD YSUBP
14880 070000      .WORD YCMPP
14881 070002      .WORD YCVTPN
14882 070004      .WORD YMULP
14883 070006      .WORD YDIVP
14884 070010      .WORD YASHP
14885 070012      .WORD YCVTLP
14886 070014      .WORD YL2D
14887 070016      .WORD YL3D
14888
```

| Address | Instruction | PC | SP | OP | Label | Comment |
|---------|-------------|--------|--------|----|---------------|-------------------------|
| 14890 | | | | | .SBTTL | ERROR MESSAGE HEADERS |
| 14891 | | | | | :ERROR | MESSAGE HEADERS |
| 14892 | | | | | : | |
| 14893 | 070020 | | | | YMOVVC: | |
| 14894 | 070020 | | | | PRINTB | #AMOVVC |
| (6) | 070020 | 012746 | 007434 | | MOV | #AMOVVC,-(SP) |
| (3) | 070024 | 010600 | | | MOV | SP,R0 |
| (4) | 070026 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14895 | 070032 | 004737 | 071554 | | JSR | PC,PRNTIQ ;PRINT TEST # |
| 14896 | 070036 | 000405 | | | BR 1\$ | ;CNTL-T RETURN |
| 14897 | 070040 | | | | PRINTB | #FORM1 ;NORMAL RETURN |
| (6) | 070040 | 012746 | 010002 | | MOV | #FORM1,-(SP) |
| (3) | 070044 | 010600 | | | MOV | SP,R0 |
| (4) | 070046 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14898 | 070052 | 000207 | | | RTS | PC |
| 14899 | 070054 | | | | 1\$: | |
| 14900 | 070054 | | | | YMOVRC: | |
| (6) | 070054 | 012746 | 007444 | | PRINTB | #AMOVRC |
| (3) | 070060 | 010600 | | | MOV | #AMOVRC,-(SP) |
| (4) | 070062 | 004737 | 065304 | | MOV | SP,R0 |
| 14901 | 070066 | 004737 | 071554 | | JSR PC,FPRINT | |
| 14902 | 070072 | 000405 | | | JSR | PC,PRNTIQ ;PRINT TEST # |
| 14903 | 070074 | | | | BR 1\$ | |
| (6) | 070074 | 012746 | 010002 | | PRINTB | #FORM1 |
| (3) | 070100 | 010600 | | | MOV | #FORM1,-(SP) |
| (4) | 070102 | 004737 | 065304 | | MOV | SP,R0 |
| 14904 | 070106 | 000207 | | | JSR PC,FPRINT | |
| 14905 | 070110 | | | | RTS | PC |
| 14906 | 070110 | | | | 1\$: | |
| (6) | 070110 | 012746 | 007455 | | YMOVTC: | |
| (3) | 070114 | 010600 | | | PRINTB | #AMOVTC |
| (4) | 070116 | 004737 | 065304 | | MOV | #AMOVTC,-(SP) |
| 14907 | 070122 | 004737 | 071554 | | MOV | SP,R0 |
| 14908 | 070126 | 000405 | | | JSR PC,FPRINT | |
| 14909 | 070130 | | | | JSR | PC,PRNTIQ ;PRINT TEST # |
| (6) | 070130 | 012746 | 010062 | | BR 1\$ | |
| (3) | 070134 | 010600 | | | PRINTB | #FORM2 |
| (4) | 070136 | 004737 | 065304 | | MOV | #FORM2,-(SP) |
| 14910 | 070142 | 000207 | | | MOV | SP,R0 |
| 14911 | 070144 | | | | JSR PC,FPRINT | |
| 14912 | 070144 | | | | RTS | PC |
| (6) | 070144 | 012746 | 007466 | | 1\$: | |
| (3) | 070150 | 010600 | | | YLOCC: | |
| (4) | 070152 | 004737 | 065304 | | PRINTB | #ALOCC |
| 14913 | 070156 | 004737 | 071554 | | MOV | #ALOCC,-(SP) |
| 14914 | 070162 | 000405 | | | MOV | SP,R0 |
| 14915 | 070164 | | | | JSR PC,FPRINT | |
| (6) | 070164 | 012746 | 010150 | | JSR | PC,PRNTIQ ;PRINT TEST # |
| (3) | 070170 | 010600 | | | BR 1\$ | |
| (4) | 070172 | 004737 | 065304 | | PRINTB | #FORM3 |
| 14916 | 070176 | 000207 | | | MOV | #FORM3,-(SP) |
| 14917 | 070200 | | | | MOV | SP,R0 |
| 14918 | 070200 | | | | JSR PC,FPRINT | |
| (6) | 070200 | 012746 | 007476 | | RTS | PC |
| | | | | | 1\$: | |
| | | | | | YSKPC: | |
| | | | | | PRINTB | #ASKPC |
| | | | | | MOV | #ASKPC,-(SP) |

| | | | | | | |
|-------|--------|--------|--------|--------|----------------|---------------|
| (3) | 070204 | 010600 | | MOV | SP,R0 | |
| (4) | 070206 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14919 | 070212 | 004737 | 071554 | JSR | PC,PRNTIQ | :PRINT TEST # |
| 14920 | 070216 | 000405 | | BR | 1\$ | |
| 14921 | 070220 | | | PRINTB | #FORM3 | |
| (6) | 070220 | 012746 | 010150 | MOV | #FORM3,-(SP) | |
| (3) | 070224 | 010600 | | MOV | SP,R0 | |
| (4) | 070226 | 004737 | 065304 | JSR | PC,FPRINT | |
| 14922 | 070232 | 000207 | | RTS | PC | |
| 14923 | 070234 | | | | | |
| 14924 | 070234 | | | | | |
| (6) | 070234 | 012746 | 007506 | PRINTB | #ASCANC | |
| (3) | 070240 | 010600 | | MOV | #ASCANC,-(SP) | |
| (4) | 070242 | 004737 | 065304 | MOV | SP,R0 | |
| 14925 | 070246 | 004737 | 071554 | JSR | PC,FPRINT | |
| 14926 | 070252 | 000405 | | JSR | PC,PRNTIQ | :PRINT TEST # |
| 14927 | 070254 | | | BR | 1\$ | |
| (6) | 070254 | 012746 | 010216 | PRINTB | #FORM4 | |
| (3) | 070260 | 010600 | | MOV | #FORM4,-(SP) | |
| (4) | 070262 | 004737 | 065304 | MOV | SP,R0 | |
| 14928 | 070266 | 000207 | | JSR | PC,FPRINT | |
| 14929 | 070270 | | | RTS | PC | |
| 14930 | 070270 | | | | | |
| (6) | 070270 | 012746 | 007517 | PRINTB | #ASPANC | |
| (3) | 070274 | 010600 | | MOV | #ASPANC,-(SP) | |
| (4) | 070276 | 004737 | 065304 | MOV | SP,R0 | |
| 14931 | 070302 | 004737 | 071554 | JSR | PC,FPRINT | |
| 14932 | 070306 | 000405 | | JSR | PC,PRNTIQ | :PRINT TEST # |
| 14933 | 070310 | | | BR | 1\$ | |
| (6) | 070310 | 012746 | 010216 | PRINTB | #FORM4 | |
| (3) | 070314 | 010600 | | MOV | #FORM4,-(SP) | |
| (4) | 070316 | 004737 | 065304 | MOV | SP,R0 | |
| 14934 | 070322 | 000207 | | JSR | PC,FPRINT | |
| 14935 | 070324 | | | RTS | PC | |
| 14936 | 070324 | | | | | |
| (6) | 070324 | 012746 | 007530 | PRINTB | #ACMPC | |
| (3) | 070330 | 010600 | | MOV | #ACMPC,-(SP) | |
| (4) | 070332 | 004737 | 065304 | MOV | SP,R0 | |
| 14937 | 070336 | 004737 | 071554 | JSR | PC,FPRINT | |
| 14938 | 070342 | 000405 | | JSR | PC,PRNTIQ | :PRINT TEST # |
| 14939 | 070344 | | | BR | 1\$ | |
| (6) | 070344 | 012746 | 010272 | PRINTB | #FORM5 | |
| (3) | 070350 | 010600 | | MOV | #FORM5,-(SP) | |
| (4) | 070352 | 004737 | 065304 | MOV | SP,R0 | |
| 14940 | 070356 | 000207 | | JSR | PC,FPRINT | |
| 14941 | 070360 | | | RTS | PC | |
| 14942 | 070360 | | | | | |
| (6) | 070360 | 012746 | 007540 | PRINTB | #AMATCHC | |
| (3) | 070364 | 010600 | | MOV | #AMATCHC,-(SP) | |
| (4) | 070366 | 004737 | 065304 | MOV | SP,R0 | |
| 14943 | 070372 | 004737 | 071554 | JSR | PC,FPRINT | |
| 14944 | 070376 | 000405 | | JSR | PC,PRNTIQ | :PRINT TEST # |
| 14945 | 070400 | | | BR | 1\$ | |
| (6) | 070400 | 012746 | 010356 | PRINTB | #FORM6 | |
| | | | | MOV | #FORM6,-(SP) | |

| | | | | | | |
|-------|--------|--------|--------|---------|--------------------|--|
| (3) | 070404 | 010600 | | | MOV SP,R0 | |
| (4) | 070406 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14946 | 070412 | 000207 | | 1\$: | RTS PC | |
| 14947 | 070414 | | | YADDN: | | |
| 14948 | 070414 | | | | PRINTB #AADDN | |
| (6) | 070414 | 012746 | 007550 | | MOV #AADDN,-(SP) | |
| (3) | 070420 | 010600 | | | MOV SP,R0 | |
| (4) | 070422 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14949 | 070426 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 14950 | 070432 | 000410 | | | BR 1\$ | |
| 14951 | 070434 | | | | PRINTB #FORM7 | |
| (6) | 070434 | 012746 | 010430 | | MOV #FORM7,-(SP) | |
| (3) | 070440 | 010600 | | | MOV SP,R0 | |
| (4) | 070442 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14952 | 070446 | 012737 | 000016 | 002466 | MOV #000016,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2,DST |
| 14953 | 070454 | 000207 | | 1\$: | RTS PC | |
| 14954 | 070456 | | | YSUBN: | | |
| 14955 | 070456 | | | | PRINTB #ASUBN | |
| (6) | 070456 | 012746 | 007560 | | MOV #ASUBN,-(SP) | |
| (3) | 070462 | 010600 | | | MOV SP,R0 | |
| (4) | 070464 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14956 | 070470 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 14957 | 070474 | 000410 | | | BR 1\$ | |
| 14958 | 070476 | | | | PRINTB #FORM7 | |
| (6) | 070476 | 012746 | 010430 | | MOV #FORM7,-(SP) | |
| (3) | 070502 | 010600 | | | MOV SP,R0 | |
| (4) | 070504 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14959 | 070510 | 012737 | 000016 | 002466 | MOV #000016,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2,DST. |
| 14960 | 070516 | 000207 | | 1\$: | RTS PC | |
| 14961 | 070520 | | | YCMPN: | | |
| 14962 | 070520 | | | | PRINTB #ACMPN | |
| (6) | 070520 | 012746 | 007570 | | MOV #ACMPN,-(SP) | |
| (3) | 070524 | 010600 | | | MOV SP,R0 | |
| (4) | 070526 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14963 | 070532 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 14964 | 070536 | 000410 | | | BR 1\$ | |
| 14965 | 070540 | | | | PRINTB #FORM8 | |
| (6) | 070540 | 012746 | 010524 | | MOV #FORM8,-(SP) | |
| (3) | 070544 | 010600 | | | MOV SP,R0 | |
| (4) | 070546 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14966 | 070552 | 012737 | 000006 | 002466 | MOV #000006,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2. |
| 14967 | 070560 | 000207 | | 1\$: | RTS PC | |
| 14968 | 070562 | | | YCVTNL: | | |
| 14969 | 070562 | | | | PRINTB #ACVTNL | |
| (6) | 070562 | 012746 | 007600 | | MOV #ACVTNL,-(SP) | |
| (3) | 070566 | 010600 | | | MOV SP,R0 | |
| (4) | 070570 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14970 | 070574 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 14971 | 070600 | 000410 | | | BR 1\$ | |
| 14972 | 070602 | | | | PRINTB #FORM9 | |
| (6) | 070602 | 012746 | 010602 | | MOV #FORM9,-(SP) | |
| (3) | 070606 | 010600 | | | MOV SP,R0 | |
| (4) | 070610 | 004737 | 065304 | | JSR PC,FPRINT | |
| 14973 | 070614 | 012737 | 000001 | 002466 | MOV #000001,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED SRC. |

| | | | | | | | |
|-------|--------|--------|--------|---------|--------|----------------|---|
| 14974 | 070622 | 000207 | | 1\$: | RTS | PC | |
| 14975 | 070624 | | | YCVTPN: | | | |
| 14976 | 070624 | | | | PRINTB | #ACVTPN | |
| (6) | 070624 | 012746 | 007611 | | MOV | #ACVTPN,-(SP) | |
| (3) | 070630 | 010600 | | | MOV | SP,R0 | |
| (4) | 070632 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14977 | 070636 | 004737 | 071554 | | JSR | PC,PRNTIQ | ;PRINT TEST # |
| 14978 | 070642 | 000410 | | | BR | 1\$ | |
| 14979 | 070644 | | | | PRINTB | #FORM10 | |
| (6) | 070644 | 012746 | 010656 | | MOV | #FORM10,-(SP) | |
| (3) | 070650 | 010600 | | | MOV | SP,R0 | |
| (4) | 070652 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14980 | 070656 | 012737 | 000420 | 002466 | MOV | #000420,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC AND ZONED DST. |
| 14981 | 070664 | 000207 | | 1\$: | RTS | PC | |
| 14982 | 070666 | | | YCVTNP: | | | |
| 14983 | 070666 | | | | PRINTB | #ACVTNP | |
| (6) | 070666 | 012746 | 007622 | | MOV | #ACVTNP,-(SP) | |
| (3) | 070672 | 010600 | | | MOV | SP,R0 | |
| (4) | 070674 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14984 | 070700 | 004737 | 071554 | | JSR | PC,PRNTIQ | ;PRINT TEST # |
| 14985 | 070704 | 000410 | | | BR | 1\$ | |
| 14986 | 070706 | | | | PRINTB | #FORM10 | |
| (6) | 070706 | 012746 | 010656 | | MOV | #FORM10,-(SP) | |
| (3) | 070712 | 010600 | | | MOV | SP,R0 | |
| (4) | 070714 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14987 | 070720 | 012737 | 010001 | 002466 | MOV | #010001,PZCODE | ;SET PRINTCODE TO DSIPLAY ZONED SRC AND PACKED DST. |
| 14988 | 070726 | 000207 | | 1\$: | RTS | PC | |
| 14989 | 070730 | | | YASHN: | | | |
| 14990 | 070730 | | | | PRINTB | #AASHN | |
| (6) | 070730 | 012746 | 007633 | | MOV | #AASHN,-(SP) | |
| (3) | 070734 | 010600 | | | MOV | SP,R0 | |
| (4) | 070736 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14991 | 070742 | 004737 | 071554 | | JSR | PC,PRNTIQ | ;PRINT TEST # |
| 14992 | 070746 | 000410 | | | BR | 1\$ | |
| 14993 | 070750 | | | | PRINTB | #FORM11 | |
| (6) | 070750 | 012746 | 010730 | | MOV | #FORM11,-(SP) | |
| (3) | 070754 | 010600 | | | MOV | SP,R0 | |
| (4) | 070756 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14994 | 070762 | 012737 | 000021 | 002466 | MOV | #000021,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED SRC,DST. |
| 14995 | 070770 | 000207 | | 1\$: | RTS | PC | |
| 14996 | 070772 | | | YCVTLN: | | | |
| 14997 | 070772 | | | | PRINTB | #ACVTLN | |
| (6) | 070772 | 012746 | 007643 | | MOV | #ACVTLN,-(SP) | |
| (3) | 070776 | 010600 | | | MOV | SP,R0 | |
| (4) | 071000 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 14998 | 071004 | 004737 | 071554 | | JSR | PC,PRNTIQ | ;PRINT TEST # |
| 14999 | 071010 | 000410 | | | BR | 1\$ | |
| 15000 | 071012 | | | | PRINTB | #FORM12 | |
| (6) | 071012 | 012746 | 011012 | | MOV | #FORM12,-(SP) | |
| (3) | 071016 | 010600 | | | MOV | SP,R0 | |
| (4) | 071020 | 004737 | 065304 | | JSR | PC,FPRINT | |
| 15001 | 071024 | 012737 | 000040 | 002466 | MOV | #000040,PZCODE | ;SET PRINTCODE TO DISPLAY ZONED DST. |
| 15002 | 071032 | 000207 | | 1\$: | RTS | PC | |
| 15003 | 071034 | | | YADDP: | | | |

| | | | | | | |
|-------|--------|--------|--------|--------|--------------------|---|
| 15004 | 071034 | | | | PRINTB #AADDP | |
| (6) | 071034 | 012746 | 007654 | | MOV #AADDP,-(SP) | |
| (3) | 071040 | 010600 | | | MOV SP,R0 | |
| (4) | 071042 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15005 | 071046 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15006 | 071052 | 000410 | | | BR 1\$ | |
| 15007 | 071054 | | | | PRINTB #FORM7 | |
| (6) | 071054 | 012746 | 010430 | | MOV #FORM7,-(SP) | |
| (3) | 071060 | 010600 | | | MOV SP,R0 | |
| (4) | 071062 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15008 | 071066 | 012737 | 007000 | 002466 | MOV #007000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST. |
| 15009 | 071074 | 000207 | | | RTS PC | |
| 15010 | 071076 | | | | 1\$: YSUBP: | |
| 15011 | 071076 | | | | PRINTB #ASUBP | |
| (6) | 071076 | 012746 | 007664 | | MOV #ASUBP,-(SP) | |
| (3) | 071102 | 010600 | | | MOV SP,R0 | |
| (4) | 071104 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15012 | 071110 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15013 | 071114 | 000410 | | | BR 1\$ | |
| 15014 | 071116 | | | | PRINTB #FORM7 | |
| (6) | 071116 | 012746 | 010430 | | MOV #FORM7,-(SP) | |
| (3) | 071122 | 010600 | | | MOV SP,R0 | |
| (4) | 071124 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15015 | 071130 | 012737 | 007000 | 002466 | MOV #007000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST. |
| 15016 | 071136 | 000207 | | | RTS PC | |
| 15017 | 071140 | | | | 1\$: YCMPP: | |
| 15018 | 071140 | | | | PRINTB #ACMPP | |
| (6) | 071140 | 012746 | 007674 | | MOV #ACMPP,-(SP) | |
| (3) | 071144 | 010600 | | | MOV SP,R0 | |
| (4) | 071146 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15019 | 071152 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15020 | 071156 | 000410 | | | BR 1\$ | |
| 15021 | 071160 | | | | PRINTB #FORM8 | |
| (6) | 071160 | 012746 | 010524 | | MOV #FORM8,-(SP) | |
| (3) | 071164 | 010600 | | | MOV SP,R0 | |
| (4) | 071166 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15022 | 071172 | 012737 | 003000 | 002466 | MOV #003000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC1, AND SRC2. |
| 15023 | 071200 | 000207 | | | RTS PC | |
| 15024 | 071202 | | | | 1\$: YCVTPL: | |
| 15025 | 071202 | | | | PRINTB #ACVTPL | |
| (6) | 071202 | 012746 | 007704 | | MOV #ACVTPL,-(SP) | |
| (3) | 071206 | 010600 | | | MOV SP,R0 | |
| (4) | 071210 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15026 | 071214 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15027 | 071220 | 000410 | | | BR 1\$ | |
| 15028 | 071222 | | | | PRINTB #FORM9 | |
| (6) | 071222 | 012746 | 010602 | | MOV #FORM9,-(SP) | |
| (3) | 071226 | 010600 | | | MOV SP,R0 | |
| (4) | 071230 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15029 | 071234 | 012737 | 000400 | 002466 | MOV #000400,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC. |
| 15030 | 071242 | 000207 | | | RTS PC | |
| 15031 | 071244 | | | | 1\$: YMULP: | |
| 15032 | 071244 | | | | PRINTB #AMULP | |
| (6) | 071244 | 012746 | 007715 | | MOV #AMULP,-(SP) | |

| | | | | | | |
|-------|--------|--------|--------|--------|--------------------|---|
| (3) | 071250 | 010600 | | | MOV SP,R0 | |
| (4) | 071252 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15033 | 071256 | 004737 | 071554 | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15034 | 071262 | 000410 | | | BR 1\$ | |
| 15035 | 071264 | | | | PRINTB #FORM7 | |
| (6) | 071264 | 012746 | 010430 | | MOV #FORM7,-(SP) | |
| (3) | 071270 | 010600 | | | MOV SP,R0 | |
| (4) | 071272 | 004737 | 065304 | | JSR PC,FPRINT | |
| 15036 | 071276 | 012737 | 007000 | 002466 | MOV #007000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST. |
| 15037 | 071304 | 000207 | | | RTS PC | |
| 15038 | 071306 | | | | | |
| 15039 | 071306 | | | | | |
| (6) | 071306 | 012746 | 007725 | | PRINTB #ADIVP | |
| (3) | 071312 | 010600 | | | MOV #ADIVP,-(SP) | |
| (4) | 071314 | 004737 | 065304 | | MOV SP,R0 | |
| 15040 | 071320 | 004737 | 071554 | | JSR PC,FPRINT | |
| 15041 | 071324 | 000410 | | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15042 | 071326 | | | | BR 1\$ | |
| (6) | 071326 | 012746 | 010430 | | PRINTB #FORM7 | |
| (3) | 071332 | 010600 | | | MOV #FORM7,-(SP) | |
| (4) | 071334 | 004737 | 065304 | | MOV SP,R0 | |
| 15043 | 071340 | 012737 | 007000 | 002466 | JSR PC,FPRINT | |
| 15044 | 071346 | 000207 | | | MOV #007000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST. |
| 15045 | 071350 | | | | RTS PC | |
| 15046 | 071350 | | | | | |
| (6) | 071350 | 012746 | 007735 | | PRINTB #AASHP | |
| (3) | 071354 | 010600 | | | MOV #AASHP,-(SP) | |
| (4) | 071356 | 004737 | 065304 | | MOV SP,R0 | |
| 15047 | 071362 | 004737 | 071554 | | JSR PC,FPRINT | |
| 15048 | 071366 | 000410 | | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15049 | 071370 | | | | BR 1\$ | |
| (6) | 071370 | 012746 | 010730 | | PRINTB #FORM11 | |
| (3) | 071374 | 010600 | | | MOV #FORM11,-(SP) | |
| (4) | 071376 | 004737 | 065304 | | MOV SP,R0 | |
| 15050 | 071402 | 012737 | 010400 | 002466 | JSR PC,FPRINT | |
| 15051 | 071410 | 000207 | | | MOV #010400,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED SRC,DST. |
| 15052 | 071412 | | | | RTS PC | |
| 15053 | 071412 | | | | | |
| (6) | 071412 | 012746 | 007745 | | PRINTB #ACVTLP | |
| (3) | 071416 | 010600 | | | MOV #ACVTLP,-(SP) | |
| (4) | 071420 | 004737 | 065304 | | MOV SP,R0 | |
| 15054 | 071424 | 004737 | 071554 | | JSR PC,FPRINT | |
| 15055 | 071430 | 000410 | | | JSR PC,PRNTIQ | ;PRINT TEST # |
| 15056 | 071432 | | | | BR 1\$ | |
| (6) | 071432 | 012746 | 011012 | | PRINTB #FORM12 | |
| (3) | 071436 | 010600 | | | MOV #FORM12,-(SP) | |
| (4) | 071440 | 004737 | 065304 | | MOV SP,R0 | |
| 15057 | 071444 | 012737 | 020000 | 002466 | JSR PC,FPRINT | |
| 15058 | 071452 | 000207 | | | MOV #020000,PZCODE | ;SET PRINTCODE TO DISPLAY PACKED DST. |
| 15059 | 071454 | | | | RTS PC | |
| 15060 | 071454 | | | | | |
| (7) | 071454 | 013746 | 047464 | | PRINTB #AL2D,TINST | |
| (6) | 071460 | 012746 | 007756 | | MOV TINST,-(SP) | |
| (3) | 071464 | 010600 | | | MOV #AL2D,-(SP) | |
| | | | | | MOV SP,R0 | |

```

(4) 071466 004737 065304 JSR PC,FPRINT
15061 071472 004737 071554 JSR PC,PRNTIQ
15062 071476 000405 BR 1$
15063 071500 PRINTB #FORM41
(6) 071500 012746 014573 MOV #FORM41,-(SP)
(3) 071504 010600 MOV SP,R0
(4) 071506 004737 065304 JSR PC,FPRINT
15064 071512 000207 1$: RTS PC
15065 071514 YL3D:
15066 071514 PRINTB #AL3D,TINST
(7) 071514 013746 047464 MOV TINST,-(SP)
(6) 071520 012746 007770 MOV #AL3D,-(SP)
(3) 071524 010600 MOV SP,R0
(4) 071526 004737 065304 JSR PC,FPRINT
15067 071532 004737 071554 JSR PC,PRNTIQ
15068 071536 000405 BR 1$
15069 071540 PRINTB #FORM41
(6) 071540 012746 014573 MOV #FORM41,-(SP)
(3) 071544 010600 MOV SP,R0
(4) 071546 004737 065304 JSR PC,FPRINT
15070 071552 000207 1$: RTS PC
15071
15072
15073 :SUBROUTINE TO PRINT CURRENT TEST # (TOTTC)
15074
15075 071554 032737 000100 047464 PRNTIQ: BIT #100,TINST ;SUBROUTINE TO PRINT I ON END OF IN-LINE OPCODE
15076 ; NEUMONIC AND TEST NUMBER.
15077 ; IS OPCODE UNDER TEST AN IN-LINE TYPE?
15078 071562 001405 BEQ 1$ ;BRANCH IF NO
15079 071564 PRINTB #FORM31 ;PRINT I
(6) 071564 012746 014176 MOV #FORM31,-(SP)
(3) 071570 010600 MOV SP,R0
(4) 071572 004737 065304 JSR PC,FPRINT
15080 071576 1$: PRINTB #FORM30,TOTTCH,TOTTC ;PRINT TEST #
(8) 071576 013746 001420 MOV TOTTC,-(SP)
(7) 071602 013746 001416 MOV TOTTCH,-(SP)
(6) 071606 012746 014146 MOV #FORM30,-(SP)
(3) 071612 010600 MOV SP,R0
(4) 071614 004737 065304 JSR PC,FPRINT
15081 071620 005737 001762 TST CACT ;CNTL-T REQUEST ACTIVE?
15082 071624 001066 BNE 3$ ;BRANCH IF YES
15083 071626 PRINTB #FORM40,INTCT,REGSET ;PRINT INTERRUPT COUNT AND REG SET
(8) 071626 013746 002052 MOV REGSET,-(SP)
(7) 071632 013746 002542 MOV INTCT,-(SP)
(6) 071636 012746 014513 MOV #FORM40,-(SP)
(3) 071642 010600 MOV SP,R0
(4) 071644 004737 065304 JSR PC,FPRINT
15084 071650 062716 000002 ADD #2,(SP)
15085 071654 005737 002160 TST MODE ;FORM ASCII MODE CHARACTER
15086 071660 001004 BNE 4$
15087 071662 112737 000113 066016 MOVB #113,TDIG ;KERNEL MODE (K)
15088 071670 000413 BR 10$
15089 071672 022737 000001 002160 4$: CMP #1,MODE
15090 071700 001004 BNE 5$

```

| | | | | | | | | |
|-------|--------|--------|--------|--------|-------|-------------------|--|----------------------|
| 15091 | 071702 | 112737 | 000123 | 066016 | | MOVB #123,TDIG | | ;SUPERVISOR MODE (S) |
| 15092 | 071710 | 000403 | | | | BR 10\$ | | |
| 15093 | 071712 | 112737 | 000125 | 066016 | 5\$: | MOVB #125,TDIG | | ;USER MODE (U) |
| 15094 | 071720 | 104400 | | | 10\$: | TYPE | | |
| 15095 | 071722 | 066016 | | | | TDIG | | |
| 15096 | 071724 | | | | | PRINTB #FRM40A | | ;PRINT D EN |
| (6) | 071724 | 012746 | 014561 | | | MOV #FRM40A,-(SP) | | |
| (3) | 071730 | 010600 | | | | MOV SP,R0 | | |
| (4) | 071732 | 004737 | 065304 | | | JSR PC,FPRINT | | |
| 15097 | 071736 | 005737 | 002162 | | | TST DEN | | ;PRINT Y OR N |
| 15098 | 071742 | 001004 | | | | BNE 6\$ | | |
| 15099 | 071744 | 112737 | 000116 | 066016 | | MOVB #116,TDIG | | |
| 15100 | 071752 | 000403 | | | | BR 11\$ | | |
| 15101 | 071754 | 112737 | 000131 | 066016 | 6\$: | MOVB #131,TDIG | | |
| 15102 | 071762 | 104400 | | | 11\$: | TYPE | | |
| 15103 | 071764 | 066016 | | | | TDIG | | |
| 15104 | 071766 | | | | | PRINTB #FORM21 | | ;PRINT CRLF |
| (6) | 071766 | 012746 | 014020 | | | MOV #FORM21,-(SP) | | |
| (3) | 071772 | 010600 | | | | MOV SP,R0 | | |
| (4) | 071774 | 004737 | 065304 | | | JSR PC,FPRINT | | |
| 15105 | 072000 | 000405 | | | | BR 2\$ | | |
| 15106 | 072002 | | | | 3\$: | PRINTB #FORM21 | | ;PRINT CRLF |
| (6) | 072002 | 012746 | 014020 | | | MOV #FORM21,-(SP) | | |
| (3) | 072006 | 010600 | | | | MOV SP,R0 | | |
| (4) | 072010 | 004737 | 065304 | | | JSR PC,FPRINT | | |
| 15107 | 072014 | 005037 | 001762 | | 2\$: | CLR CTACT | | ;CLEAR CNTL-T FLAG |
| 15108 | 072020 | 000207 | | | | RTS PC | | |
| 15109 | | | | | | | | |
| 15110 | | | | | | | | |
| 15111 | | | | | | .EVEN | | |
| 15112 | | | | | | | | |
| 15113 | | | | | | | | |

```
15116  
15117  
15118  
15119  
15120 072022 000000  
15121 072024 000000  
15122 072026 000000  
15123 072030 000000  
15124 072032 000000  
15125 072034 000000  
15126 072036 000000  
15127 072040 000000  
15128 072042 000000  
15129 072044 000000  
15130 072046 000000  
15131 072050 000000  
15132 072052 000000  
15133 072054 000000  
15134 072056 000000  
15135 072060 000000  
15136 072062 000000  
15137 072064 000000  
15138 072066 000000  
15139 072070 000000  
15140 072072 000000  
15141 072074 000000  
15142
```

```
.SBTTL DUMMY INPUT TABLE (FOR RANDOM MODE)  
:DUMMY INPUT TABLE - USED ONLY IN RANDOM EXERCISE MODE  
:IDUM: .WORD 0 :INST  
:TYPE = 0  
:IP1  
:IP2  
:IP3  
:IP4  
:IP5  
:IP6  
:IP7  
:IP10  
:IP11  
:IP12  
:IP13  
:IP14  
:IP15  
:IP16  
:IP17  
:IP20  
:IP21  
:IP22  
:IP23  
:IP24  
:IDUME: .WORD 0
```

15144
15145
15146
15147
15148 072076 000000
15149

.SBTTL CIS INST INPUT TABLES
:INPUT TABLES
:
:INPTBL: .WORD 0
:

```
15151          .SBTTL          L2D TABLES
15152          :
15153          :ENTRY 0.1 - INSTRUCTION UNDER TEST = L2DR
15154          :
15155          :L2D:      .WORD 32          ;INST=L2DR; NOTE:R IS CALCULATED USING IP7.
15156          :          .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
15157          :          .WORD T53        ;IP1 - R0 PATTERN
15158          :          .WORD T54        ;IP2 - R1 PATTERN
15159          :          .WORD T55        ;IP3 - R2 PATTERN
15160          :          .WORD T56        ;IP4 - R3 PATTERN
15161          :          .WORD T57        ;IP5 - R4 PATTERN
15162          :          .WORD T60        ;IP6 - R5 PATTERN
15163          :          .WORD T61        ;IP7 - DESC ADDRESS
15164          :          .WORD 0          ;IP10
15165          :          .WORD 0          ;IP11
15166          :          .WORD 0          ;IP12
15167          :          .WORD 0          ;IP13
15168          :          .WORD 0          ;IP14
15169          :          .WORD 0          ;IP15
15170          :          .WORD 0          ;IP16
15171          :          .WORD 0          ;IP17
15172          :          .WORD 0          ;IP20
15173          :          .WORD 0          ;IP21
15174          :          .WORD 0          ;IP22
15175          :          .WORD 0          ;IP23
15176          :          .WORD 0          ;IP24
15177
15178          ;TOTAL # OF TESTS = 8
```


15180
 15181
 15182
 15183
 15184 072154 000033
 15185 072156 000003
 15186 072160 107474
 15187 072162 107504
 15188 072164 107514
 15189 072166 107524
 15190 072170 107534
 15191 072172 107544
 15192 072174 107554
 15193 072176 000000
 15194 072200 000000
 15195 072202 000000
 15196 072204 000000
 15197 072206 000000
 15198 072210 000000
 15199 072212 000000
 15200 072214 000000
 15201 072216 000000
 15202 072220 000000
 15203 072222 000000
 15204 072224 000000
 15205 072226 000000
 15206
 15207

```

.SBTTL          L3D TABLES
:ENTRY 0.2 - INSTRUCTION UNDER TEST = L3DR
:IL3D:          .WORD 33          ;INST=L3DR; NOTE:R IS CALCULATED USING IP7.
                .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
                .WORD T53        ;IP1 - R0 PATTERN
                .WORD T54        ;IP2 - R1 PATTERN
                .WORD T55        ;IP3 - R2 PATTERN
                .WORD T56        ;IP4 - R3 PATTERN
                .WORD T57        ;IP5 - R4 PATTERN
                .WORD T60        ;IP6 - R5 PATTERN
                .WORD T61        ;IP7 - DESC ADDRESS
                .WORD 0          ;IP10
                .WORD 0          ;IP11
                .WORD 0          ;IP12
                .WORD 0          ;IP13
                .WORD 0          ;IP14
                .WORD 0          ;IP15
                .WORD 0          ;IP16
                .WORD 0          ;IP17
                .WORD 0          ;IP20
                .WORD 0          ;IP21
                .WORD 0          ;IP22
                .WORD 0          ;IP23
                .WORD 0          ;IP24

;TOTAL # OF TESTS = 8

```

15210
15211
15212
15213 072230 000001
15214 072232 000003
15215 072234 102752
15216 072236 103012
15217 072240 102772
15218 072242 103046
15219 072244 103704
15220 072246 103734
15221 072250 103766
15222 072252 104004
15223 072254 104020
15224 072256 104036
15225 072260 104054
15226 072262 104070
15227 072264 000000
15228 072266 102436
15229 072270 000000
15230 072272 000000
15231 072274 000000
15232 072276 000000
15233 072300 000000
15234 072302 000000
15235
15236
15237
15238
15239
15240
15241
15242
15243
15244
15245
15246
15247
15248
15249
15250
15251
15252

.SBTTL MOV C TABLES
:ENTRY 1 - INST UNDER TEST = MOV C

:MOV C: .WORD 1 :INST=MOV C
.WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
.WORD TL1C :IP1 - SRC.LEN
.WORD T2 :IP2 - SRC.ADR
.WORD TL2C :IP3 - DST.LEN
.WORD T4 :IP4 - DST.ADR
.WORD T5 :IP5 - FILL
.WORD T6 :IP6 - SRC DATA
.WORD T7 :IP7 - SRC SURR DATA
.WORD T10 :IP10 - SRC.SURR.LEN
.WORD T11 :IP11 - DST DATA
.WORD T12 :IP12 - DST SURR DATA
.WORD T13 :IP13 - DST.SURR.LEN
.WORD T14 :IP14 - SEPARATION CONSTANT
.WORD 0 :IP15 - (TRANSLATION TABLE FOR MOVTC)
.WORD T0 :IP16 - SPECIAL HANDLING
.WORD 0 :IP17
.WORD 0 :IP20
.WORD 0 :IP21
.WORD 0 :IP22
.WORD 0 :IP23
.WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:
: SOURCE LENGTH - 0,1,300
: DESTINATION LENGTH - 0,1,5
: SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD < DEST. AD
: - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD > DEST. AD
: - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
: - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
: - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
: - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
: - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START # = 1.
: TOTAL # OF TEST CONDITIONS = 48
: TOTAL # OF TESTS = (1 REG. + 1 INLINE)48 = 96
:

15254
15255
15256 072304 000001
15257 072306 000003
15258 072310 102462
15259 072312 103030
15260 072314 102462
15261 072316 103310
15262 072320 103704
15263 072322 103756
15264 072324 102436
15265 072326 102436
15266 072330 102436
15267 072332 102436
15268 072334 102436
15269 072336 104070
15270 072340 000000
15271 072342 102706
15272 072344 000000
15273 072346 000000
15274 072350 000000
15275 072352 000000
15276 072354 000000
15277 072356 000000
15278
15279
15280
15281
15282
15283
15284
15285
15286
15287
15288
15289
15290

:ENTRY 2 - INST UNDER TEST = MOV C

```

:IMOV C1: .WORD 1          :INST=MOV C
          .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1A        :IP1 - SRC.LEN
          .WORD T2AA       :IP2 - SRC.ADR
          .WORD T1A        :IP3 - DST.LEN
          .WORD T4A        :IP4 - DST.ADR
          .WORD T5         :IP5 - FILL
          .WORD T6A        :IP6 - SRC DATA
          .WORD T0         :IP7 - SRC SURR DATA
          .WORD T0         :IP10 - SRC.SURR.LEN
          .WORD T0         :IP11 - DST DATA
          .WORD T0         :IP12 - DST SURR DATA
          .WORD T0         :IP13 - DST.SURR.LEN
          .WORD T14        :IP14 - SEPARATION CONSTANT
          .WORD 0          :IP15 - (TRANSLATION TABLE FOR MOV C)
          .WORD TSPA       :IP16 - SPECIAL HANDLING
          .WORD 0          :IP17
          .WORD 0          :IP20
          .WORD 0          :IP21
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
:                       - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START#=0

```

```

: TOTAL # OF TEST CONDITIONS = 256
: TOTAL # OF TESTS = (1 REG.)256 = 256
:

```

15292
15293
15294 072360 000001
15295 072362 000000
15296 072364 100010
15297 072366 000201
15298 072370 000011
15299 072372 000224
15300 072374 000377
15301 072376 103736
15302 072400 000240
15303 072402 000000
15304 072404 000252
15305 072406 000360
15306 072410 000000
15307 072412 000000
15308 072414 000000
15309 072416 000003
15310 072420 000000
15311 072422 000000
15312 072424 000000
15313 072426 000000
15314 072430 000000
15315 072432 000000
15316
15317
15318
15319
15320
15321
15322
15323
15324
15325
15326
15327

:ENTRY 3 - INST UNDER TEST = MOV C

```
IMOV C2: .WORD 1          :INST=MOV C
          .WORD 0          :TYPE = 0
          .WORD 100010     :IP1 - SRC.LEN
          .WORD 201        :IP2 - SRC.ADR
          .WORD 000011     :IP3 - DST.LEN
          .WORD 224         :IP4 - DST.ADR
          .WORD 377         :IP5 - FILL
          .WORD T6+2       :IP6 - SRC DATA
          .WORD 240         :IP7 - SRC SURR DATA
          .WORD 0           :IP10 - SRC.SURR.LEN
          .WORD 252         :IP11 - DST DATA
          .WORD 360         :IP12 - DST SURR DATA
          .WORD 0           :IP13 - DST.SURR.LEN
          .WORD 0           :IP14 - SEPARATION CONSTANT
          .WORD 0           :IP15
          .WORD 3           :IP16 - SPECIAL HANDLING
          .WORD 0           :IP17
          .WORD 0           :IP20
          .WORD 0           :IP21
          .WORD 0           :IP22
          .WORD 0           :IP23
          .WORD 0           :IP24
```

:THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:

```
:
: SOURCE LENGTH - 100010
: DESTINATION LENGTH - 11
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - 224 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1.
```

```
: THIS TEST WAS ADDED TO EXERCISE & TEST THE MOV C V-BIT OPERATION
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2
:
```

15329
15330
15331
15332
15333 072434 000004
15334 072436 000003
15335 072440 102752
15336 072442 103012
15337 072444 104114
15338 072446 103734
15339 072450 103766
15340 072452 104004
15341 072454 102436
15342 072456 000000
15343 072460 000000
15344 072462 000000
15345 072464 000000
15346 072466 000000
15347 072470 000000
15348 072472 000000
15349 072474 000000
15350 072476 000000
15351 072500 000000
15352 072502 000000
15353 072504 000000
15354 072506 000000
15355
15356
15357
15358
15359
15360
15361
15362
15363
15364
15365

```
.SBTTL          LOCC TABLES
:ENTRY 4 - INSTRUCTION UNDER TEST = LOCC
:LOCC:  .WORD 4          :INST=LOCC
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD TL1C       :IP1 - SRC.LEN
        .WORD T2         :IP2 - SRC.ADR
        .WORD T15        :IP3 - CHAR
        .WORD T6         :IP4 - SRC DATA
        .WORD T7         :IP5 - SRC SURR DATA
        .WORD T10        :IP6 - SRC.SURR.LEN
        .WORD T0         :IP7 - SPECIAL HANDLING
        .WORD 0          :IP10
        .WORD 0          :IP11
        .WORD 0          :IP12
        .WORD 0          :IP13
        .WORD 0          :IP14
        .WORD 0          :IP15
        .WORD 0          :IP16
        .WORD 0          :IP17
        .WORD 0          :IP20
        .WORD 0          :IP21
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
```

```
:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:   SOURCE LENGTH - 0,1,300
:   SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:   CHAR - 004,375,240
:   SOURCE DATA - INCREMENTING SEQUENCE;INC=1,START#=1
:
:TOTAL # OF TEST CONDITIONS = 9
:TOTAL # OF TESTS = (1 REG. + 1 INLINE)9 = 18
:
```

15367
15368
15369
15370 072510 000004
15371 072512 000003
15372 072514 102462
15373 072516 103030
15374 072520 104134
15375 072522 103756
15376 072524 102436
15377 072526 102436
15378 072530 102706
15379 072532 000000
15380 072534 000000
15381 072536 000000
15382 072540 000000
15383 072542 000000
15384 072544 000000
15385 072546 000000
15386 072550 000000
15387 072552 000000
15388 072554 000000
15389 072556 000000
15390 072560 000000
15391 072562 000000
15392
15393
15394
15395
15396
15397
15398
15399
15400
15401
15402

:ENTRY 5 - INSTRUCTION UNDER TEST = LOCC

```
:ILOCC1: .WORD 4 ;INST=LOCC
          .WORD 3 ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1A ;IP1 - SRC.LEN
          .WORD T2AA ;IP2 - SRC.ADR
          .WORD T15A ;IP3 - CHAR
          .WORD T6A ;IP4 - SRC DATA
          .WORD T0 ;IP5 - SRC SURR DATA
          .WORD T0 ;IP6 - SRC.SURR.LEN
          .WORD TSPA ;IP7 - SPECIAL HANDLING
          .WORD 0 ;IP10
          .WORD 0 ;IP11
          .WORD 0 ;IP12
          .WORD 0 ;IP13
          .WORD 0 ;IP14
          .WORD 0 ;IP15
          .WORD 0 ;IP16
          .WORD 0 ;IP17
          .WORD 0 ;IP20
          .WORD 0 ;IP21
          .WORD 0 ;IP22
          .WORD 0 ;IP23
          .WORD 0 ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:SOURCE LENGTH - 0,1,2,3,4,5,11,20
:SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
:CHAR - 004
:SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START # =0
```

```
:TOTAL # OF TEST CONDITIONS = 16
:TOTAL # OF TESTS = (1 REG.)16 = 16
```

15404
15405
15406
15407 072564 000004
15408 072566 000000
15409 072570 100011
15410 072572 000201
15411 072574 000002
15412 072576 103736
15413 072600 000000
15414 072602 000000
15415 072604 000003
15416 072606 000000
15417 072610 000000
15418 072612 000000
15419 072614 000000
15420 072616 000000
15421 072620 000000
15422 072622 000000
15423 072624 000000
15424 072626 000000
15425 072630 000000
15426 072632 000000
15427 072634 000000
15428 072636 000000

:
: ENTRY 6 - INSTRUCTION UNDER TEST = LOCC
:

:ILOCC2: .WORD 4 ;INST=LOCC
: .WORD 0 ;TYPE = 0
: .WORD 100011 ;IP1 - SRC.LEN
: .WORD 201 ;IP2 - SRC.ADR
: .WORD 2 ;IP3 - CHAR
: .WORD T6+2 ;IP4 - SRC DATA DESCRIPTOR ADR
: .WORD 0 ;IP5 - SRC SURR DATA DESCRIPTOR ADR
: .WORD 0 ;IP6 - SRC.SURR.LEN
: .WORD 3 ;IP7 - SPECIAL HANDLING
: .WORD 0 ;IP10
: .WORD 0 ;IP11
: .WORD 0 ;IP12
: .WORD 0 ;IP13
: .WORD 0 ;IP14
: .WORD 0 ;IP15
: .WORD 0 ;IP16
: .WORD 0 ;IP17
: .WORD 0 ;IP20
: .WORD 0 ;IP21
: .WORD 0 ;IP22
: .WORD 0 ;IP23
: .WORD 0 ;IP24

: THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:
:

: SOURCE LENGTH - 100011
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: CHAR - 002
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START#=1

: THIS TEST WAS ADDED TO EXERCISE & TEST THE LOCC V-BIT OPERATION
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2
:

15429
15430
15431
15432
15433
15434
15435
15436
15437
15438
15439

15441
15442
15443
15444
15445 072640 000010
15446 072642 000003
15447 072644 102752
15448 072646 103012
15449 072650 102772
15450 072652 103656
15451 072654 103716
15452 072656 105050
15453 072660 102436
15454 072662 104004
15455 072664 105050
15456 072666 104036
15457 072670 104054
15458 072672 104070
15459 072674 102436
15460 072676 000000
15461 072700 000000
15462 072702 000000
15463 072704 000000
15464 072706 000000
15465 072710 000000
15466 072712 000000
15467
15468
15469
15470
15471
15472
15473
15474
15475
15476
15477
15478
15479
15480
15481

.SBTTL CMPC TABLES
:ENTRY 7 - INSTRUCTION UNDER TEST = CMPC

:CMPC: .WORD 10 ;INST = CMPC
:WORD 3 ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:WORD TL1C ;IP1 - SRC1.LEN
:WORD T2 ;IP2 - SRC1.ADR
:WORD TL2C ;IP3 - SRC2.LEN
:WORD T411 ;IP4 - SRC2.ADR
:WORD T511 ;IP5 - FILL
:WORD T20 ;IP6 - SRC1.DATA
:WORD T0 ;IP7 - SRC1.SURR.DATA
:WORD T10 ;IP10 - SRC1.SURR.LEN
:WORD T20 ;IP11 - SRC2.DATA
:WORD T12 ;IP12 - SRC2.SURR.DATA
:WORD T13 ;IP13 - SRC2.SURR.LEN
:WORD T14 ;IP14 - SEPARATION CONSTANT
:WORD T0 ;IP15 - SPECIAL HANDLING
:WORD 0 ;IP16
:WORD 0 ;IP17
:WORD 0 ;IP20
:WORD 0 ;IP21
:WORD 0 ;IP22
:WORD 0 ;IP23
:WORD 0 ;IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:SOURCE 1 LENGTH - 0,1,300
:SOURCE 2 LENGTH - 0,1,5
:SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE 2 ADDRESS - NO OVERLAP OF SOURCE STRINGS
:FILL CHAR - 201,377,127
:SOURCE 1 DATA - ALL STRING CHARS IDENTICAL = 127
:SOURCE 2 DATA - ALL STRING CHARS IDENTICAL = 127

:TOTAL # OF TEST CONDITIONS = 27
:TOTAL # OF TESTS = (1 REG. + 1 INLINE)27 = 54

15483
 15484
 15485
 15486 072714 000010
 15487 072716 000001
 15488 072720 105140
 15489 072722 103012
 15490 072724 105156
 15491 072726 103046
 15492 072730 103704
 15493 072732 103734
 15494 072734 103766
 15495 072736 104004
 15496 072740 103734
 15497 072742 104036
 15498 072744 104054
 15499 072746 104070
 15500 072750 102436
 15501 072752 000000
 15502 072754 000000
 15503 072756 000000
 15504 072760 000000
 15505 072762 000000
 15506 072764 000000
 15507 072766 000000
 15508
 15509
 15510
 15511
 15512
 15513
 15514
 15515
 15516
 15517
 15518
 15519
 15520
 15521
 15522
 15523
 15524
 15525
 15526
 15527

:ENTRY 8 - INSTRUCTION UNDER TEST = CMPC

```

:CMPC1: .WORD 10          :INST = CMPC
        .WORD 1          :TYPE = 1
        .WORD TL21C     :IP1 - SRC1.LEN
        .WORD T2        :IP2 - SRC1.ADR
        .WORD TL22C     :IP3 - SRC2.LEN
        .WORD T4        :IP4 - SRC2.ADR
        .WORD T5        :IP5 - FILL
        .WORD T6        :IP6 - SRC1.DATA
        .WORD T7        :IP7 - SRC1.SURR.DATA
        .WORD T10       :IP10 - SRC1.SURR.LEN
        .WORD T6        :IP11 - SRC2.DATA
        .WORD T12       :IP12 - SRC2.SURR.DATA
        .WORD T13       :IP13 - SRC2.SURR.LEN
        .WORD T14       :IP14 - SEPARATION CONSTANT
        .WORD T0        :IP15 - SPECIAL HANDLING
        .WORD 0         :IP16
        .WORD 0         :IP17
        .WORD 0         :IP20
        .WORD 0         :IP21
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD < DEST. ADD
:                   - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD > DEST. AD
:                   - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
:                   - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
:                   - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
:                   - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
:                   - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
:
: FILL CHAR - 377
: SOURCE 1 DATA - INCREMENTING SEQUENCE; INC=1,START # =1
: SOURCE 2 DATA - INCREMENTING SEQUENCE; INC=1,START # =1
  
```

```

: TOTAL # OF TEST CONDITIONS = 26
: TOTAL # OF TESTS = (1 REG + 1 IN-LINE)26 = 52
:
  
```

15529
 15530
 15531
 15532 072770 000010
 15533 072772 000003
 15534 072774 102462
 15535 072776 103030
 15536 073000 102462
 15537 073002 103310
 15538 073004 103704
 15539 073006 103756
 15540 073010 102436
 15541 073012 102436
 15542 073014 103756
 15543 073016 102436
 15544 073020 102436
 15545 073022 104070
 15546 073024 102706
 15547 073026 000000
 15548 073030 000000
 15549 073032 000000
 15550 073034 000000
 15551 073036 000000
 15552 073040 000000
 15553 073042 000000

:ENTRY 9 - INSTRUCTION UNDER TEST = CMPC

```

:CMPC2: .WORD 10          ;INST = CMPC
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1A       ;IP1 - SRC1.LEN
        .WORD T2AA      ;IP2 - SRC1.ADR
        .WORD T1A       ;IP3 - SRC2.LEN
        .WORD T4A       ;IP4 - SRC2.ADR
        .WORD T5        ;IP5 - FILL
        .WORD T6A       ;IP6 - SRC1.DATA
        .WORD T0        ;IP7 - SRC1.SURR.DATA
        .WORD T0        ;IP10 - SRC1.SURR.LEN
        .WORD T6A       ;IP11 - SRC2.DATA
        .WORD T0        ;IP12 - SRC2.SURR.DATA
        .WORD T0        ;IP13 - SRC2.SURR.LEN
        .WORD T14       ;IP14 - SEPARATION CONSTANT
        .WORD TSPA      ;IP15 - SPECIAL HANDLING
        .WORD 0         ;IP16
        .WORD 0         ;IP17
        .WORD 0         ;IP20
        .WORD 0         ;IP21
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DEST
:                   - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
:
: FILL CHAR - 006
: SOURCE 1 DATA - INCREMENTING SEQUENCE; INC=1,START # =0
: SOURCE 2 DATA - INCREMENTING SEQUENCE; INC=1,START # =0
  
```

```

:TOTAL # OF TEST CONDITIONS = 256
:TOTAL # OF TESTS = (1 REG.)256 = 256
:
  
```

15554
 15555
 15556
 15557
 15558
 15559
 15560
 15561
 15562
 15563
 15564
 15565
 15566
 15567
 15568

15570
15571
15572
15573
15574 073044 000002
15575 073046 000003
15576 073050 102752
15577 073052 103012
15578 073054 102772
15579 073056 103046
15580 073060 103704
15581 073062 103734
15582 073064 103766
15583 073066 104004
15584 073070 104020
15585 073072 104036
15586 073074 104054
15587 073076 104070
15588 073100 000000
15589 073102 102436
15590 073104 000000
15591 073106 000000
15592 073110 000000
15593 073112 000000
15594 073114 000000
15595 073116 000000
15596
15597
15598
15599
15600
15601
15602
15603
15604
15605
15606
15607
15608
15609
15610
15611
15612
15613

```

.SBTTL      MOVRC TABLES
:ENTRY 10 - INSTRUCTION UNDER TEST = MOVRC
:IMOVR:     .WORD      2           :INST=MOVRC
            .WORD      3           :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
            .WORD      TL1C        :IP1 - SRC.LEN
            .WORD      T2         :IP2 - SRC.ADR
            .WORD      TL2C        :IP3 - DST.LEN
            .WORD      T4         :IP4 - DST.ADR
            .WORD      T5         :IP5 - FILL
            .WORD      T6         :IP6 - SRC DATA
            .WORD      T7         :IP7 - SRC SURR DATA
            .WORD      T10        :IP10 - SRC.SURR.LEN
            .WORD      T11        :IP11 - DST DATA
            .WORD      T12        :IP12 - DST SURR DATA
            .WORD      T13        :IP13 - DST.SURR.LEN
            .WORD      T14        :IP14 - SEPARATION CONSTANT
            .WORD      0          :IP15
            .WORD      T0         :IP16 - SPECIAL HANDLING
            .WORD      0          :IP17
            .WORD      0          :IP20
            .WORD      0          :IP21
            .WORD      0          :IP22
            .WORD      0          :IP23
            .WORD      0          :IP24

```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE LENGTH - 0,1,300
:DESTINATION LENGTH - 0,1,5
:SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD < DEST. AD
:                      - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD > DEST. AD
:                      - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
:                      - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
:                      - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
:                      - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
:                      - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
:SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START # = 1.
:
:TOTAL # OF TEST CONDITIONS = 48
:TOTAL # OF TESTS = (1 REG. + 1 INLINE) 48 = 96
:

```

```

15615
15616
15617
15618 073120 000002
15619 073122 000001
15620 073124 102462
15621 073126 103030
15622 073130 102462
15623 073132 103310
15624 073134 103704
15625 073136 103756
15626 073140 102436
15627 073142 102436
15628 073144 102436
15629 073146 102436
15630 073150 102436
15631 073152 104070
15632 073154 000000
15633 073156 102706
15634 073160 000000
15635 073162 000000
15636 073164 000000
15637 073166 000000
15638 073170 000000
15639 073172 000000

```

```

: ENTRY 11 - INSTRUCTION UNDER TEST = MOVRC
:
:IMOVR1: .WORD 2          :INST=MOVRC
          .WORD 1          :TYPE = 1
          .WORD T1A        :IP1 - SRC.LEN
          .WORD T2AA       :IP2 - SRC.ADR
          .WORD T1A        :IP3 - DST.LEN
          .WORD T4A        :IP4 - DST.ADR
          .WORD T5         :IP5 - FILL
          .WORD T6A        :IP6 - SRC DATA
          .WORD T0         :IP7 - SRC SURR DATA
          .WORD T0         :IP10 - SRC.SURR.LEN
          .WORD T0         :IP11 - DST DATA
          .WORD T0         :IP12 - DST SURR DATA
          .WORD T0         :IP13 - DST.SURR.LEN
          .WORD T14        :IP14 - SEPARATION CONSTANT
          .WORD 0          :IP15
          .WORD TSPA       :IP16 - SPECIAL HANDLING
          .WORD 0          :IP17
          .WORD 0          :IP20
          .WORD 0          :IP21
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24

```

```

15640
15641
15642
15643
15644
15645
15646
15647
15648
15649
15650
15651
15652

```

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD<DEST
:                       - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD>DEST
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START#=0
:
: TOTAL # OF TEST CONDITIONS = 256
: TOTAL # OF TESTS = (1 REG.)256 = 256
:

```

15654
15655
15656
15657 073174 000002
15658 073176 000000
15659 073200 100100
15660 073202 100000
15661 073204 000111
15662 073206 000100
15663 073210 000376
15664 073212 103736
15665 073214 000240
15666 073216 000000
15667 073220 000255
15668 073222 000366
15669 073224 000000
15670 073226 000000
15671 073230 000000
15672 073232 000037
15673 073234 000000
15674 073236 000000
15675 073240 000000
15676 073242 000000
15677 073244 000000
15678 073246 000000

```

:
:ENTRY 12 - INSTRUCTION UNDER TEST = MOVRC
:
:IMOVR2: .WORD 2          :INST=MOVRC
          .WORD 0          :TYPE = 0
          .WORD 100100     :IP1 - SRC.LEN
          .WORD 100000     :IP2 - SRC.ADR
          .WORD 000111     :IP3 - DST.LEN
          .WORD 100        :IP4 - DST.ADR
          .WORD 376        :IP5 - FILL
          .WORD T6+2       :IP6 - SRC DATA
          .WORD 240        :IP7 - SRC SURR DATA
          .WORD 0          :IP10 - SRC.SURR.LEN
          .WORD 255        :IP11 - DST DATA
          .WORD 366       :IP12 - DST SURR DATA
          .WORD 0          :IP13 - DST.SURR.LEN
          .WORD 0          :IP14 - SEPARATION CONSTANT
          .WORD 0          :IP15
          .WORD 37         :IP16 - SPECIAL HANDLING
          .WORD 0          :IP17
          .WORD 0          :IP20
          .WORD 0          :IP21
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24

```

: THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:

```

:
: SOURCE LENGTH - 100100
: DESTINATION LENGTH - 111
: SOURCE ADDRESS - 100000 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - 100 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1.

```

: THIS TEST WAS ADDED TO EXERCISE & TEST THE MOVRC V-BIT OPERATION
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2

15679
15680
15681
15682
15683
15684
15685
15686
15687
15688
15689
15690

15692
15693
15694
15695 073250 000003
15696 073252 000001
15697 073254 102752
15698 073256 103012
15699 073260 102772
15700 073262 103046
15701 073264 103704
15702 073266 103734
15703 073270 103766
15704 073272 104004
15705 073274 104020
15706 073276 104036
15707 073300 104054
15708 073302 104070
15709 073304 105234
15710 073306 102436
15711 073310 000000
15712 073312 000000
15713 073314 000000
15714 073316 000000
15715 073320 000000
15716 073322 000000
15717
15718
15719
15720
15721
15722
15723
15724
15725
15726
15727
15728
15729
15730
15731
15732
15733
15734
15735

```

.SBTTL      MOVTC TABLES
:ENTRY 13 - INSTRUCTION UNDER TEST = MOVTC
:IMOVT:
:WORD 3      :INST=MOVTC
:WORD 1      :TYPE = 1
:WORD TL1C   :IP1 - SRC.LEN
:WORD T2     :IP2 - SRC.ADR
:WORD TL2C   :IP3 - DST.LEN
:WORD T4     :IP4 - DST.ADR
:WORD T5     :IP5 - FILL
:WORD T6     :IP6 - SRC DATA
:WORD T7     :IP7 - SRC SURR DATA
:WORD T10    :IP10 - SRC.SURR.LEN
:WORD T11    :IP11 - DST DATA
:WORD T12    :IP12 - DST SURR DATA
:WORD T13    :IP13 - DST.SURR.LEN
:WORD T14    :IP14 - SEPARATION CONSTANT
:WORD T24    :IP15 - TRANSLATION TABLE
:WORD T0     :IP16 - SPECIAL HANDLING
:WORD 0      :IP17
:WORD 0      :IP20
:WORD 0      :IP21
:WORD 0      :IP22
:WORD 0      :IP23
:WORD 0      :IP24

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

- : SOURCE LENGTH - 0,1,300
- : DESTINATION LENGTH - 0,1,5
- : SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
- : DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD < DEST. AD
- : - NO OVERLAP OF SOURCE & DEST. STRINGS; SOURCE ADD > DEST. AD
- : - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
- : - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
- : - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
- : - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
- : - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
- : SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1.
- : TRANSLATION TABLE DATA = 1 IN LOC 0, 2 IN LOC 1, ETC

: TOTAL # OF TEST CONDITIONS = 48
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) 48 = 96

15737
15738
15739 073324 000003
15740 073326 000003
15741 073330 102462
15742 073332 103030
15743 073334 102462
15744 073336 103310
15745 073340 103704
15746 073342 103756
15747 073344 102436
15748 073346 102436
15749 073350 102436
15750 073352 102436
15751 073354 102436
15752 073356 104070
15753 073360 105234
15754 073362 102706
15755 073364 000000
15756 073366 000000
15757 073370 000000
15758 073372 000000
15759 073374 000000
15760 073376 000000
15761
15762
15763
15764
15765
15766
15767
15768
15769
15770
15771
15772
15773
15774
15775

:ENTRY 14 - INSTRUCTION UNDER TEST = MOVTC

```

:IMOVTC: .WORD 3          ;INST=MOVTC
          .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1A        ;IP1 - SRC.LEN
          .WORD T2AA       ;IP2 - SRC.ADR
          .WORD T1A        ;IP3 - DST.LEN
          .WORD T4A        ;IP4 - DST.ADR
          .WORD T5         ;IP5 - FILL
          .WORD T6A        ;IP6 - SRC DATA
          .WORD T0         ;IP7 - SRC SURR DATA
          .WORD T0         ;IP10 - SRC.SURR.LEN
          .WORD T0         ;IP11 - DST DATA
          .WORD T0         ;IP12 - DST SURR DATA
          .WORD T0         ;IP13 - DST.SURR.LEN
          .WORD T14        ;IP14 - SEPARATION CONSTANT
          .WORD T24        ;IP15 - TRANSLATION TABLE
          .WORD TSPA       ;IP16 - SPECIAL HANDLING
          .WORD 0          ;IP17
          .WORD 0          ;IP20
          .WORD 0          ;IP21
          .WORD 0          ;IP22
          .WORD 0          ;IP23
          .WORD 0          ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD<DEST
:                       - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD>DEST
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START#=0
: TRANSLATION TABLE DATA = 1 IN LOC 0, 2 IN LOC 1, ETC

```

```

:TOTAL # OF TEST CONDITIONS = 256
:TOTAL # OF TESTS = (1 REG.)256 = 256
:

```

15777
15778
15779 073400 000003
15780 073402 000000
15781 073404 100010
15782 073406 000201
15783 073410 000011
15784 073412 000225
15785 073414 000375
15786 073416 103736
15787 073420 000240
15788 073422 000000
15789 073424 000254
15790 073426 000355
15791 073430 000000
15792 073432 000000
15793 073434 001734
15794 073436 000003
15795 073440 000000
15796 073442 000000
15797 073444 000000
15798 073446 000000
15799 073450 000000
15800 073452 000000

:ENTRY 15 - INSTRUCTION UNDER TEST = MOVTC
:IMOVTC: .WORD 3 :INST=MOVTC
: .WORD 0 :TYPE = 0
: .WORD 100010 :IP1 - SRC.LEN
: .WORD 201 :IP2 - SRC.ADR
: .WORD 000011 :IP3 - DST.LEN
: .WORD 225 :IP4 - DST.ADR
: .WORD 375 :IP5 - FILL
: .WORD T6+2 :IP6 - SRC DATA
: .WORD 240 :IP7 - SRC SURR DATA
: .WORD 0 :IP10 - SRC.SURR.LEN
: .WORD 254 :IP11 - DST DATA
: .WORD 355 :IP12 - DST SURR DATA
: .WORD 0 :IP13 - DST.SURR.LEN
: .WORD 0 :IP14 - SEPARATION CONSTANT
: .WORD IXLTB1 :IP15 - TRANSLATION TABLE
: .WORD 3 :IP16 - SPECIAL HANDLING
: .WORD 0 :IP17
: .WORD 0 :IP20
: .WORD 0 :IP21
: .WORD 0 :IP22
: .WORD 0 :IP23
: .WORD 0 :IP24

15801
15802
15803
15804
15805
15806
15807
15808
15809
15810
15811
15812
15813

:THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:
: SOURCE LENGTH - 100010
: DESTINATION LENGTH - 11
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - 225 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1.
: TRANSLATION TABLE DATA = 1 IN LOC 0, 2 IN LOC 1, ETC
:THIS TEST WAS ADDED TO EXERCISE & TEST THE MOVTC V-BIT OPERATION
:TOTAL # OF TESTS = (1 REG. + 1 INLINE)= 2

15815
 15816
 15817
 15818 073454 000005
 15819 073456 000001
 15820 073460 102752
 15821 073462 103012
 15822 073464 105250
 15823 073466 105274
 15824 073470 103766
 15825 073472 104004
 15826 073474 102436
 15827 073476 000000
 15828 073500 000000
 15829 073502 000000
 15830 073504 000000
 15831 073506 000000
 15832 073510 000000
 15833 073512 000000
 15834 073514 000000
 15835 073516 000000
 15836 073520 000000
 15837 073522 000000
 15838 073524 000000
 15839 073526 000000
 15840
 15841
 15842
 15843
 15844
 15845
 15846
 15847
 15848
 15849
 15850

```

.SBTTL          SKPC TABLES
:ENTRY 16 - INSTRUCTION UNDER TEST = SKPC
:
:SKPC:  .WORD    5          :INST=SKPC
        .WORD    1          :TYPE = 1
        .WORD   TL1C       :IP1 - SRC.LEN
        .WORD    T2        :IP2 - SRC.ADR
        .WORD   T25        :IP3 - CHAR
        .WORD   T26        :IP4 - SRC DATA
        .WORD    T7        :IP5 - SRC SURR DATA
        .WORD   T10        :IP6 - SRC.SURF.LEN
        .WORD   T0        :IP7 - SPECIAL HANDLING
        .WORD    0          :IP10
        .WORD    0          :IP11
        .WORD    0          :IP12
        .WORD    0          :IP13
        .WORD    0          :IP14
        .WORD    0          :IP15
        .WORD    0          :IP16
        .WORD    0          :IP17
        .WORD    0          :IP20
        .WORD    0          :IP21
        .WORD    0          :IP22
        .WORD    0          :IP23
        .WORD    0          :IP24

```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:      SOURCE LENGTH - 0,1,300
:      SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:      SOURCE DATA - CHARACTERS FROM STRING = 001,001,007
:      CHAR - 001,240
:
:TOTAL # OF TEST CONDITIONS = 6
:TOTAL # OF TESTS (1 REG + 1 INLINE)6 = 12
:

```

15852
15853
15854 073530 000005
15855 073532 000001
15856 073534 102462
15857 073536 103030
15858 073540 105266
15859 073542 103756
15860 073544 102436
15861 073546 102436
15862 073550 102706
15863 073552 000000
15864 073554 000000
15865 073556 000000
15866 073560 000000
15867 073562 000000
15868 073564 000000
15869 073566 000000
15870 073570 000000
15871 073572 000000
15872 073574 000000
15873 073576 000000
15874 073600 000000
15875 073602 000000
15876
15877
15878
15879
15880
15881
15882
15883
15884
15885

:ENTRY 17 - INSTRUCTION UNDER TEST = SKPC

```
:SKPC1: .WORD 5          :INST=SKPC
        .WORD 1          :TYPE = 1
        .WORD T1A        :IP1 - SRC.LEN
        .WORD T2AA       :IP2 - SRC.ADR
        .WORD T25A       :IP3 - CHAR
        .WORD T6A        :IP4 - SRC DATA
        .WORD T0         :IP5 - SRC SURR DATA
        .WORD T0         :IP6 - SRC.SURR.LEN
        .WORD TSPA       :IP7 - SPECIAL HANDLING
        .WORD 0          :IP10
        .WORD 0          :IP11
        .WORD 0          :IP12
        .WORD 0          :IP13
        .WORD 0          :IP14
        .WORD 0          :IP15
        .WORD 0          :IP16
        .WORD 0          :IP17
        .WORD 0          :IP20
        .WORD 0          :IP21
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201
: CHAR - 007
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1,START # =0
:
: TOTAL # OF TEST CONDITIONS = 16
: TOTAL # OF TESTS = (1 REG.)16 = 16
:
```

15887
 15888
 15889
 15890 073604 000005
 15891 073606 000000
 15892 073610 100111
 15893 073612 000203
 15894 073614 000003
 15895 073616 103736
 15896 073620 000000
 15897 073622 000000
 15898 073624 000003
 15899 073626 000000
 15900 073630 000000
 15901 073632 000000
 15902 073634 000000
 15903 073636 000000
 15904 073640 000000
 15905 073642 000000
 15906 073644 000000
 15907 073646 000000
 15908 073650 000000
 15909 073652 000000
 15910 073654 000000
 15911 073656 000000

ENTRY 18 - INSTRUCTION UNDER TEST = SKPC

```

:SKPC2: .WORD 5          :INST=SKPC
         .WORD 0          :TYPE = 0
         .WORD 100111     :IP1 - SRC.LEN
         .WORD 203        :IP2 - SRC.ADR
         .WORD 3          :IP3 - CHAR
         .WORD T6+2       :IP4 - SRC DATA
         .WORD 0          :IP5 - SRC SURR DATA
         .WORD 0          :IP6 - SRC.SURR.LEN
         .WORD 3          :IP7 - SPECIAL HANDLING
         .WORD 0          :IP10
         .WORD 0          :IP11
         .WORD 0          :IP12
         .WORD 0          :IP13
         .WORD 0          :IP14
         .WORD 0          :IP15
         .WORD 0          :IP16
         .WORD 0          :IP17
         .WORD 0          :IP20
         .WORD 0          :IP21
         .WORD 0          :IP22
         .WORD 0          :IP23
         .WORD 0          :IP24
  
```

: THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:

```

: SOURCE LENGTH - 100111
: SOURCE ADDRESS - 203
: CHAR - 003
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1.
  
```

```

: THIS TEST WAS ADDED TO EXERCISE & TEST THE SKPC V-BIT OPERATION.
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2
  
```

15912
 15913
 15914
 15915
 15916
 15917
 15918
 15919
 15920
 15921
 15922

15924
15925
15926
15927
15928 073660 000011
15929 073662 000003
15930 073664 102752
15931 073666 103012
15932 073670 102772
15933 073672 103366
15934 073674 103704
15935 073676 105050
15936 073700 102436
15937 073702 104004
15938 073704 105050
15939 073706 104036
15940 073710 104054
15941 073712 104070
15942 073714 102436
15943 073716 000000
15944 073720 000000
15945 073722 000000
15946 073724 000000
15947 073726 000000
15948 073730 000000
15949 073732 000000

.SBTTL MATCHC TABLES

:ENTRY 19 - INSTRUCTION UNDER TEST = MATCHC

```

:IMATC: .WORD 11          ;INST = MATCHC
         .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
         .WORD TL1C      ;IP1 - OBJ.LEN
         .WORD T2        ;IP2 - OBJ.ADR
         .WORD TL2C      ;IP3 - SRC.LEN
         .WORD T4I       ;IP4 - SRC.ADR
         .WORD T5        ;IP5 - UNUSED PORTION OF REGISTER 4
         .WORD T20       ;IP6 - OBJ.DATA
         .WORD T0        ;IP7 - OBJ.SURR.DATA
         .WORD T10       ;IP10 - OBJ.SURR.LEN
         .WORD T20       ;IP11 - SRC.DATA
         .WORD T12       ;IP12 - SRC.SURR.DATA
         .WORD T13       ;IP13 - SRC.SURR.LEN
         .WORD T14       ;IP14 - SEPARATION CONSTANT
         .WORD T0        ;IP15 - SPECIAL HANDLING
         .WORD 0         ;IP16
         .WORD 0         ;IP17
         .WORD 0         ;IP20
         .WORD 0         ;IP21
         .WORD 0         ;IP22
         .WORD 0         ;IP23
         .WORD 0         ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

OBJECT LENGTH - 0,1,300
SOURCE LENGTH - 0,1,5
OBJECT ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
SOURCE ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD < DEST. ADD
                  - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD > DEST. AD
                  - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
                  - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
                  - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
                  - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
                  - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
OBJECT DATA - ALL STRING CHARACTERS IDENTICAL = 127
SOURCE DATA - ALL STRING CHARACTERS IDENTICAL = 127

```

```

:TOTAL # OF TEST CONDITIONS =
:TOTAL # OF TESTS =

```

15950
15951
15952
15953
15954
15955
15956
15957
15958
15959
15960
15961
15962
15963
15964
15965
15966
15967
15968

15970
 15971
 15972
 15973
 15974 073734 000011
 15975 073736 000003
 15976 073740 102462
 15977 073742 103030
 15978 073744 102462
 15979 073746 103366
 15980 073750 102436
 15981 073752 103756
 15982 073754 102436
 15983 073756 102436
 15984 073760 103734
 15985 073762 102436
 15986 073764 102436
 15987 073766 104070
 15988 073770 102706
 15989 073772 000000
 15990 073774 000000
 15991 073776 000000
 15992 074000 000000
 15993 074002 000000
 15994 074004 000000
 15995 074006 000000
 15996
 15997
 15998
 15999
 16000
 16001
 16002
 16003
 16004
 16005
 16006
 16007
 16008
 16009
 16010
 16011
 16012
 16013
 16014

: ENTRY 21 - INSTRUCTION UNDER TEST = MATCHC

```

:IMATC1: .WORD 11          ;INST = MATCHC
          .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1A       ;IP1 - OBJ.LEN
          .WORD T2AA      ;IP2 - OBJ.ADR
          .WORD T1A       ;IP3 - SRC.LEN
          .WORD T4I       ;IP4 - SRC.ADR
          .WORD T0        ;IP5 - 0
          .WORD T6A       ;IP6 - OBJ.DATA
          .WORD T0        ;IP7 - OBJ.SURR.DATA
          .WORD T0        ;IP10 - OBJ.SURR.LEN
          .WORD T6        ;IP11 - SRC.DATA
          .WORD T0        ;IP12 - SRC.SURR.DATA
          .WORD T0        ;IP13 - SRC.SURR.LEN
          .WORD T14       ;IP14 - SEPARATION CONSTANT
          .WORD TSPA      ;IP15 - SPECIAL HANDLING
          .WORD 0         ;IP16
          .WORD 0         ;IP17
          .WORD 0         ;IP20
          .WORD 0         ;IP21
          .WORD 0         ;IP22
          .WORD 0         ;IP23
          .WORD 0         ;IP24
  
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: OBJECT LENGTH - 0,1,2,3,4,5,11,20
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: OBJECT ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE ADDRESS - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD < DEST. ADD
:                   - NO OVERLAP OF SOURCE & DEST. STRINGS;SOURCE ADD > DEST. AD
:                   - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD < DEST ADD
:                   - SOURCE & DEST. STRINGS ADJACENT; SOURCE ADD > DEST ADD
:                   - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD < DE
:                   - SOURCE STRING PARTIALLY OVERLAPS DEST STRING; SRC ADD > DE
:                   - COMPLETE OVERLAP OF SOURCE AND DEST STRINGS; SOURCE ADD =
: OBJECT DATA - INCREMENTING SEQUENCE; INC=1,START # = 0
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1; START # = 1
  
```

```

: TOTAL # OF TEST CONDITIONS = 896
: TOTAL # OF TESTS = (1 REG.)896 = 896
:
:
  
```

16016
16017
16018
16019 074010 000011
16020 074012 000000
16021 074014 000020
16022 074016 000100
16023 074020 100020
16024 074022 000201
16025 074024 000000
16026 074026 103736
16027 074030 000000
16028 074032 000000
16029 074034 103736
16030 074036 000000
16031 074040 000000
16032 074042 000000
16033 074044 000003
16034 074046 000000
16035 074050 000000
16036 074052 000000
16037 074054 000000
16038 074056 000000
16039 074060 000000
16040 074062 000000
16041
16042
16043
16044
16045
16046
16047
16048
16049
16050
16051
16052
16053

...
:ENTRY 22 - INSTRUCTION UNDER TEST = MATCHC
...

IMATC2: .WORD 11 ;INST = MATCHC
.WORD 0 ;TYPE = 0
.WORD 20 ;IP1 - OBJ.LEN
.WORD 100 ;IP2 - OBJ.ADR
.WORD 100020 ;IP3 - SRC.LEN
.WORD 201 ;IP4 - SRC.ADR
.WORD 0 ;IP5 - 0
.WORD T6+2 ;IP6 - OBJ.DATA
.WORD 0 ;IP7 - OBJ.SURR.DATA
.WORD 0 ;IP10 - OBJ.SURR.LEN
.WORD T6+2 ;IP11 - SRC.DATA
.WORD 0 ;IP12 - SRC.SURR.DATA
.WORD 0 ;IP13 - SRC.SURR.LEN
.WORD 0 ;IP14 - SEPARATION CONSTANT
.WORD 3 ;IP15 - SPECIAL HANDLING
.WORD 0 ;IP16
.WORD 0 ;IP17
.WORD 0 ;IP20
.WORD 0 ;IP21
.WORD 0 ;IP22
.WORD 0 ;IP23
.WORD 0 ;IP24

...
:THIS TABLE EXERCISE THE FOLLOWING TEST CONDITION:
...

OBJECT LENGTH - 20
SOURCE LENGTH - 100020
OBJECT ADDRESS - 100 (RELATIVE TO START OF TEST BUFFER)
SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
OBJECT DATA - INCREMENTING SEQUENCE; INC=1, START # = 1
SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # = 1

...
:THIS TEST WAS ADDED TO EXERCISE & TEST THE MATCHC N-BIT OPERATION.
:TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2
...

16055
16056
16057
16058
16059
16060 074064 000006
16061 074066 000003
16062 074070 102752
16063 074072 103012
16064 074074 105312
16065 074076 105326
16066 074100 105402
16067 074102 103734
16068 074104 104036
16069 074106 104054
16070 074110 105442
16071 074112 103766
16072 074114 104004
16073 074116 104070
16074 074120 102436
16075 074122 000000
16076 074124 000000
16077 074126 000000
16078 074130 000000
16079 074132 000000
16080 074134 000000
16081 074136 000000
16082
16083
16084
16085
16086
16087
16088
16089
16090
16091
16092
16093
16094
16095
16096
16097
16098

.SBTTL SCANC TABLES

: ENTRY 23 - INSTRUCTION UNDER TEST = SCANC

```

:ISCAN: .WORD 6           :INST = SCANC
         .WORD 3           :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
         .WORD TL1C        :IP1 - SRC.LEN
         .WORD T2          :IP2 - SRC.ADR
         .WORD T27         :IP3 - TABLE LEN (256 BYTES)
         .WORD T30         :IP4 - MASK
         .WORD T31         :IP5 - TABLE ADR
         .WORD T6          :IP6 - SRC.DATA
         .WORD T12         :IP7 - SRC.SURR DATA
         .WORD T13         :IP10 - SRC.SURR.LEN
         .WORD T32         :IP11 - TABLE DATA
         .WORD T7          :IP12 - TABLE SURR DATA
         .WORD T10         :IP13 - TABLE SURR LEN
         .WORD T14         :IP14 - SEPARATION CONSTANT
         .WORD T0          :IP15 - SPECIAL HANDLING
         .WORD 0           :IP16
         .WORD 0           :IP17
         .WORD 0           :IP20
         .WORD 0           :IP21
         .WORD 0           :IP22
         .WORD 0           :IP23
         .WORD 0           :IP24

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE LENGTH - 0,1,300
: SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: MASK - 0,1,377
: TABLE ADDRESS - NO OVERLAP WITH SOURCE STRING
:                   - OVERLAP - TABLE ADDRESS=SOURCE ADDRESS
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # =1
: TABLE DATA - ALL BYTES IDENTICAL = 0
:                 - ALL BYTES IDENTICAL = 377
:                 - INCREMENTING SEQUENCE; INC=1, START # =1

```

```

: TOTAL # OF TEST CONDITIONS = 54
: TOTAL # OF TESTS = (1 REG. + 1 INLINE)54=108

```

16100
16101
16102
16103 074140 000006
16104 074142 000003
16105 074144 102462
16106 074146 103030
16107 074150 105312
16108 074152 105346
16109 074154 105354
16110 074156 103756
16111 074160 102436
16112 074162 102436
16113 074164 103756
16114 074166 102436
16115 074170 102436
16116 074172 104070
16117 074174 102706
16118 074176 000000
16119 074200 000000
16120 074202 000000
16121 074204 000000
16122 074206 000000
16123 074210 000000
16124 074212 000000

```

:
: ENTRY 24 - INSTRUCTION UNDER TEST = SCANC
:
: ISCAN1: .WORD 6           ;INST = SCANC
:          .WORD 3           ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:          .WORD T1A        ;IP1 - SRC.LEN
:          .WORD T2AA       ;IP2 - SRC.ADR
:          .WORD T27        ;IP3 - TABLE LEN (256 BYTES)
:          .WORD T30A       ;IP4 - MASK
:          .WORD T31A       ;IP5 - TABLE ADR
:          .WORD T6A        ;IP6 - SRC.DATA
:          .WORD T0         ;IP7 - SRC.SURR DATA
:          .WORD T0         ;IP10 - SRC.SURR.LEN
:          .WORD T6A        ;IP11 - TABLE DATA
:          .WORD T0         ;IP12 - TABLE SURR DATA
:          .WORD T0         ;IP13 - TABLE SURR LEN
:          .WORD T14        ;IP14 - SEPARATION CONSTANT
:          .WORD TSPA       ;IP15 - SPECIAL HANDLING
:          .WORD 0          ;IP16
:          .WORD 0          ;IP17
:          .WORD 0          ;IP20
:          .WORD 0          ;IP21
:          .WORD 0          ;IP22
:          .WORD 0          ;IP23
:          .WORD 0          ;IP24

```

16125
16126
16127
16128
16129
16130
16131
16132
16133
16134
16135
16136

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: MASK - 252
: TABLE ADDRESS - NO OVERLAP WITH SOURCE STRING
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START #=0
: TABLE DATA - INCREMENTING SEQUENCE; INC=1, START #=0
:
: TOTAL # OF TEST CONDITIONS = 16
: TOTAL # OF TESTS = (1 REG.) = 16
:

```


16138
16139
16140
16141 074214 000006
16142 074216 000000
16143 074220 100040
16144 074222 000110
16145 074224 000256
16146 074226 000377
16147 074230 000200
16148 074232 103736
16149 074234 000000
16150 074236 000000
16151 074240 105450
16152 074242 000000
16153 074244 000000
16154 074246 000000
16155 074250 000003
16156 074252 000000
16157 074254 000000
16158 074256 000000
16159 074260 000000
16160 074262 000000
16161 074264 000000
16162 074266 000000
16163
16164
16165
16166
16167
16168
16169
16170
16171
16172
16173
16174
16175

: ENTRY 25 - INSTRUCTION UNDER TEST = SCANC
: SCANC2: .WORD 6 : INST = SCANC
: .WORD 0 : TYPE = 0
: .WORD 100040 : IP1 - SRC.LEN
: .WORD 110 : IP2 - SRC.ADR
: .WORD 256 : IP3 - TABLE LEN (256 BYTES)
: .WORD 377 : IP4 - MASK
: .WORD 200 : IP5 - TABLE ADR
: .WORD T6+2 : IP6 - SRC.DATA
: .WORD 0 : IP7 - SRC.SURR DATA
: .WORD 0 : IP10 - SRC.SURR.LEN
: .WORD T32+6 : IP11 - TABLE DATA
: .WORD 0 : IP12 - TABLE SURR DATA
: .WORD 0 : IP13 - TABLE SURR LEN
: .WORD 0 : IP14 - SEPARATION CONSTANT
: .WORD 3 : IP15 - SPECIAL HANDLING
: .WORD 0 : IP16
: .WORD 0 : IP17
: .WORD 0 : IP20
: .WORD 0 : IP21
: .WORD 0 : IP22
: .WORD 0 : IP23
: .WORD 0 : IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
: SOURCE LENGTH - 100040
: SOURCE ADDRESS - 10
: MARK - 377
: TABLE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START#=1
: TABLE DATA - ALL BYTES IDENTICAL = 377

: THIS TEST WAS ADDED TO EXERCISE & TEST THE SCANC N-BIT OPERATION.
: TOTAL # OF TESTS = (1 REG. + 1 INLINE)=2

16177
16178
16179
16180
16181
16182 074270 000007
16183 074272 000001
16184 074274 102752
16185 074276 103012
16186 074300 105312
16187 074302 105326
16188 074304 105402
16189 074306 103734
16190 074310 104036
16191 074312 104054
16192 074314 105442
16193 074316 103766
16194 074320 104004
16195 074322 104070
16196 074324 102436
16197 074326 000000
16198 074330 000000
16199 074332 000000
16200 074334 000000
16201 074336 000000
16202 074340 000000
16203 074342 000000
16204
16205
16206
16207
16208
16209
16210
16211
16212
16213
16214
16215
16216
16217
16218
16219
16220

.SBTTL SPANC TABLES

: ENTRY 26 - INSTRUCTION UNDER TEST = SPANC

| | | |
|---------|------------|-------------------------------|
| : SPAN: | .WORD 7 | : INST = SPANC |
| | .WORD 1 | : TYPE = 1 |
| | .WORD TL1C | : IP1 - SRC.LEN |
| | .WORD T2 | : IP2 - SRC.ADR |
| | .WORD T27 | : IP3 - TABLE LEN (256 BYTES) |
| | .WORD T30 | : IP4 - MASK |
| | .WORD T31 | : IP5 - TABLE ADR |
| | .WORD T6 | : IP6 - SRC.DATA |
| | .WORD T12 | : IP7 - SRC.SURR DATA |
| | .WORD T13 | : IP10 - SRC.SURR.LEN |
| | .WORD T32 | : IP11 - TABLE DATA |
| | .WORD T7 | : IP12 - TABLE SURR DATA |
| | .WORD T10 | : IP13 - TABLE SURR LEN |
| | .WORD T14 | : IP14 - SEPARATION CONSTANT |
| | .WORD T0 | : IP15 - SPECIAL HANDLING |
| | .WORD 0 | : IP16 |
| | .WORD 0 | : IP17 |
| | .WORD 0 | : IP20 |
| | .WORD 0 | : IP21 |
| | .WORD 0 | : IP22 |
| | .WORD 0 | : IP23 |
| | .WORD 0 | : IP24 |

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE LENGTH - 0,1,300
: SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: MASK - 0,1,377
: TABLE ADDRESS - NO OVERLAP WITH SOURCE STRING
: - OVERLAP - TABLE ADDRESS=SOURCE ADDRESS
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START # =1
: TABLE DATA - ALL BYTES IDENTICAL = 0
: - ALL BYTES IDENTICAL = 377
: - INCREMENTING SEQUENCE; INC=1, START # =1

: TOTAL # OF TEST CONDITIONS = 54
: TOTAL # OF TESTS = (1 REG. + 1 INLINE)54=108

16222
16223
16224
16225 074344 000007
16226 074346 000001
16227 074350 102462
16228 074352 103030
16229 074354 105312
16230 074356 105346
16231 074360 105354
16232 074362 103756
16233 074364 102436
16234 074366 102436
16235 074370 103756
16236 074372 102436
16237 074374 102436
16238 074376 104070
16239 074400 102706
16240 074402 000000
16241 074404 000000
16242 074406 000000
16243 074410 000000
16244 074412 000000
16245 074414 000000
16246 074416 000000
16247
16248
16249
16250
16251
16252
16253
16254
16255
16256
16257
16258

: ENTRY 27 - INSTRUCTION UNDER TEST = SPANC

: SPAN1: .WORD 7 ; INST = SPANC
: .WORD 1 ; TYPE = 1
: .WORD T1A ; IP1 - SRC.LEN
: .WORD T2AA ; IP2 - SRC.ADR
: .WORD T27 ; IP3 - TABLE LEN (256 BYTES)
: .WORD T30A ; IP4 - MASK
: .WORD T31A ; IP5 - TABLE ADR
: .WORD T6A ; IP6 - SRC.DATA
: .WORD T0 ; IP7 - SRC.SURR DATA
: .WORD T0 ; IP10 - SRC.SURR.LEN
: .WORD T6A ; IP11 - TABLE DATA
: .WORD T0 ; IP12 - TABLE SURR DATA
: .WORD T0 ; IP13 - TABLE SURR LEN
: .WORD T14 ; IP14 - SEPARATION CONSTANT
: .WORD TSPA ; IP15 - SPECIAL HANDLING
: .WORD 0 ; IP16
: .WORD 0 ; IP17
: .WORD 0 ; IP20
: .WORD C ; IP21
: .WORD 0 ; IP22
: .WORD 0 ; IP23
: .WORD 0 ; IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: MASK - 252
: TABLE ADDRESS - NO OVERLAP WITH SOURCE STRING
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START#=0
: TABLE DATA - INCREMENTING SEQUENCE; INC=1, START#=0

: TOTAL # OF TEST CONDITIONS = 16
: TOTAL # OF TESTS = (1 REG.) = 16

16260
 16261
 16262
 16263 074420 000007
 16264 074422 000000
 16265 074424 100040
 16266 074426 000110
 16267 074430 000256
 16268 074432 000000
 16269 074434 000200
 16270 074436 103736
 16271 074440 000000
 16272 074442 000000
 16273 074444 105450
 16274 074446 000000
 16275 074450 000000
 16276 074452 000000
 16277 074454 000003
 16278 074456 000000
 16279 074460 000000
 16280 074462 000000
 16281 074464 000000
 16282 074466 000000
 16283 074470 000000
 16284 074472 000000
 16285
 16286
 16287
 16288
 16289
 16290
 16291
 16292
 16293
 16294
 16295
 16296
 16297

:
: ENTRY 28 - INSTRUCTION UNDER TEST = SPANC
:

```

: SPAN2: .WORD 7           ; INST = SPANC
          .WORD 0           ; TYPE = 0
          .WORD 100040      ; IP1 - SRC.LEN
          .WORD 110        ; IP2 - SRC.ADR
          .WORD 256        ; IP3 - TABLE LEN (256 BYTES)
          .WORD 0           ; IP4 - MASK
          .WORD 200        ; IP5 - TABLE ADR
          .WORD T6+2       ; IP6 - SRC.DATA
          .WORD 0           ; IP7 - SRC.SURR DATA
          .WORD 0           ; IP10 - SRC.SURR.LEN
          .WORD T32+6      ; IP11 - TABLE DATA
          .WORD 0           ; IP12 - TABLE SURR DATA
          .WORD 0           ; IP13 - TABLE SURR LEN
          .WORD 0           ; IP14 - SEPARATION CONSTANT
          .WORD 3           ; IP15 - SPECIAL HANDLING
          .WORD 0           ; IP16
          .WORD 0           ; IP17
          .WORD 0           ; IP20
          .WORD 0           ; IP21
          .WORD 0           ; IP22
          .WORD 0           ; IP23
          .WORD 0           ; IP24

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE LENGTH - 100040
: SOURCE ADDRESS - 10
: MARK - 377
: TABLE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - INCREMENTING SEQUENCE; INC=1, START#=1
: TABLE DATA - ALL BYTES IDENTICAL = 377

```

```

: THIS TEST WAS ADDED TO EXERCISE & TEST THE SPANC N-BIT OPERATION.
: TOTAL # OF TESTS = (1 REG. + 1 INLINE)=2
:

```

16299
 16300
 16301
 16302
 16303
 16304 074474 000016
 16305 074476 000001
 16306 074500 105526
 16307 074502 103012
 16308 074504 105546
 16309 074506 105634
 16310 074510 105712
 16311 074512 103766
 16312 074514 104004
 16313 074516 104020
 16314 074520 104036
 16315 074522 104054
 16316 074524 104070
 16317 074526 102436
 16318 074530 000000
 16319 074532 000000
 16320 074534 000000
 16321 074536 000000
 16322 074540 000000
 16323 074542 000000
 16324 074544 000000
 16325 074546 000000
 16326
 16327
 16328
 16329
 16330
 16331
 16332
 16333
 16334
 16335
 16336
 16337
 16338
 16339
 16340

```

.SBTTL          CVTPN TABLES
:
:ENTRY 29 - INSTRUCTION UNDER TEST = CVTPN
:
:ICPZ:          .WORD 16          ;INST=CVTPN
                .WORD 1          ;TYPE = 1
                .WORD T331       ;IP1 - SRC.LEN
                .WORD T2         ;IP2 - SRC.ADR
                .WORD T332       ;IP3 - DST.LEN
                .WORD T34        ;IP4 - DST.ADR
                .WORD T35        ;IP5 - SRC DATA
                .WORD T7         ;IP6 - SRC SURR DATA
                .WORD T10        ;IP7 - SRC SURR LEN
                .WORD T11        ;IP10 - DST DATA
                .WORD T12        ;IP11 - DST SURR DATA
                .WORD T13        ;IP12 - DST SURR LEN
                .WORD T14        ;IP13 - SEPARATION CONSTANT
                .WORD T0         ;IP14 - SPECIAL HANDLING
                .WORD 0          ;IP15
                .WORD 0          ;IP16
                .WORD 0          ;IP17
                .WORD 0          ;IP20
                .WORD 0          ;IP21
                .WORD 0          ;IP22
                .WORD 0          ;IP23
                .WORD 0          ;IP24

```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE LENGTH - 0,1,37
:DESTINATION LENGTH - 0,1,37
:SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
:DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
:                      - SOURCE & DESTINATION STRINGS ADJACENT
:SOURCE DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:              - ALL DIGITS IDENTICAL = 8; SIGN -
:              - ALL DIGITS IDENTICAL = 0; SIGN -
:
:TOTAL # OF TEST CONDITIONS = 54
:TOTAL # OF TESTS = (2 DATA TYPES + 1 IN LINE)54=162
:

```

16342
 16343
 16344 074550 000016
 16345 074552 000001
 16346 074554 102462
 16347 074556 103022
 16348 074560 102462
 16349 074562 105602
 16350 074564 106000
 16351 074566 102436
 16352 074570 102436
 16353 074572 102436
 16354 074574 102436
 16355 074576 102436
 16356 074600 104070
 16357 074602 102706
 16358 074604 000000
 16359 074606 000000
 16360 074610 000000
 16361 074612 000000
 16362 074614 000000
 16363 074616 000000
 16364 074620 000000
 16365 074622 000000
 16366
 16367
 16368
 16369
 16370
 16371
 16372
 16373
 16374
 16375
 16376
 16377
 16378

```

:ENTRY 30 - INSTRUCTION UNDER TEST = CVTPN
:
:ICPZ1: .WORD 16          :INST=CVTPN
        .WORD 1          :TYPE = 1
        .WORD T1A        :IP1 - SRC.LEN
        .WORD T2A        :IP2 - SRC.ADR
        .WORD T1A        :IP3 - DST.LEN
        .WORD T34A       :IP4 - DST.ADR
        .WORD TP19       :IP5 - SRC DATA
        .WORD T0         :IP6 - SRC SURR DATA
        .WORD T0         :IP7 - SRC SURR LEN
        .WORD T0         :IP10 - DST DATA
        .WORD T0         :IP11 - DST SURR DATA
        .WORD T0         :IP12 - DST SURR LEN
        .WORD T14        :IP13 - SEPARATION CONSTANT
        .WORD TSFA       :IP14 - SPECIAL HANDLING
        .WORD 0          :IP15
        .WORD 0          :IP16
        .WORD 0          :IP17
        .WORD 0          :IP20
        .WORD 0          :IP21
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
  
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
: SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 64
: TOTAL # OF TESTS = (1 REG.)64 = 64
:
:

```

16380
16381
16382
16383 074624 000017
16384 074626 000003
16385 074630 105526
16386 074632 103012
16387 074634 105546
16388 074636 106170
16389 074640 106252
16390 074642 103766
16391 074644 104004
16392 074646 104020
16393 074650 104036
16394 074652 104054
16395 074654 104070
16396 074656 102436
16397 074660 000000
16398 074662 000000
16399 074664 000000
16400 074666 000000
16401 074670 000000
16402 074672 000000
16403 074674 000000
16404 074676 000000
16405
16406
16407
16408
16409
16410
16411
16412
16413
16414
16415
16416
16417
16418
16419

```
.SBTTL          CVTNP TABLES
:ENTRY 31 - INSTRUCTION UNDER TEST = CVTNP
:
:ICZP:          .WORD 17          ;INST=CVTNP
                .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
                .WORD T331       ;IP1 - SRC.LEN
                .WORD T2         ;IP2 - SRC.ADR
                .WORD T332       ;IP3 - DST.LEN
                .WORD T36        ;IP4 - DST.ADR
                .WORD T37        ;IP5 - SRC DATA
                .WORD T7         ;IP6 - SRC SURR DATA
                .WORD T10        ;IP7 - SRC SURR LEN
                .WORD T11        ;IP10 - DST DATA
                .WORD T12        ;IP11 - DST SURR DATA
                .WORD T13        ;IP12 - DST SURR LEN
                .WORD T14        ;IP13 - SEPARATION CONSTANT
                .WORD T0         ;IP14 - SPECIAL HANDLING
                .WORD 0          ;IP15
                .WORD 0          ;IP16
                .WORD 0          ;IP17
                .WORD 0          ;IP20
                .WORD 0          ;IP21
                .WORD 0          ;IP22
                .WORD 0          ;IP23
                .WORD 0          ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
:   SOURCE LENGTH - 0,1,37
:   DESTINATION LENGTH - 0,1,37
:   SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:   DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
:                       - SOURCE & DESTINATION STRINGS ADJACENT
:   SOURCE DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:                 - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:                 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
```

```
:TOTAL # OF TEST CONDITIONS = 54
:TOTAL # OF TESTS = (6 DATA TYPES + 1 IN LINE)54 = 378
:
```

16421
16422
16423 074700 000017
16424 074702 000001
16425 074704 102462
16426 074706 103022
16427 074710 102462
16428 074712 106142
16429 074714 106340
16430 074716 102436
16431 074720 102436
16432 074722 102436
16433 074724 102436
16434 074726 102436
16435 074730 104070
16436 074732 102706
16437 074734 000000
16438 074736 000000
16439 074740 000000
16440 074742 000000
16441 074744 000000
16442 074746 000000
16443 074750 000000
16444 074752 000000
16445
16446
16447
16448
16449
16450
16451
16452
16453
16454
16455
16456

:ENTRY 32 - INSTRUCTION UNDER TEST = CVTNP

```
:ICZP1: .WORD 17          ;INST=CVTNP
         .WORD 1          ;TYPE = 1
         .WORD T1A       ;IP1 - SRC.LEN
         .WORD T2A       ;IP2 - SRC.ADR
         .WORD T1A       ;IP3 - DST.LEN
         .WORD T36A      ;IP4 - DST.ADR
         .WORD T219      ;IP5 - SRC DATA
         .WORD T0        ;IP6 - SRC SURR DATA
         .WORD T0        ;IP7 - SRC SURR LEN
         .WORD T0        ;IP10 - DST DATA
         .WORD T0        ;IP11 - DST SURR DATA
         .WORD T0        ;IP12 - DST SURR LEN
         .WORD T14       ;IP13 - SEPARATION CONSTANT
         .WORD TSPA      ;IP14 - SPECIAL HANDLING
         .WORD 0         ;IP15
         .WORD 0         ;IP16
         .WORD 0         ;IP17
         .WORD 0         ;IP20
         .WORD 0         ;IP21
         .WORD 0         ;IP22
         .WORD 0         ;IP23
         .WORD 0         ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
: SOURCE DATA - DIGITS FROM STRING= 1234567891234567891234000891233; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 64
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)64 = 64
:
```


16458
16459
16460 074754 000017
16461 074756 000001
16462 074760 102506
16463 074762 103030
16464 074764 102506
16465 074766 106142
16466 074770 106340
16467 074772 102436
16468 074774 102436
16469 074776 102436
16470 075000 102436
16471 075002 102436
16472 075004 104070
16473 075006 102436
16474 075010 000000
16475 075012 000000
16476 075014 000000
16477 075016 000000
16478 075020 000000
16479 075022 000000
16480 075024 000000
16481 075026 000000
16482
16483
16484
16485
16486
16487
16488
16489
16490
16491
16492

:ENTRY 32A - INSTRUCTION UNDER TEST = CVTNP

:ICZP2: .WORD 17 ;INST=CVTNP
 .WORD 1 ;TYPE = 1
 .WORD T1B ;IP1 - SRC.LEN
 .WORD T2AA ;IP2 - SRC.ADR
 .WORD T1B ;IP3 - DST.LEN
 .WORD T36A ;IP4 - DST.ADR
 .WORD TZ19 ;IP5 - SRC DATA
 .WORD T0 ;IP6 - SRC SURR DATA
 .WORD T0 ;IP7 - SRC SURR LEN
 .WORD T0 ;IP10 - DST DATA
 .WORD T0 ;IP11 - DST SURR DATA
 .WORD T0 ;IP12 - DST SURR LEN
 .WORD T14 ;IP13 - SEPARATION CONSTANT
 .WORD T0 ;IP14 - SPECIAL HANDLING
 .WORD 0 ;IP15
 .WORD 0 ;IP16
 .WORD 0 ;IP17
 .WORD 0 ;IP20
 .WORD 0 ;IP21
 .WORD 0 ;IP22
 .WORD 0 ;IP23
 .WORD 0 ;IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE LENGTH - 1,2,3
: DESTINATION LENGTH - 1,2,3
: SOURCE ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
: SOURCE DATA - DIGITS FROM STRING= 1234567891234567891234000891233; SIGN +
: TOTAL # OF TEST CONDITIONS = 18
: TOTAL # OF TESTS = (6 DATA TYPES + 1 IN LINE)18 = 126

16494
16495
16496
16497 075030 000031
16498 075032 000003
16499 075034 106416
16500 075036 106476
16501 075040 105472
16502 075042 103012
16503 075044 104020
16504 075046 104036
16505 075050 104054
16506 075052 102436
16507 075054 000000
16508 075056 000000
16509 075060 000000
16510 075062 000000
16511 075064 000000
16512 075066 000000
16513 075070 000000
16514 075072 000000
16515 075074 000000
16516 075076 000000
16517 075100 000000
16518 075102 000000
16519
16520
16521
16522
16523
16524
16525
16526
16527
16528
16529
16530

```
.SBTTL          CVTLP TABLES
:ENTRY 33 - INSTRUCTION UNDER TEST = CVTLP
:
:ICLP:  .WORD 31          :INST=CVTLP
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T40       :IP1 - SRC.HIGH (R2)
        .WORD T41       :IP2 - SRC.LOW (R3)
        .WORD T33       :IP3 - DST.LEN (R4)
        .WORD T2        :IP4 - DST.ADR (R5)
        .WORD T11       :IP5 - DST DATA
        .WORD T12       :IP6 - DST SURR DATA
        .WORD T13       :IP7 - DST SURR LEN
        .WORD T0        :IP10 - SPECIAL HANDLING
        .WORD 0         :IP11
        .WORD 0         :IP12
        .WORD 0         :IP13
        .WORD 0         :IP14
        .WORD 0         :IP15
        .WORD 0         :IP16
        .WORD 0         :IP17
        .WORD 0         :IP20
        .WORD 0         :IP21
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
:   DESTINATION LENGTH - 0,1,20
:   DESTINATION ADDRESS - 200 (REATIVE TO START OF BUFFER)
:   SOURCE DATA HIGH - 0+,0-,77777+,77777-,5+
:   SOURCE DATA LOW - 0+,4+,77777-
```

```
: TOTAL # OF TEST CONDITIONS = 45
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)45 = 135
:
```

16532
16533
16534 075104 000031
16535 075106 000003
16536 075110 106444
16537 075112 106462
16538 075114 102462
16539 075116 103022
16540 075120 102436
16541 075122 102436
16542 075124 102436
16543 075126 102706
16544 075130 000000
16545 075132 000000
16546 075134 000000
16547 075136 000000
16548 075140 000000
16549 075142 000000
16550 075144 000000
16551 075146 000000
16552 075150 000000
16553 075152 000000
16554 075154 000000
16555 075156 000000
16556
16557
16558
16559
16560
16561
16562
16563
16564
16565
16566

:ENTRY 34 - INSTRUCTION UNDER TEST = CVTLP

:ICLP1: .WORD 31 :INST=CVTLP
:WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:WORD T40A :IP1 - SRC.HIGH (R2)
:WORD T41A :IP2 - SRC.LOW (R3)
:WORD T1A :IP3 - DST.LEN (R4)
:WORD T2A :IP4 - DST.ADR (R5)
:WORD T0 :IP5 - DST DATA
:WORD T0 :IP6 - DST SURR DATA
:WORD T0 :IP7 - DST SURR LEN
:WORD TSPA :IP10 - SPECIAL HANDLING
:WORD 0 :IP11
:WORD 0 :IP12
:WORD 0 :IP13
:WORD 0 :IP14
:WORD 0 :IP15
:WORD 0 :IP16
:WORD 0 :IP17
:WORD 0 :IP20
:WORD 0 :IP21
:WORD 0 :IP22
:WORD 0 :IP23
:WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION ADDRESS - 201
: SOURCE DATA HIGH - 0+
: SOURCE DATA LOW - 77777-
:

:TOTAL # OF TEST CONDITIONS = 8
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)8 = 8
:

16568
16569
16570 075160 000031
16571 075162 000003
16572 075164 106450
16573 075166 106516
16574 075170 104070
16575 075172 103012
16576 075174 102436
16577 075176 102436
16578 075200 102436
16579 075202 102436
16580 075204 000000
16581 075206 000000
16582 075210 000000
16583 075212 000000
16584 075214 000000
16585 075216 000000
16586 075220 000000
16587 075222 000000
16588 075224 000000
16589 075226 000000
16590 075230 000000
16591 075232 000000
16592
16593
16594
16595
16596
16597
16598
16599
16600
16601
16602

:ENTRY 34A - INSTRUCTION UNDER TEST = CVTLP

:ICLP2: .WORD 31 :INST=CVTLP
 .WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
 .WORD T40B :IP1 - SRC.HIGH (R2)
 .WORD T41B :IP2 - SRC.LOW (R3)
 .WORD T14 :IP3 - DST.LEN (R4)
 .WORD T2 :IP4 - DST.ADR (R5)
 .WORD T0 :IP5 - DST DATA
 .WORD T0 :IP6 - DST SURR DATA
 .WORD T0 :IP7 - DST SURR LEN
 .WORD T0 :IP10 - SPECIAL HANDLING
 .WORD 0 :IP11
 .WORD 0 :IP12
 .WORD 0 :IP13
 .WORD 0 :IP14
 .WORD 0 :IP15
 .WORD 0 :IP16
 .WORD 0 :IP17
 .WORD 0 :IP20
 .WORD 0 :IP21
 .WORD 0 :IP22
 .WORD 0 :IP23
 .WORD 0 :IP24

:THIS TABLE EXERCISES THE FOLLOWING TEST CONDITION

:
: DESTINATION LENGTH - 10
: DESTINATION ADDRESS - 200
: SOURCE DATA HIGH - 0,231,252
: SOURCE DATA LOW - 120360,0,125

:TOTAL # OF TEST CONDITIONS = 9
:TOTAL # OF TESTS = (2 DATA TYPES + 1 IN-LINE)9 = 27
:

16604
16605
16606
16607 075234 000021
16608 075236 000001
16609 075240 106416
16610 075242 106476
16611 075244 105472
16612 075246 103012
16613 075250 104020
16614 075252 104036
16615 075254 104054
16616 075256 102436
16617 075260 000000
16618 075262 000000
16619 075264 000000
16620 075266 000000
16621 075270 000000
16622 075272 000000
16623 075274 000000
16624 075276 000000
16625 075300 000000
16626 075302 000000
16627 075304 000000
16628 075306 000000
16629
16630
16631
16632
16633
16634
16635
16636
16637
16638
16639

.SBTTL CVTLN TABLES
:ENTRY 35 - INSTRUCTION UNDER TEST = CVTLN
:ICLZ: .WORD 21 ;INST=CVTLN
 .WORD 1 ;TYPE = 1
 .WORD T40 ;IP1 - SRC.HIGH (R2)
 .WORD T41 ;IP2 - SRC.LOW (R3)
 .WORD T33 ;IP3 - DST.LEN (R4)
 .WORD T2 ;IP4 - DST.ADR (R5)
 .WORD T11 ;IP5 - DST DATA
 .WORD T12 ;IP6 - DST SURR DATA
 .WORD T13 ;IP7 - DST SURR LEN
 .WORD T0 ;IP10 - SPECIAL HANDLING
 .WORD 0 ;IP11
 .WORD 0 ;IP12
 .WORD 0 ;IP13
 .WORD 0 ;IP14
 .WORD 0 ;IP15
 .WORD 0 ;IP16
 .WORD 0 ;IP17
 .WORD 0 ;IP20
 .WORD 0 ;IP21
 .WORD 0 ;IP22
 .WORD 0 ;IP23
 .WORD 0 ;IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
: DESTINATION LENGTH - 0,1,20
: DESTINATION ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA HIGH - 0+,0-,77777+,77777-,5+
: SOURCE DATA LOW - 0+,4+,77777-
: TOTAL # OF TEST CONDITIONS = 45
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)45 = 315
:

16641
16642
16643 075310 000021
16644 075312 000001
16645 075314 106466
16646 075316 106472
16647 075320 102462
16648 075322 103022
16649 075324 102436
16650 075326 102436
16651 075330 102436
16652 075332 102706
16653 075334 000000
16654 075336 000000
16655 075340 000000
16656 075342 000000
16657 075344 000000
16658 075346 000000
16659 075350 000000
16660 075352 000000
16661 075354 000000
16662 075356 000000
16663 075360 000000
16664 075362 000000
16665
16666
16667
16668
16669
16670
16671
16672
16673
16674
16675

:ENTRY 36 - INSTRUCTION UNDER TEST = CVTLN

:ICLZ1: .WORD 21 :INST=CVTLN
:WORD 1 :TYPE = 1
:WORD T40AA :IP1 - SRC.HIGH (R2)
:WORD T41AA :IP2 - SRC.LOW (R3)
:WORD T1A :IP3 - DST.LEN (R4)
:WORD T2A :IP4 - DST.ADR (R5)
:WORD T0 :IP5 - DST DATA
:WORD T0 :IP6 - DST SURR DATA
:WORD T0 :IP7 - DST SURR LEN
:WORD TSPA :IP10 - SPECIAL HANDLING
:WORD 0 :IP11
:WORD 0 :IP12
:WORD 0 :IP13
:WORD 0 :IP14
:WORD 0 :IP15
:WORD 0 :IP16
:WORD 0 :IP17
:WORD 0 :IP20
:WORD 0 :IP21
:WORD 0 :IP22
:WORD 0 :IP23
:WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA HIGH - 5+
: SOURCE DATA LOW - 4+
:
:TOTAL # OF TEST CONDITIONS = 8
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)8 = 8
:

16677
16678
16679
16680 075364 000025
16681 075366 000003
16682 075370 105472
16683 075372 103012
16684 075374 105712
16685 075376 103766
16686 075400 104004
16687 075402 103704
16688 075404 102436
16689 075406 000000
16690 075410 000000
16691 075412 000000
16692 075414 000000
16693 075416 000000
16694 075420 000000
16695 075422 000000
16696 075424 000000
16697 075426 000000
16698 075430 000000
16699 075432 000000
16700 075434 000000
16701 075436 000000
16702
16703
16704
16705
16706
16707
16708
16709
16710
16711
16712
16713

```
.SBTTL          CVTPL TABLES
:ENTRY 37 - INSTRUCTION UNDER TEST = CVTPL
:
:ICPL:          .WORD 25          ;INST = CVTPL
                .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
                .WORD T33       ;IP1 - SRC.LEN (R0)
                .WORD T2        ;IP2 - SRC.ADR (R1)
                .WORD T35       ;IP3 - SRC DATA
                .WORD T7        ;IP4 - SRC SURR DATA
                .WORD T10       ;IP5 - SRC SURR LEN
                .WORD T5        ;IP6- UNUSED PORTION OF REGISTER 4
                .WORD T0        ;IP7 - SPECIAL HANDLING
                .WORD 0         ;IP10
                .WORD 0         ;IP11
                .WORD 0         ;IP12
                .WORD 0         ;IP13
                .WORD 0         ;IP14
                .WORD 0         ;IP15
                .WORD 0         ;IP16
                .WORD 0         ;IP17
                .WORD 0         ;IP20
                .WORD 0         ;IP21
                .WORD 0         ;IP22
                .WORD 0         ;IP23
                .WORD 0         ;IP24
```

```
:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:   SOURCE LENGTH - 0,1,20
:   SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:   SOURCE DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:                 - ALL DIGITS IDENTICAL = 8; SIGN -
:                 - ALL DIGITS IDENTICAL = 0; SIGN -
:
:TOTAL # OF TEST CONDITIONS = 9
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)9 = 27
:
```

16715
 16716
 16717
 16718
 16719 075440 000025
 16720 075442 000003
 16721 075444 105512
 16722 075446 103012
 16723 075450 107400
 16724 075452 103766
 16725 075454 104004
 16726 075456 103704
 16727 075460 102436
 16728 075462 000000
 16729 075464 000000
 16730 075466 000000
 16731 075470 000000
 16732 075472 000000
 16733 075474 000000
 16734 075476 000000
 16735 075500 000000
 16736 075502 000000
 16737 075504 000000
 16738 075506 000000
 16739 075510 000000
 16740 075512 000000
 16741
 16742
 16743
 16744
 16745
 16746
 16747
 16748
 16749
 16750
 16751
 16752
 16753
 16754
 16755

: ENTRY 40 - INSTRUCTION UNDER TEST = CVTPL

```

:ICPL1: .WORD 25          ;INST = CVTPL
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T33A      ;IP1 - SRC.LEN (R0)
        .WORD T2        ;IP2 - SRC.ADR (R1)
        .WORD T51       ;IP3 - SRC DATA
        .WORD T7        ;IP4 - SRC SURR DATA
        .WORD T10       ;IP5 - SRC SURR LEN
        .WORD T5        ;IP6- UNUSED PORTION OF REGISTER 4
        .WORD T0        ;IP7 - SPECIAL HANDLING
        .WORD 0         ;IP10
        .WORD 0         ;IP11
        .WORD 0         ;IP12
        .WORD 0         ;IP13
        .WORD 0         ;IP14
        .WORD 0         ;IP15
        .WORD 0         ;IP16
        .WORD 0         ;IP17
        .WORD 0         ;IP20
        .WORD 0         ;IP21
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
  
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE LENGTH - 12
: SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - DIGITS FROM STRING = 2,147,483,648+
:               - DIGITS FROM STRING = 2,147,483,647+
:               - DIGITS FROM STRING = 2,147,483,648-
:               - DIGITS FROM STRING = 2,147,483,649-
:               - DIGITS FROM STRING = 4,294,967,294+
:               - DIGITS FROM STRING = 42,949,672,940+
  
```

```

: TOTAL # OF TEST CONDITIONS = 6
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)6 = 18
  
```


16757
16758
16759 075514 000025
16760 075516 000003
16761 075520 102462
16762 075522 103022
16763 075524 106000
16764 075526 102436
16765 075530 102436
16766 075532 103704
16767 075534 102706
16768 075536 000000
16769 075540 000000
16770 075542 000000
16771 075544 000000
16772 075546 000000
16773 075550 000000
16774 075552 000000
16775 075554 000000
16776 075556 000000
16777 075560 000000
16778 075562 000000
16779 075564 000000
16780 075566 000000
16781
16782
16783
16784
16785
16786
16787
16788
16789
16790

:ENTRY 41 - INSTRUCTION UNDER TEST = CVTPL

:ICPL2: .WORD 25 :INST = CVTPL
:WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:WORD T1A :IP1 - SRC.LEN (R0)
:WORD T2A :IP2 - SRC.ADR (R1)
:WORD TP19 :IP3 - SRC DATA
:WORD T0 :IP4 - SRC SURR DATA
:WORD T0 :IP5 - SRC SURR LEN
:WORD T5 :IP6- UNUSED PORTION OF REGISTER 4
:WORD TSPA :IP7 - SPECIAL HANDLING
:WORD 0 :IP10
:WORD 0 :IP11
:WORD 0 :IP12
:WORD 0 :IP13
:WORD 0 :IP14
:WORD 0 :IP15
:WORD 0 :IP16
:WORD 0 :IP17
:WORD 0 :IP20
:WORD 0 :IP21
:WORD 0 :IP22
:WORD 0 :IP23
:WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: TOTAL # OF TEST CONDITIONS = 8
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)8 = 8
:

16792
16793
16794 075570 000025
16795 075572 000003
16796 075574 105512
16797 075576 103012
16798 075600 106010
16799 075602 102436
16800 075604 102436
16801 075606 103704
16802 075610 102436
16803 075612 000000
16804 075614 000000
16805 075616 000000
16806 075620 000000
16807 075622 000000
16808 075624 000000
16809 075626 000000
16810 075630 000000
16811 075632 000000
16812 075634 000000
16813 075636 000000
16814 075640 000000
16815 075642 000000
16816
16817
16818
16819
16820
16821
16822
16823
16824
16825
16826

:ENTRY 41A - INSTRUCTION UNDER TEST = CVTPL

:ICPL3: .WORD 25 :INST = CVTPL
:WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:WORD T33A :IP1 - SRC.LEN (R0)
:WORD T2 :IP2 - SRC.ADR (R1)
:WORD TP19A :IP3 - SRC DATA
:WORD T0 :IP4 - SRC SURR DATA
:WORD T0 :IP5 - SRC SURR LEN
:WORD T5 :IP6- UNUSED PORTION OF REGISTER 4
:WORD T0 :IP7 - SPECIAL HANDLING
:WORD 0 :IP10
:WORD 0 :IP11
:WORD 0 :IP12
:WORD 0 :IP13
:WORD 0 :IP14
:WORD 0 :IP15
:WORD 0 :IP16
:WORD 0 :IP17
:WORD 0 :IP20
:WORD 0 :IP21
:WORD 0 :IP22
:WORD 0 :IP23
:WORD 0 :IP24

:THIS TABLE EXERCISES THE FOLLOWING TEST CONDITION

:SOURCE LENGTH - 12
:SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE DATA - 3 X 2 ** 31 = 6442450944 +
:- 3 X 2 ** 31 (-) = 6442450944 -

:TOTAL # OF TEST CONDITIONS = 2
:TOTAL # OF TESTS = (2 DATA TYPES + 1 IN-LINE) 2 = 6

16828
16829
16830
16831 075644 000015
16832 075646 000001
16833 075650 105472
16834 075652 103012
16835 075654 106252
16836 075656 103766
16837 075660 104004
16838 075662 103704
16839 075664 102436
16840 075666 000000
16841 075670 000000
16842 075672 000000
16843 075674 000000
16844 075676 000000
16845 075700 000000
16846 075702 000000
16847 075704 000000
16848 075706 000000
16849 075710 000000
16850 075712 000000
16851 075714 000000
16852 075716 000000
16853
16854
16855
16856
16857
16858
16859
16860
16861
16862
16863
16864

CVTNL TABLES
:SBTTL
:ENTRY 42 - INSTRUCTION UNDER TEST = CVTNL
:ICZL: .WORD 15 ;INST = CVTNL
 .WORD 1 ;TYPE = 1
 .WORD T33 ;IP1 - SRC.LEN (R0)
 .WORD T2 ;IP2 - SRC.ADR (R1)
 .WORD T37 ;IP3 - SRC DATA
 .WORD T7 ;IP4 - SRC SURR DATA
 .WORD T10 ;IP5 - SRC SURR LEN
 .WORD T5 ;IP6 - UNUSED PORTION OF REGISTER 4
 .WORD T0 ;IP7 - SPECIAL HANDLING
 .WORD 0 ;IP10
 .WORD 0 ;IP11
 .WORD 0 ;IP12
 .WORD 0 ;IP13
 .WORD 0 ;IP14
 .WORD 0 ;IP15
 .WORD 0 ;IP16
 .WORD 0 ;IP17
 .WORD 0 ;IP20
 .WORD 0 ;IP21
 .WORD 0 ;IP22
 .WORD 0 ;IP23
 .WORD 0 ;IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:SOURCE LENGTH - 0,1,20
:SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
 - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:TOTAL # OF TEST CONDITIONS = 9
:TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)9 = 63
:

16866
16867
16868
16869
16870 075720 000015
16871 075722 000001
16872 075724 105512
16873 075726 103012
16874 075730 107442
16875 075732 103766
16876 075734 104004
16877 075736 103704
16878 075740 102436
16879 075742 000000
16880 075744 000000
16881 075746 000000
16882 075750 000000
16883 075752 000000
16884 075754 000000
16885 075756 000000
16886 075760 000000
16887 075762 000000
16888 075764 000000
16889 075766 000000
16890 075770 000000
16891 075772 000000
16892
16893
16894
16895
16896
16897
16898
16899
16900
16901
16902
16903
16904

:
: ENTRY 43 - INSTRUCTION UNDER TEST = CVTNL
:

ICZL1: .WORD 15 ;INST = CVTNL
 .WORD 1 ;TYPE = 1
 .WORD T33A ;IP1 - SRC.LEN (R0)
 .WORD T2 ;IP2 - SRC.ADR (R1)
 .WORD T52 ;IP3 - SRC DATA
 .WORD T7 ;IP4 - SRC SURR DATA
 .WORD T10 ;IP5 - SRC SURR LEN
 .WORD T5 ;IP6 - UNUSED PORTION OF REGISTER 4
 .WORD T0 ;IP7 - SPECIAL HANDLING
 .WORD 0 ;IP10
 .WORD 0 ;IP11
 .WORD 0 ;IP12
 .WORD 0 ;IP13
 .WORD 0 ;IP14
 .WORD 0 ;IP15
 .WORD 0 ;IP16
 .WORD 0 ;IP17
 .WORD 0 ;IP20
 .WORD 0 ;IP21
 .WORD 0 ;IP22
 .WORD 0 ;IP23
 .WORD 0 ;IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

:
: SOURCE LENGTH - 12
: SOURCE ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE DATA - DIGITS FROM STRING = 2,147,483,648+
: - DIGITS FROM STRING = 2,147,483,647+
: - DIGITS FROM STRING = 2,147,483,648-
: - DIGITS FROM STRING = 2,147,483,649-
:

: TOTAL # OF TEST CONDITIONS = 4
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)4 = 28
:

16906
16907
16908 075774 000015
16909 075776 000001
16910 076000 102462
16911 076002 103022
16912 076004 106340
16913 076006 102436
16914 076010 102436
16915 076012 103704
16916 076014 102706
16917 076016 000000
16918 076020 000000
16919 076022 000000
16920 076024 000000
16921 076026 000000
16922 076030 000000
16923 076032 000000
16924 076034 000000
16925 076036 000000
16926 076040 000000
16927 076042 000000
16928 076044 000000
16929 076046 000000
16930
16931
16932
16933
16934
16935
16936
16937
16938
16939

:ENTRY 44 - INSTRUCTION UNDER TEST = CVTNL

```

:ICZL2: .WORD 15          ;INST = CVTNL
         .WORD 1          ;TYPE = 1
         .WORD T1A       ;IP1 - SRC.LEN (R0)
         .WORD T2A       ;IP2 - SRC.ADR (R1)
         .WORD T219      ;IP3 - SRC DATA
         .WORD T0        ;IP4 - SRC SURR DATA
         .WORD T0        ;IP5 - SRC SURR LEN
         .WORD T5        ;IP6 - UNUSED PORTION OF REGISTER 4
         .WORD TSPA      ;IP7 - SPECIAL HANDLING
         .WORD 0         ;IP10
         .WORD 0         ;IP11
         .WORD 0         ;IP12
         .WORD 0         ;IP13
         .WORD 0         ;IP14
         .WORD 0         ;IP15
         .WORD 0         ;IP16
         .WORD 0         ;IP17
         .WORD 0         ;IP20
         .WORD 0         ;IP21
         .WORD 0         ;IP22
         .WORD 0         ;IP23
         .WORD 0         ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
:   SOURCE LENGTH - 0,1,2,3,4,5,11,20
:   SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
:   SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 8
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)8 = 8
:

```

16941
16942
16943
16944 076050 000022
16945 076052 000003
16946 076054 102436
16947 076056 103012
16948 076060 102436
16949 076062 104260
16950 076064 102442
16951 076066 102436
16952 076070 105176
16953 076072 103766
16954 076074 104004
16955 076076 105176
16956 076100 105050
16957 076102 102436
16958
16959 076104 104020
16960 076106 104036
16961 076110 104054
16962 076112 104070
16963 076114 102436
16964 076116 000000
16965 076120 000000
16966 076122 000000
16967
16968
16969
16970
16971
16972
16973
16974
16975
16976
16977
16978
16979
16980
16981

.SBTTL ADDP TABLES
:ENTRY 45 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP: .WORD 22          ;INST=ADDP
        .WORD 3         ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T0        ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T0        ;IP3 - SRC2.LEN
        .WORD T16       ;IP4 - SRC2.ADR
        .WORD XT1       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22       ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T22       ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11       ;IP15 - DST DATA
        .WORD T12       ;IP16 - DST SURR DATA
        .WORD T13       ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRINGS ALIGNED WITH DESTINATION STRING
:
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGNT +
:
:TOTAL # OF TEST CONDITIONS = 6
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)6 = 18
:

```

16983
16984
16985 076124 000022
16986 076126 000001
16987 076130 102436
16988 076132 103012
16989 076134 105066
16990 076136 104260
16991 076140 102442
16992 076142 102436
16993 076144 105176
16994 076146 103766
16995 076150 104004
16996 076152 104774
16997 076154 105050
16998 076156 102436
16999
17000 076160 104020
17001 076162 104036
17002 076164 104054
17003 076166 104070
17004 076170 102436
17005 076172 000000
17006 076174 000000
17007 076176 000000
17008
17009
17010
17011
17012
17013
17014
17015
17016
17017
17018
17019
17020
17021
17022
17023
17024
17025

:ENTRY 46 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP1: .WORD 22          ;INST=ADDP
        .WORD 1          ;TYPE = 1
        .WORD T0         ;IP1 - SRC1.LEN
        .WORD T2         ;IP2 - SRC1.ADR
        .WCRD T21        ;IP3 - SRC2.LEN
        .WORD T16        ;IP4 - SRC2.ADR
        .WORD XT1        ;IP5 - DST.LEN
        .WORD T0         ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22        ;IP7 - SRC1 DATA
        .WORD T7         ;IP10 - SRC1 SURR DATA
        .WORD T10        ;IP11 - SRC1 SURR LEN
        .WORD T17        ;IP12 - SRC2 DATA
        .WORD T20        ;IP13 - SRC2 SURR DATA
        .WORD T0         ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11        ;IP15 - DST DATA
        .WORD T12        ;IP16 - DST SURR DATA
        .WORD T13        ;IP17 - DST SURR LEN
        .WORD T14        ;IP20 - SEPARATION CONSTRAINT
        .WORD T0         ;IP21 - SPECIAL HANDLING
        .WORD 0          ;IP22
        .WORD 0          ;IP23
        .WORD 0          ;IP24
    
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                                     - ALL DIGITS IDENTICAL = 5; SIGN +
:                                     - ALL DIGITS IDENTICAL = 3; SIGN -
:                                     - ALL DIGITS IDENTICAL = 0; SIGN +
    
```

: TOTAL # OF TEST CONDITIONS = 48

: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144

17027
 17028
 17029 076200 000022
 17030 076202 000001
 17031 076204 105066
 17032 076206 103012
 17033 076210 102436
 17034 076212 104260
 17035 076214 102442
 17036 076216 102436
 17037 076220 104774
 17038 076222 103766
 17039 076224 104004
 17040 076226 105176
 17041 076230 105050
 17042 076232 102436
 17043
 17044 076234 104020
 17045 076236 104036
 17046 076240 104054
 17047 076242 104070
 17048 076244 102436
 17049 076246 000000
 17050 076250 000000
 17051 076252 000000
 17052
 17053
 17054
 17055
 17056
 17057
 17058
 17059
 17060
 17061
 17062
 17063
 17064
 17065
 17066
 17067
 17068
 17069

:ENTRY 47 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP2: .WORD 22          ;INST=ADDP
        .WORD 1          ;TYPE = 1
        .WORD T21       ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T0        ;IP3 - SRC2.LEN
        .WORD T16       ;IP4 - SRC2.ADR
        .WORD XT1       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17       ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T22       ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11       ;IP15 - DST DATA
        .WORD T12       ;IP16 - DST SURR DATA
        .WORD T13       ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                               - ALL DIGITS IDENTICAL = 5; SIGN +
:                               - ALL DIGITS IDENTICAL = 3; SIGN -
:                               - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:
: TOTAL # OF TEST CONDITIONS =48
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144
:
  
```


17071
17072
17073 076254 000022
17074 076256 000001
17075 076260 105102
17076 076262 103012
17077 076264 105120
17078 076266 104260
17079 076270 102642
17080 076272 102436
17081 076274 104774
17082 076276 103766
17083 076300 104004
17084 076302 104774
17085 076304 105050
17086 076306 102436
17087
17088 076310 104020
17089 076312 104036
17090 076314 104054
17091 076316 104102
17092 076320 102436
17093 076322 000000
17094 076324 000000
17095 076326 000000
17096
17097
17098
17099
17100
17101
17102
17103
17104
17105
17106
17107
17108
17109
17110
17111
17112
17113
17114
17115
17116

:ENTRY 48 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP3: .WORD 22          :INST=ADDP
        .WORD 1           :TYPE = 1
        .WORD T211        :IP1 - SRC1.LEN
        .WORD T2          :IP2 - SRC1.ADR
        .WORD T212        :IP3 - SRC2.LEN
        .WORD T16         :IP4 - SRC2.ADR
        .WORD T1P         :IP5 - DST.LEN
        .WORD T0          :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17         :IP7 - SRC1 DATA
        .WORD T7          :IP10 - SRC1 SURR DATA
        .WORD T10         :IP11 - SRC1 SURR LEN
        .WORD T17         :IP12 - SRC2 DATA
        .WORD T20         :IP13 - SRC2 SURR DATA
        .WORD T0          :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11         :IP15 - DST DATA
        .WORD T12         :IP16 - DST SURR DATA
        .WORD T13         :IP17 - DST SURR LEN
        .WORD T14A        :IP20 - SEPARATION CONSTRAINT
        .WORD T0          :IP21 - SPECIAL HANDLING
        .WORD 0           :IP22
        .WORD 0           :IP23
        .WORD 0           :IP24
    
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ALIGNED WITH DEST STRING
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                 - ALL DIGITS IDENTICAL = 5; SIGN +
:                 - ALL DIGITS IDENTICAL = 3; SIGN -
:                 - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                 - ALL DIGITS IDENTICAL = 5; SIGN +
:                 - ALL DIGITS IDENTICAL = 3; SIGN -
:                 - ALL DIGITS IDENTICAL = 0; SIGN +
:
    
```

```

: TOTAL # OF TEST CONDITIONS = 384
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)384 = 1152
:
    
```

17118
17119
17120 076330 000022
17121 076332 000003
17122 076334 102462
17123 076336 103022
17124 076340 102462
17125 076342 104142
17126 076344 102462
17127 076346 102436
17128 076350 106000
17129 076352 102436
17130 076354 102436
17131 076356 106000
17132 076360 102436
17133 076362 102436
17134
17135 076364 102436
17136 076366 102436
17137 076370 102436
17138 076372 104070
17139 076374 102706
17140 076376 000000
17141 076400 000000
17142 076402 000000
17143
17144
17145
17146
17147
17148
17149
17150
17151
17152
17153
17154
17155
17156

:ENTRY 49 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP4: .WORD 22          :INST=ADDP
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1A       :IP1 - SRC1.LEN
        .WORD T2A       :IP2 - SRC1.ADR
        .WORD T1A       :IP3 - SRC2.LEN
        .WORD T16A      :IP4 - SRC2.ADR
        .WORD T1A       :IP5 - DST.LEN
        .WORD T0        :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TP19      :IP7 - SRC1 DATA
        .WORD T0        :IP10 - SRC1 SURR DATA
        .WORD T0        :IP11 - SRC1 SURR LEN
        .WORD TP19      :IP12 - SRC2 DATA
        .WORD T0        :IP13 - SRC2 SURR DATA
        .WORD T0        :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        .WORD T0        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        :IP15 - DST DATA
        .WORD T0        :IP16 - DST SURR DATA
        .WORD T0        :IP17 - DST SURR LEN
        .WORD T14       :IP20 - SEPARATION CONSTRAINT
        .WORD TSPA      :IP21 - SPECIAL HANDLING
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:

```

```

:TOTAL # OF TEST CONDITIONS = 512
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
:

```

```

17158      .SBTTL      ADDN TABLES
17159      ;ENTRY 50 - INSTRUCTION UNDER TEST = ADDN
17160      ;
17161      ;ADDN:  .WORD   12      ;INST=ADDN
17162      .WORD   1      ;TYPE = 1
17163      .WORD   T0      ;IP1 - SRC1.LEN
17164      .WORD   T2      ;IP2 - SRC1.ADR
17165      .WORD   T0      ;IP3 - SRC2.LEN
17166      .WORD   T16Z    ;IP4 - SRC2.ADR
17167      .WORD   XT1     ;IP5 - DST.LEN
17168      .WORD   T0      ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
17169      .WORD   T42     ;IP7 - SRC1 DATA
17170      .WORD   T7      ;IP10 - SRC1 SURR DATA
17171      .WORD   T10     ;IP11 - SRC1 SURR LEN
17172      .WORD   T42     ;IP12 - SRC2 DATA
17173      .WORD   T20     ;IP13 - SRC2 SURR DATA
17174      .WORD   T0      ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
17175      ; AS NOT TO DESTROY ANY OF SRC1)
17176      .WORD   T11     ;IP15 - DST DATA
17177      .WORD   T0      ;IP16 - DST SURR DATA
17178      .WORD   T0      ;IP17 - DST SURR LEN
17179      .WORD   T43     ;IP20 - SEPARATION CONSTRAINT
17180      .WORD   T0      ;IP21 - SPECIAL HANDLING
17181      .WORD   0      ;IP22
17182      .WORD   0      ;IP23
17183      .WORD   0      ;IP24

```

;THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

;
; SOURCE 1 LENGTH - 0
; SOURCE 2 LENGTH - 0
; DESTINATION LENGTH - 0,1,5
; SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
; SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
;                               - SOURCE 2 STRINGS ALIGNED WITH DESTINATION STRING
; SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
; SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE =17
;
;TOTAL # OF TEST CONDITIONS = 6
;TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)6 = 42
;

```

17184
17185
17186
17187
17188
17189
17190
17191
17192
17193
17194
17195
17196
17197
17198

17200
17201
17202 076460 000012
17203 076462 000001
17204 076464 102436
17205 076466 103012
17206 076470 105066
17207 076472 104632
17208 076474 102442
17209 076476 102436
17210 076500 106530
17211 076502 103766
17212 076504 104004
17213 076506 106252
17214 076510 105050
17215 076512 102436
17216
17217 076514 104020
17218 076516 104036
17219 076520 104054
17220 076522 104070
17221 076524 102436
17222 076526 000000
17223 076530 000000
17224 076532 000000
17225
17226
17227
17228
17229
17230
17231
17232
17233
17234
17235
17236
17237
17238
17239
17240
17241

:ENTRY 51 - INSTRUCTION UNDER TEST = ADDN

:ADDN1: .WORD 12 :INST=ADDN
: .WORD 1 :TYPE = 1
: .WORD T0 :IP1 - SRC1.LEN
: .WORD T2 :IP2 - SRC1.ADR
: .WORD T21 :IP3 - SRC2.LEN
: .WORD T16Z :IP4 - SRC2.ADR
: .WORD XT1 :IP5 - DST.LEN
: .WORD T0 :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
: .WORD T42 :IP7 - SRC1 DATA
: .WORD T7 :IP10 - SRC1 SURR DATA
: .WORD T10 :IP11 - SRC1 SURR LEN
: .WORD T37 :IP12 - SRC2 DATA
: .WORD T20 :IP13 - SRC2 SURR DATA
: .WORD T0 :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
: AS NOT TO DESTROY ANY OF SRC1)
: .WORD T11 :IP15 - DST DATA
: .WORD T12 :IP16 - DST SURR DATA
: .WORD T13 :IP17 - DST SURR LEN
: .WORD T14 :IP20 - SEPARATION CONSTRAINT
: .WORD T0 :IP21 - SPECIAL HANDLING
: .WORD 0 :IP22
: .WORD 0 :IP23
: .WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
: - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
: - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
: TOTAL # OF TEST CONDITIONS = 36
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)36 = 252

17243
17244
17245 076534 000012
17246 076536 000001
17247 076540 105066
17248 076542 103012
17249 076544 102436
17250 076546 104632
17251 076550 102442
17252 076552 102436
17253 076554 106252
17254 076556 103766
17255 076560 104004
17256 076562 106530
17257 076564 105050
17258 076566 102436
17259
17260 076570 104020
17261 076572 104036
17262 076574 104054
17263 076576 104070
17264 076600 102436
17265 076602 000000
17266 076604 000000
17267 076606 000000
17268
17269
17270
17271
17272
17273
17274
17275
17276
17277
17278
17279
17280
17281
17282
17283
17284

:ENTRY 52 - INSTRUCTION UNDER TEST = ADDN

```

:ADDN2: .WORD 12          ;INST=ADDN
        .WORD 1          ;TYPE = 1
        .WORD T21       ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T0        ;IP3 - SRC2.LEN
        .WORD T16Z      ;IP4 - SRC2.ADR
        .WORD XT1       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T37       ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T42       ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11       ;IP15 - DST DATA
        .WORD T12       ;IP16 - DST SURR DATA
        .WORD T13       ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:                                     - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:                                     - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
:
: TOTAL # OF TEST CONDITIONS = 36
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)36 = 252
:

```

17286
17287
17288 076610 000012
17289 076612 000001
17290 076614 105102
17291 076616 103012
17292 076620 105120
17293 076622 104632
17294 076624 102642
17295 076626 102436
17296 076630 106302
17297 076632 103766
17298 076634 104004
17299 076636 106302
17300 076640 105050
17301 076642 102436
17302
17303 076644 104020
17304 076646 102436
17305 076650 102436
17306 076652 106562
17307 076654 102436
17308 076656 000000
17309 076660 000000
17310 076662 000000
17311
17312
17313
17314
17315
17316
17317
17318
17319
17320
17321
17322
17323
17324
17325
17326
17327
17328
17329
17330
17331

:ENTRY 53 - INSTRUCTION UNDER TEST = ADDN

```

:ADDN3: .WORD 12          ;INST=ADDN
        .WORD 1          ;TYPE = 1
        .WORD T211      ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T212      ;IP3 - SRC2.LEN
        .WORD T16Z      ;IP4 - SRC2.ADR
        .WORD T1P       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T37A      ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T37A      ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11      ;IP15 - DST DATA
        .WORD T0        ;IP16 - DST SURR DATA
        .WORD T0        ;IP17 - DST SURR LEN
        .WORD T43A      ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
    
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ALIGNED WITH DEST STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:                 - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:                 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:                 - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:                 - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:                 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:                 - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
    
```

```

:TOTAL # OF TEST CONDITIONS = 384
:TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)384 = 2688
:
    
```

17333
17334
17335 076664 000012
17336 076666 000003
17337 076670 102462
17338 076672 103022
17339 076674 102462
17340 076676 104564
17341 076700 102462
17342 076702 102436
17343 076704 106340
17344 076706 102436
17345 076710 102436
17346 076712 106340
17347 076714 102436
17348 076716 102436
17349
17350 076720 102436
17351 076722 102436
17352 076724 102436
17353 076726 104070
17354 076730 102706
17355 076732 000000
17356 076734 000000
17357 076736 000000
17358
17359
17360
17361
17362
17363
17364
17365
17366
17367
17368
17369
17370
17371

:ENTRY 54 - INSTRUCTION UNDER TEST = ADDN

```
ADDN4: .WORD 12          ;INST=ADDN
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1A       ;IP1 - SRC1.LEN
        .WORD T2A       ;IP2 - SRC1.ADR
        .WORD T1A       ;IP3 - SRC2.LEN
        .WORD T16ZA     ;IP4 - SRC2.ADR
        .WORD T1A       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TZ19     ;IP7 - SRC1 DATA
        .WORD T0        ;IP10 - SRC1 SURR DATA
        .WORD T0        ;IP11 - SRC1 SURR LEN
        .WORD TZ19     ;IP12 - SRC2 DATA
        .WORD T0        ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        ;IP15 - DST DATA
        .WORD T0        ;IP16 - DST SURR DATA
        .WORD T0        ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD TSPA      ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
        SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
        SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
        DESTINATION LENGTH - 0,1,2,3,4,5,11,20
        SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
        SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
        SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
        SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
```

```
:TOTAL # OF TEST CONDITIONS = 512
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
```

17373
17374
17375 076740 000012
17376 076742 000001
17377 076744 102520
17378 076746 103030
17379 076750 102520
17380 076752 104564
17381 076754 104070
17382 076756 102436
17383 076760 106340
17384 076762 102436
17385 076764 102436
17386 076766 106340
17387 076770 102436
17388 076772 102436
17389
17390 076774 102436
17391 076776 102436
17392 077000 102436
17393 077002 104070
17394 077004 102436
17395 077006 000000
17396 077010 000000
17397 077012 000000
17398
17399
17400
17401
17402
17403
17404
17405
17406
17407
17408
17409
17410
17411

:ENTRY 54A - INSTRUCTION UNDER TEST = ADDN

```

:ADDN5: .WORD 12          ;INST=ADDN
        .WORD 1          ;TYPE = 1
        .WORD T1C       ;IP1 - SRC1.LEN
        .WORD T2AA      ;IP2 - SRC1.ADR
        .WORD T1C       ;IP3 - SRC2.LEN
        .WORD T16ZA     ;IP4 - SRC2.ADR
        .WORD T14       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TZ19     ;IP7 - SRC1 DATA
        .WORD T0        ;IP10 - SRC1 SURR DATA
        .WORD T0        ;IP11 - SRC1 SURR LEN
        .WORD TZ19     ;IP12 - SRC2 DATA
        .WORD T0        ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        ;IP15 - DST DATA
        .WORD T0        ;IP16 - DST SURR DATA
        .WORD T0        ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 2,4,6
: SOURCE 2 LENGTH - 2,4,6
: DESTINATION LENGTH - 10
: SOURCE 1 ADDRESS - 200,201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:

```

```

:TOTAL # OF TEST CONDITIONS = 18
:TOTAL # OF TESTS = (6 DATA TYPES + 1 IN-LINE)18 = 126
:

```


17413
17414
17415
17416 077014 000023
17417 077016 000001
17418 077020 102436
17419 077022 103012
17420 077024 102436
17421 077026 104260
17422 077030 102442
17423 077032 102436
17424 077034 105176
17425 077036 103766
17426 077040 104004
17427 077042 105176
17428 077044 105050
17429 077046 102436
17430
17431 077050 104020
17432 077052 104036
17433 077054 104054
17434 077056 104070
17435 077060 102436
17436 077062 000000
17437 077064 000000
17438 077066 000000
17439
17440
17441
17442
17443
17444
17445
17446
17447
17448
17449
17450
17451
17452
17453

```

.SBTTL          SUBP TABLES
:ENTRY 55 - INSTRUCTION UNDER TEST = SUBP
:
:SUBP: .WORD    23          :INST=SUBP
        .WORD    1          :TYPE = 1
        .WORD   T0          :IP1 - SRC1.LEN
        .WORD   T2          :IP2 - SRC1.ADR
        .WORD   T0          :IP3 - SRC2.LEN
        .WORD  T16          :IP4 - SRC2.ADR
        .WORD  XT1          :IP5 - DST.LEN
        .WORD   T0          :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD  T22          :IP7 - SRC1 DATA
        .WORD   T7          :IP10 - SRC1 SURR DATA
        .WORD  T10          :IP11 - SRC1 SURR LEN
        .WORD  T22          :IP12 - SRC2 DATA
        .WORD  T20          :IP13 - SRC2 SURR DATA
        .WORD   T0          :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD  T11          :IP15 - DST DATA
        .WORD  T12          :IP16 - DST SURR DATA
        .WORD  T13          :IP17 - DST SURR LEN
        .WORD  T14          :IP20 - SEPARATION CONSTRAINT
        .WORD   T0          :IP21 - SPECIAL HANDLING
        .WORD    0          :IP22
        .WORD    0          :IP23
        .WORD    0          :IP24
    
```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:   SOURCE 1 LENGTH - 0
:   SOURCE 2 LENGTH - 0
:   DESTINATION LENGTH - 0,1,5
:   SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:   SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRINGS ALIGNED WITH DESTINATION STRING
:   SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:   SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGNT +
:
:TOTAL # OF TEST CONDITIONS = 6
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)6 = 18
:
    
```

17455
17456
17457
17458
17459 077070 000023
17460 077072 000001
17461 077074 102436
17462 077076 103012
17463 077100 105066
17464 077102 104260
17465 077104 102442
17466 077106 102436
17467 077110 105176
17468 077112 103766
17469 077114 104004
17470 077116 104774
17471 077120 105050
17472 077122 102436
17473
17474 077124 104020
17475 077126 104036
17476 077130 104054
17477 077132 104070
17478 077134 102436
17479 077136 000000
17480 077140 000000
17481 077142 000000
17482
17483
17484
17485
17486
17487
17488
17489
17490
17491
17492
17493
17494
17495
17496
17497
17498
17499

: ENTRY 56 - INSTRUCTION UNDER TEST = SUBP

```

:SUBP1: .WORD 23      ;INST=SUBP
        .WORD 1       ;TYPE = 1
        .WORD T0      ;IP1 - SRC1.LEN
        .WORD T2      ;IP2 - SRC1.ADR
        .WORD T21     ;IP3 - SRC2.LEN
        .WORD T16     ;IP4 - SRC2.ADR
        .WORD XT1     ;IP5 - DST.LEN
        .WORD T0      ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22     ;IP7 - SRC1 DATA
        .WORD T7      ;IP10 - SRC1 SURR DATA
        .WORD T10     ;IP11 - SRC1 SURR LEN
        .WORD T17     ;IP12 - SRC2 DATA
        .WORD T20     ;IP13 - SRC2 SURR DATA
        .WORD T0      ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11     ;IP15 - DST DATA
        .WORD T12     ;IP16 - DST SURR DATA
        .WORD T13     ;IP17 - DST SURR LEN
        .WORD T14     ;IP20 - SEPARATION CONSTRAINT
        .WORD T0      ;IP21 - SPECIAL HANDLING
        .WORD 0       ;IP22
        .WORD 0       ;IP23
        .WORD 0       ;IP24
    
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                                     - ALL DIGITS IDENTICAL = 5; SIGN +
:                                     - ALL DIGITS IDENTICAL = 3; SIGN -
:                                     - ALL DIGITS IDENTICAL = 0; SIGN +
    
```

```

: TOTAL # OF TEST CONDITIONS = 48
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144
:
    
```

17501
 17502
 17503 077144 000023
 17504 077146 000001
 17505 077150 105066
 17506 077152 103012
 17507 077154 102436
 17508 077156 104260
 17509 077160 102442
 17510 077162 102436
 17511 077164 104774
 17512 077166 103766
 17513 077170 104004
 17514 077172 105176
 17515 077174 105050
 17516 077176 102436
 17517
 17518 077200 104020
 17519 077202 104036
 17520 077204 104054
 17521 077206 104070
 17522 077210 102436
 17523 077212 000000
 17524 077214 000000
 17525 077216 000000
 17526
 17527
 17528
 17529
 17530
 17531
 17532
 17533
 17534
 17535
 17536
 17537
 17538
 17539
 17540
 17541
 17542
 17543

:ENTRY 57 - INSTRUCTION UNDER TEST = SUBP

```

:SUBP2: .WORD 23      ;INST=SUBP
        .WORD 1       ;TYPE = 1
        .WORD T21    ;IP1 - SRC1.LEN
        .WORD T2     ;IP2 - SRC1.ADR
        .WORD T0     ;IP3 - SRC2.LEN
        .WORD T16    ;IP4 - SRC2.ADR
        .WORD XT1    ;IP5 - DST.LEN
        .WORD T0     ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17    ;IP7 - SRC1 DATA
        .WORD T7     ;IP10 - SRC1 SURR DATA
        .WORD T10    ;IP11 - SRC1 SURR LEN
        .WORD T22    ;IP12 - SRC2 DATA
        .WORD T20    ;IP13 - SRC2 SURR DATA
        .WORD T0     ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11    ;IP15 - DST DATA
        .WORD T12    ;IP16 - DST SURR DATA
        .WORD T13    ;IP17 - DST SURR LEN
        .WORD T14    ;IP20 - SEPARATION CONSTRAINT
        .WORD T0     ;IP21 - SPECIAL HANDLING
        .WORD 0      ;IP22
        .WORD 0      ;IP23
        .WORD 0      ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                   - ALL DIGITS IDENTICAL = 5; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:
: TOTAL # OF TEST CONDITIONS =48
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144
:
:

```

17545
17546
17547 077220 000023
17548 077222 000001
17549 077224 105102
17550 077226 103012
17551 077230 105120
17552 077232 104260
17553 077234 102642
17554 077236 102436
17555 077240 104774
17556 077242 103766
17557 077244 104004
17558 077246 104774
17559 077250 105050
17560 077252 102436
17561
17562 077254 104020
17563 077256 104036
17564 077260 104054
17565 077262 104102
17566 077264 102436
17567 077266 000000
17568 077270 000000
17569 077272 000000
17570
17571
17572
17573
17574
17575
17576
17577
17578
17579
17580
17581
17582
17583
17584
17585
17586
17587
17588
17589
17590

:ENTRY 58 - INSTRUCTION UNDER TEST = SUBP

```

:SUBP3: .WORD 23          ;INST=SUBP
        .WORD 1          ;TYPE = 1
        .WORD T211      ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T212      ;IP3 - SRC2.LEN
        .WORD T16       ;IP4 - SRC2.ADR
        .WORD T1P       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17       ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T17       ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11       ;IP15 - DST DATA
        .WORD T12       ;IP16 - DST SURR DATA
        .WORD T13       ;IP17 - DST SURR LEN
        .WORD T14A      ;IP20 - SEPARATION CONSTRAINT
        .WORD T0        ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
    
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:   - SOURCE 2 STRING ALIGNED WITH DEST STRING
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:   - ALL DIGITS IDENTICAL = 5; SIGN +
:   - ALL DIGITS IDENTICAL = 3; SIGN -
:   - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:   - ALL DIGITS IDENTICAL = 5; SIGN +
:   - ALL DIGITS IDENTICAL = 3; SIGN -
:   - ALL DIGITS IDENTICAL = 0; SIGN +
    
```

```

:TOTAL # OF TEST CONDITIONS = 384
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)384 = 1152
:
    
```

17592
17593
17594 077274 000023
17595 077276 000001
17596 077300 102462
17597 077302 103022
17598 077304 102462
17599 077306 104142
17600 077310 102462
17601 077312 102436
17602 077314 106000
17603 077316 102436
17604 077320 102436
17605 077322 106000
17606 077324 102436
17607 077326 102436
17608
17609 077330 102436
17610 077332 102436
17611 077334 102436
17612 077336 104070
17613 077340 102706
17614 077342 000000
17615 077344 000000
17616 077346 000000
17617
17618
17619
17620
17621
17622
17623
17624
17625
17626
17627
17628
17629
17630

:ENTRY 59 - INSTRUCTION UNDER TEST = SUBP

```
ISUBP4: .WORD 23          ;INST=SUBP
        .WORD 1          ;TYPE = 1
        .WORD T1A       ;IP1 - SRC1.LEN
        .WORD T2A       ;IP2 - SRC1.ADR
        .WORD T1A       ;IP3 - SRC2.LEN
        .WORD T16A      ;IP4 - SRC2.ADR
        .WORD T1A       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TP19      ;IP7 - SRC1 DATA
        .WORD T0        ;IP10 - SRC1 SURR DATA
        .WORD T0        ;IP11 - SRC1 SURR LEN
        .WORD TP19      ;IP12 - SRC2 DATA
        .WORD T0        ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        ;IP15 - DST DATA
        .WORD T0        ;IP16 - DST SURR DATA
        .WORD T0        ;IP17 - DST SURR LEN
        .WORD T14       ;IP20 - SEPARATION CONSTRAINT
        .WORD TSPA      ;IP21 - SPECIAL HANDLING
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
```

```
:TOTAL # OF TEST CONDITIONS = 512
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
:
```

17632
17633
17634
17635 077350 000013
17636 077352 000001
17637 077354 102436
17638 077356 103012
17639 077360 102436
17640 077362 104632
17641 077364 102442
17642 077366 102436
17643 077370 106530
17644 077372 103766
17645 077374 104004
17646 077376 106530
17647 077400 105050
17648 077402 102436
17649
17650 077404 104020
17651 077406 102436
17652 077410 102436
17653 077412 106546
17654 077414 102436
17655 077416 000000
17656 077420 000000
17657 077422 000000
17658
17659
17660
17661
17662
17663
17664
17665
17666
17667
17668
17669
17670
17671
17672

```

.SBTTL          SUBN TABLES
:ENTRY 60 - INSTRUCTION UNDER TEST = SUBN
:
:SUBN: .WORD    13          ;INST=SUBN
        .WORD     1          ;TYPE = 1
        .WORD    T0          ;IP1 - SRC1.LEN
        .WORD    T2          ;IP2 - SRC1.ADR
        .WORD    T0          ;IP3 - SRC2.LEN
        .WORD   T16Z         ;IP4 - SRC2.ADR
        .WORD    XT1         ;IP5 - DST.LEN
        .WORD    T0          ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD   T42         ;IP7 - SRC1 DATA
        .WORD    T7          ;IP10 - SRC1 SURR DATA
        .WORD   T10         ;IP11 - SRC1 SURR LEN
        .WORD   T42         ;IP12 - SRC2 DATA
        .WORD   T20         ;IP13 - SRC2 SURR DATA
        .WORD    T0          ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD   T11         ;IP15 - DST DATA
        .WORD    T0          ;IP16 - DST SURR DATA
        .WORD    T0          ;IP17 - DST SURR LEN
        .WORD   T43         ;IP20 - SEPARATION CONSTRAINT
        .WORD    T0          ;IP21 - SPECIAL HANDLING
        .WORD     0          ;IP22
        .WORD     0          ;IP23
        .WORD     0          ;IP24
    
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRINGS ALIGNED WITH DESTINATION STRING
:
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
:
: TOTAL # OF TEST CONDITIONS = 6
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)6 = 42
:
    
```

17674
17675
17676 077424 000013
17677 077426 000001
17678 077430 102436
17679 077432 103012
17680 077434 105066
17681 077436 104632
17682 077440 102442
17683 077442 102436
17684 077444 106530
17685 077446 103766
17686 077450 104004
17687 077452 106252
17688 077454 105050
17689 077456 102436
17690
17691 077460 104020
17692 077462 104036
17693 077464 104054
17694 077466 104070
17695 077470 102436
17696 077472 000000
17697 077474 000000
17698 077476 000000
17699
17700
17701
17702
17703
17704
17705
17706
17707
17708
17709
17710
17711
17712
17713
17714
17715

:ENTRY 61 - INSTRUCTION UNDER TEST = SUBN

```

:SUBN1: .WORD 13      ;INST=SUBN
        .WORD 1       ;TYPE = 1
        .WORD T0      ;IP1 - SRC1.LEN
        .WORD T2      ;IP2 - SRC1.ADR
        .WORD T21     ;IP3 - SRC2.LEN
        .WORD T16Z    ;IP4 - SRC2.ADR
        .WORD XT1     ;IP5 - DST.LEN
        .WORD T0      ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T42     ;IP7 - SRC1 DATA
        .WORD T7       ;IP10 - SRC1 SURR DATA
        .WORD T10     ;IP11 - SRC1 SURR LEN
        .WORD T37     ;IP12 - SRC2 DATA
        .WORD T20     ;IP13 - SRC2 SURR DATA
        .WORD T0      ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11     ;IP15 - DST DATA
        .WORD T12     ;IP16 - DST SURR DATA
        .WORD T13     ;IP17 - DST SURR LEN
        .WORD T14     ;IP20 - SEPARATION CONSTRAINT
        .WORD T0      ;IP21 - SPECIAL HANDLING
        .WORD 0       ;IP22
        .WORD 0       ;IP23
        .WORD 0       ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:   - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:   - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:   - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:
: TOTAL # OF TEST CONDITIONS = 36
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)36 = 252
:

```

17717
17718
17719
17720
17721 077500 000013
17722 077502 000001
17723 077504 105066
17724 077506 103012
17725 077510 102436
17726 077512 104632
17727 077514 102442
17728 077516 102436
17729 077520 106252
17730 077522 103766
17731 077524 104004
17732 077526 106530
17733 077530 105050
17734 077532 102436
17735
17736 077534 104020
17737 077536 104036
17738 077540 104054
17739 077542 104070
17740 077544 102436
17741 077546 000000
17742 077550 000000
17743 077552 000000
17744
17745
17746
17747
17748
17749
17750
17751
17752
17753
17754
17755
17756
17757
17758
17759
17760

: ENTRY 62 - INSTRUCTION UNDER TEST = SUBN

```

: SUBN2: .WORD 13          ; INST=SUBN
          .WORD 1          ; TYPE = 1
          .WORD T21       ; IP1 - SRC1.LEN
          .WORD T2        ; IP2 - SRC1.ADR
          .WORD T0        ; IP3 - SRC2.LEN
          .WORD T16Z      ; IP4 - SRC2.ADR
          .WORD XT1       ; IP5 - DST.LEN
          .WORD T0        ; IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD T37       ; IP7 - SRC1 DATA
          .WORD T7        ; IP10 - SRC1 SURR DATA
          .WORD T10       ; IP11 - SRC1 SURR LEN
          .WORD T42       ; IP12 - SRC2 DATA
          .WORD T20       ; IP13 - SRC2 SURR DATA
          .WORD T0        ; IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                        ; AS NOT TO DESTROY ANY OF SRC1)
          .WORD T11       ; IP15 - DST DATA
          .WORD T12       ; IP16 - DST SURR DATA
          .WORD T13       ; IP17 - DST SURR LEN
          .WORD T14       ; IP20 - SEPARATION CONSTRAINT
          .WORD T0        ; IP21 - SPECIAL HANDLING
          .WORD 0         ; IP22
          .WORD 0         ; IP23
          .WORD 0         ; IP24
    
```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRING ALIGNED WITH DESTINATION STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:                   - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:                   - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: TOTAL # OF TEST CONDITIONS = 36
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE) 36 = 252
:
    
```


17762
17763
17764 077554 000013
17765 077556 000001
17766 077560 105102
17767 077562 103012
17768 077564 105120
17769 077566 104632
17770 077570 102642
17771 077572 102436
17772 077574 106302
17773 077576 103766
17774 077600 104004
17775 077602 106302
17776 077604 105050
17777 077606 102436
17778
17779 077610 104020
17780 077612 102436
17781 077614 102436
17782 077616 106562
17783 077620 102436
17784 077622 000000
17785 077624 000000
17786 077626 000000
17787
17788
17789
17790
17791
17792
17793
17794
17795
17796
17797
17798
17799
17800
17801
17802
17803
17804
17805
17806
17807

:ENTRY 63 - INSTRUCTION UNDER TEST = SUBN

```

:SUBN3: .WORD 13          :INST=SUBN
        .WORD 1          :TYPE = 1
        .WORD T211       :IP1 - SRC1.LEN
        .WORD T2         :IP2 - SRC1.ADR
        .WORD T212       :IP3 - SRC2.LEN
        .WORD T16Z       :IP4 - SRC2.ADR
        .WORD T1P        :IP5 - DST.LEN
        .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T37A       :IP7 - SRC1 DATA
        .WORD T7         :IP10 - SRC1 SURR DATA
        .WORD T10        :IP11 - SRC1 SURR LEN
        .WORD T37A       :IP12 - SRC2 DATA
        .WORD T20        :IP13 - SRC2 SURR DATA
        .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11        :IP15 - DST DATA
        .WORD T0         :IP16 - DST SURR DATA
        .WORD T0         :IP17 - DST SURR LEN
        .WORD T43A       :IP20 - SEPARATION CONSTRAANT
        .WORD T0         :IP21 - SPECIAL HANDLING
        .WCRD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
    
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:   - SOURCE 2 STRING ALIGNED WITH DEST STRING
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:   - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:   - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:   - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +; HIGH NIBBLE = 7
:   - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:   - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:   - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
    
```

```

:TOTAL # OF TEST CONDITIONS = 384
:TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)384 = 2688
:
    
```

17809
 17810
 17811 077630 000013
 17812 077632 000001
 17813 077634 102462
 17814 077636 103022
 17815 077640 102462
 17816 077642 104564
 17817 077644 102462
 17818 077646 102436
 17819 077650 106340
 17820 077652 102436
 17821 077654 102436
 17822 077656 106340
 17823 077660 102436
 17824 077662 102436
 17825
 17826 077664 102436
 17827 077666 102436
 17828 077670 102436
 17829 077672 104070
 17830 077674 102706
 17831 077676 000000
 17832 077700 000000
 17833 077702 000000
 17834
 17835
 17836
 17837
 17838
 17839
 17840
 17841
 17842
 17843
 17844
 17845
 17846
 17847

:ENTRY 64 - INSTRUCTION UNDER TEST = SUBN

```

:SUBN4: .WORD 13          :INST=SUBN
        .WORD 1          :TYPE = 1
        .WORD T1A       :IP1 - SRC1.LEN
        .WORD T2A       :IP2 - SRC1.ADR
        .WORD T1A       :IP3 - SRC2.LEN
        .WORD T16ZA     :IP4 - SRC2.ADR
        .WORD T1A       :IP5 - DST.LEN
        .WORD T0        :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TZ19     :IP7 - SRC1 DATA
        .WORD T0        :IP10 - SRC1 SURR DATA
        .WORD T0        :IP11 - SRC1 SURR LEN
        .WORD TZ19     :IP12 - SRC2 DATA
        .WORD T0        :IP13 - SRC2 SURR DATA
        .WORD T0        :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        :IP15 - DST DATA
        .WORD T0        :IP16 - DST SURR DATA
        .WORD T0        :IP17 - DST SURR LEN
        .WORD T14       :IP21 - SEPARATION CONSTRAINT
        .WORD TSPA      :IP21 - SPECIAL HANDLING
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 512
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
:
  
```

17849
17850
17851
17852 077704 000024
17853 077706 000003
17854 077710 102436
17855 077712 103012
17856 077714 102436
17857 077716 106630
17858 077720 103704
17859 077722 105176
17860 077724 105050
17861 077726 102436
17862 077730 105176
17863 077732 103766
17864 077734 104004
17865 077736 106546
17866 077740 102436
17867 077742 000000
17868 077744 000000
17869 077746 000000
17870 077750 000000
17871 077752 000000
17872 077754 000000
17873 077756 000000
17874
17875
17876
17877
17878
17879
17880
17881
17882
17883
17884
17885
17886
17887

```
.SBTTL          CMPP TABLES
:ENTRY 65 - INSTRUCTION UNDER TEST = CMPP
:
:CMPP: .WORD 24          ;INST = CMPP
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T0         ;IP1 - SRC1.LEN
        .WORD T2         ;IP2 - SRC1.ADR
        .WORD T0         ;IP3 - SRC2.LEN
        .WORD T44        ;IP4 - SRC2.ADR
        .WORD T5         ;IP5 - UNUSED PORTION OF REG. 4
        .WORD T22        ;IP6 - SRC1.DATA
        .WORD T20        ;IP7 - SRC1.SURR.DATA
        .WORD T0         ;IP10 - SRC1.SURR.LEN
        .WORD T22        ;IP11 - SRC2.DATA
        .WORD T7         ;IP12 - SRC2.SURR.DATA
        .WORD T10        ;IP13 - SRC2.SURR.LEN
        .WORD T43        ;IP14 - SEPARATION CONSTANT
        .WORD T0         ;IP15 - SPECIAL HANDLING
        .WORD 0          ;IP16
        .WORD 0          ;IP17
        .WORD 0          ;IP20
        .WORD 0          ;IP21
        .WORD 0          ;IP22
        .WORD 0          ;IP23
        .WORD 0          ;IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
:   SOURCE 1 LENGTH - 0
:   SOURCE 2 LENGTH - 0
:   SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:   SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
:                       - ALIGNED SOURCE 1 - SOURCE 2 STRINGS
:   SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:   SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
```

```
:TOTAL # OF TEST CONDITIONS = 2
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)2 = 6
:
```

17889
17890
17891
17892 077760 000024
17893 077762 000001
17894 077764 102436
17895 077766 103012
17896 077770 105066
17897 077772 106630
17898 077774 103704
17899 077776 105176
17900 100000 105050
17901 100002 102436
17902 100004 106750
17903 100006 103766
17904 100010 104004
17905 100012 106546
17906 100014 102436
17907 100016 000000
17908 100020 000000
17909 100022 000000
17910 100024 000000
17911 100026 000000
17912 100030 000000
17913 100032 000000
17914
17915
17916
17917
17918
17919
17920
17921
17922
17923
17924
17925
17926
17927
17928
17929

:ENTRY 66 - INSTRUCTION UNDER TEST = CMPP

```
ICMPP1: .WORD 24          :INST = CMPP
         .WORD 1          :TYPE = 1
         .WORD T0         :IP1 - SRC1.LEN
         .WORD T2         :IP2 - SRC1.ADR
         .WORD T21        :IP3 - SRC2.LEN
         .WORD T44        :IP4 - SRC2.ADR
         .WORD T5         :IP5 - UNUSED PORTION OF REG. 4
         .WORD T22        :IP6 - SRC1.DATA
         .WORD T20        :IP7 - SRC1.SURR.DATA
         .WORD T0         :IP10 - SRC1.SURR.LEN
         .WORD T45        :IP11 - SRC2.DATA
         .WORD T7         :IP12 - SRC2.SURR.DATA
         .WORD T10        :IP13 - SRC2.SURR.LEN
         .WORD T43        :IP14 - SEPARATION CONSTANT
         .WORD T0         :IP15 - SPECIAL HANDLING
         .WORD 0          :IP16
         .WORD 0          :IP17
         .WORD 0          :IP20
         .WORD 0          :IP21
         .WORD 0          :IP22
         .WORD 0          :IP23
         .WORD 0          :IP24
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF THE STRINGS
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
:                   - ALL DIGITS IDENTICAL = 0; SIGN -
:
```

```
:TOTAL # OF TEST CONDITIONS = 8
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)8 = 24
:
```

17931
 17932
 17933
 17934 100034 000024
 17935 100036 000001
 17936 100040 105066
 17937 100042 103012
 17938 100044 102436
 17939 100046 106630
 17940 100050 103704
 17941 100052 106750
 17942 100054 105050
 17943 100056 102436
 17944 100060 105176
 17945 100062 103766
 17946 100064 104004
 17947 100066 106546
 17948 100070 102436
 17949 100072 000000
 17950 100074 000000
 17951 100076 000000
 17952 100100 000000
 17953 100102 000000
 17954 100104 000000
 17955 100106 000000
 17956
 17957
 17958
 17959
 17960
 17961
 17962
 17963
 17964
 17965
 17966
 17967
 17968
 17969
 17970
 17971

:ENTRY 67 - INSTRUCTION UNDER TEST = CMPP

```

:CMPP2: .WORD 24          :INST = CMPP
        .WORD 1          :TYPE = 1
        .WORD T21       :IP1 - SRC1.LEN
        .WORD T2        :IP2 - SRC1.ADR
        .WORD T0        :IP3 - SRC2.LEN
        .WORD T44       :IP4 - SRC2.ADR
        .WORD T5        :IP5 - UNUSED PORTION OF REG. 4
        .WORD T45       :IP6 - SRC1.DATA
        .WORD T20       :IP7 - SRC1.SURR.DATA
        .WORD T0        :IP10 - SRC1.SURR.LEN
        .WORD T22       :IP11 - SRC2.DATA
        .WORD T7        :IP12 - SRC2.SURR.DATA
        .WORD T10       :IP13 - SRC2.SURR.LEN
        .WORD T43       :IP14 - SEPARATION CONSTANT
        .WORD T0        :IP15 - SPECIAL HANDLING
        .WORD 0         :IP16
        .WORD 0         :IP17
        .WORD 0         :IP20
        .WORD 0         :IP21
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
:                   - ALL DIGITS IDENTICAL = 0; SIGN -
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
  
```

```

: TOTAL # OF TEST CONDITIONS = 8
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)8 = 24
:
  
```

17973
17974
17975
17976 100110 000024
17977 100112 000001
17978 100114 105102
17979 100116 103012
17980 100120 105120
17981 100122 106630
17982 100124 103704
17983 100126 106750
17984 100130 105050
17985 100132 102436
17986 100134 106750
17987 100136 103766
17988 100140 104004
17989 100142 106546
17990 100144 102436
17991 100146 000000
17992 100150 000000
17993 100152 000000
17994 100154 000000
17995 100156 000000
17996 100160 000000
17997 100162 000000
17998
17999
18000
18001
18002
18003
18004
18005
18006
18007
18008
18009
18010
18011
18012
18013
18014
18015
18016

:ENTRY 68 - INSTRUCTION UNDER TEST = CMPP

```

:CMPP3: .WORD 24          :INST = CMPP
        .WORD 1          :TYPE = 1
        .WORD T211      :IP1 - SRC1.LEN
        .WORD T2        :IP2 - SRC1.ADR
        .WORD T212      :IP3 - SRC2.LEN
        .WORD T44       :IP4 - SRC2.ADR
        .WORD T5        :IP5 - UNUSED PORTION OF REG. 4
        .WORD T45       :IP6 - SRC1.DATA
        .WORD T20       :IP7 - SRC1.SURR.DATA
        .WORD T0        :IP10 - SRC1.SURR.LEN
        .WORD T45       :IP11 - SRC2.DATA
        .WORD T7        :IP12 - SRC2.SURR.DATA
        .WORD T10       :IP13 - SRC2.SURR.LEN
        .WORD T43       :IP14 - SEPARATION CONSTANT
        .WORD T0        :IP15 - SPECIAL HANDLING
        .WORD 0         :IP16
        .WORD 0         :IP17
        .WORD 0         :IP20
        .WORD 0         :IP21
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
:                   - ALL DIGITS IDENTICAL = 0; SIGN -
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
:                   - ALL DIGITS IDENTICAL = 0; SIGN -
:

```

```

: TOTAL # OF TEST CONDITIONS = 128
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)128 = 384
:

```

18018
18019
18020 100164 000024
18021 100166 000003
18022 100170 102462
18023 100172 103022
18024 100174 102462
18025 100176 106576
18026 100200 103704
18027 100202 106000
18028 100204 102436
18029 100206 102436
18030 100210 106000
18031 100212 102436
18032 100214 102436
18033 100216 104070
18034 100220 102706
18035 100222 000000
18036 100224 000000
18037 100226 000000
18038 100230 000000
18039 100232 000000
18040 100234 000000
18041 100236 000000

:ENTRY 69 - INSTRUCTION UNDER TEST = CMPP

:CMPP4: .WORD 24 :INST = CMPP
:WORD 3 :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:WORD T1A :IP1 - SRC1.LEN
:WORD T2A :IP2 - SRC1.ADR
:WORD T1A :IP3 - SRC2.LEN
:WORD T44A :IP4 - SRC2.ADR
:WORD T5 :IP5 - UNUSED PORTION OF REG. 4
:WORD TP19 :IP6 - SRC1.DATA
:WORD T0 :IP7 - SRC1.SURR.DATA
:WORD T0 :IP10 - SRC1.SURR.LEN
:WORD TP19 :IP11 - SRC2.DATA
:WORD T0 :IP12 - SRC2.SURR.DATA
:WORD T0 :IP13 - SRC2.SURR.LEN
:WORD T14 :IP14 - SEPARATION CONSTANT
:WORD TSPA :IP15 - SPECIAL HANDLING
:WORD 0 :IP16
:WORD 0 :IP17
:WORD 0 :IP20
:WORD 0 :IP21
:WORD 0 :IP22
:WORD 0 :IP23
:WORD 0 :IP24

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF ANY OF THE STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +

:TOTAL # OF TEST CONDITIONS = 64
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)64 = 64

18042
18043
18044
18045
18046
18047
18048
18049
18050
18051
18052
18053
18054

18056
18057
18058
18059 100240 000014
18060 100242 000001
18061 100244 102436
18062 100246 103012
18063 100250 102436
18064 100252 107032
18065 100254 103704
18066 100256 106530
18067 100260 105050
18068 100262 102436
18069 100264 106530
18070 100266 103766
18071 100270 104004
18072 100272 104070
18073 100274 102436
18074 100276 000000
18075 100300 000000
18076 100302 000000
18077 100304 000000
18078 100306 000000
18079 100310 000000
18080 100312 000000

```
.SBTTL          CMPN TABLES
:ENTRY 70 - INSTRUCTION UNDER TEST = CMPN
:
:ICMPN: .WORD 14          ;INST = CMPN
        .WORD 1          ;TYPE = 1
        .WORD T0         ;IP1 - SRC1.LEN
        .WORD T2         ;IP2 - SRC1.ADR
        .WORD T0         ;IP3 - SRC2.LEN
        .WORD T46        ;IP4 - SRC2.ADR
        .WORD T5         ;IP5 - UNUSED PORTION OF REG. 4
        .WORD T42        ;IP6 - SRC1.DATA
        .WORD T20        ;IP7 - SRC1.SURR.DATA
        .WORD T0         ;IP10 - SRC1.SURR.LEN
        .WORD T42        ;IP11 - SRC2.DATA
        .WORD T7         ;IP12 - SRC2.SURR.DATA
        .WORD T10        ;IP13 - SRC2.SURR.LEN
        .WORD T14        ;IP14 - SEPARATION CONSTANT
        .WORD T0         ;IP15 - SPECIAL HANDLING
        .WORD 0          ;IP16
        .WORD 0          ;IP17
        .WORD 0          ;IP20
        .WORD 0          ;IP21
        .WORD 0          ;IP22
        .WORD 0          ;IP23
        .WORD 0          ;IP24
```

18081
18082
18083
18084
18085
18086
18087
18088
18089
18090
18091
18092
18093
18094
18095
18096

```
: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 0
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
:                   - STRINGS ADJACENT
:                   - STRINGS PARTIALLY OVERLAP
:                   - STRINGS COMPLETELY OVERLAP
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
:
: TOTAL # OF TEST CONDITIONS =
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE) =
:
```


18098
 18099
 18100
 18101 100314 000014
 18102 100316 000001
 18103 100320 102436
 18104 100322 103012
 18105 100324 105066
 18106 100326 107032
 18107 100330 103704
 18108 100332 106530
 18109 100334 105050
 18110 100336 102436
 18111 100340 107254
 18112 100342 103766
 18113 100344 104004
 18114 100346 104070
 18115 100350 102436
 18116 100352 000000
 18117 100354 000000
 18118 100356 000000
 18119 100360 000000
 18120 100362 000000
 18121 100364 000000
 18122 100366 000000
 18123
 18124
 18125
 18126
 18127
 18128
 18129
 18130
 18131
 18132
 18133
 18134
 18135
 18136
 18137
 18138
 18139
 18140
 18141

:ENTRY 71 - INSTRUCTION UNDER TEST = CMPN

```

:ICMPN1: .WORD 14          ;INST = CMPN
          .WORD 1          ;TYPE = 1
          .WORD T0        ;IP1 - SRC1.LEN
          .WORD T2        ;IP2 - SRC1.ADR
          .WORD T21       ;IP3 - SRC2.LEN
          .WORD T46       ;IP4 - SRC2.ADR
          .WORD T5        ;IP5 - UNUSED PORTION OF REG. 4
          .WORD T42       ;IP6 - SRC1.DATA
          .WORD T20       ;IP7 - SRC1.SURR.DATA
          .WORD T0        ;IP10 - SRC1.SURR.LEN
          .WORD T47       ;IP11 - SRC2.DATA
          .WORD T7        ;IP12 - SRC2.SURR.DATA
          .WORD T10       ;IP13 - SRC2.SURR.LEN
          .WORD T14       ;IP14 - SEPARATION CONSTANT
          .WORD T0        ;IP15 - SPECIAL HANDLING
          .WORD 0         ;IP16
          .WORD 0         ;IP17
          .WORD 0         ;IP20
          .WORD 0         ;IP21
          .WORD 0         ;IP22
          .WORD 0         ;IP23
          .WORD 0         ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
:                   - STRINGS ADJACENT
:                   - STRINGS PARTIALLY OVERLAP
:                   - STRINGS COMPLETELY OVERLAP
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 9; SIGN +; HIGH NIBBLE = 17
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 1
:                   - ALL DIGITS IDENTICAL = 8; SIGN +; HIGH NIBBLE = 17
:                   - ALL DIGITS IDENTICAL = 0; SIGN +; HIGH NIBBLE = 1
:                   - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 8
  
```

```

: TOTAL # OF TEST CONDITIONS =
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE) =
:
  
```

18143
 18144
 18145
 18146 100370 000014
 18147 100372 000001
 18148 100374 105066
 18149 100376 103012
 18150 100400 102436
 18151 100402 107032
 18152 100404 103704
 18153 100406 107254
 18154 100410 105050
 18155 100412 102436
 18156 100414 106530
 18157 100416 103766
 18158 100420 104004
 18159 100422 104070
 18160 100424 102436
 18161 100426 000000
 18162 100430 000000
 18163 100432 000000
 18164 100434 000000
 18165 100436 000000
 18166 100440 000000
 18167 100442 000000
 18168
 18169
 18170
 18171
 18172
 18173
 18174
 18175
 18176
 18177
 18178
 18179
 18180
 18181
 18182
 18183
 18184
 18185
 18186

:ENTRY 72 - INSTRUCTION UNDER TEST = CMPN

```

:ICMPN2: .WORD 14          ;INST = CMPN
          .WORD 1          ;TYPE = 1
          .WORD T21       ;IP1 - SRC1.LEN
          .WORD T2        ;IP2 - SRC1.ADR
          .WORD T0        ;IP3 - SRC2.LEN
          .WORD T46       ;IP4 - SRC2.ADR
          .WORD T5        ;IP5 - UNUSED PORTION OF REG. 4
          .WORD T47       ;IP6 - SRC1.DATA
          .WORD T20       ;IP7 - SRC1.SURR.DATA
          .WORD T0        ;IP10 - SRC1.SURR.LEN
          .WORD T42       ;IP11 - SRC2.DATA
          .WORD T7        ;IP12 - SRC2.SURR.DATA
          .WORD T10       ;IP13 - SRC2.SURR.LEN
          .WORD T14       ;IP14 - SEPARATION CONSTANT
          .WORD T0        ;IP15 - SPECIAL HANDLING
          .WORD 0         ;IP16
          .WORD 0         ;IP17
          .WORD 0         ;IP20
          .WORD 0         ;IP21
          .WORD 0         ;IP22
          .WORD 0         ;IP23
          .WORD 0         ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
:                   - STRINGS ADJACENT
:                   - STRINGS PARTIALLY OVERLAP
:                   - STRINGS COMPLETELY OVERLAP
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 8; SIGN -: HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 8; SIGN +: HIGH NIBBLE = 17
:                 - ALL DIGITS IDENTICAL = 0; SIGN +: HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 0; SIGN -: HIGH NIBBLE = 8
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 9; SIGN +: HIGH NIBBLE = 17
:
: TOTAL # OF TEST CONDITIONS =
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE) =
:
  
```

18188
 18189
 18190
 18191 100444 000014
 18192 100446 000001
 18193 100450 105102
 18194 100452 103012
 18195 100454 105120
 18196 100456 107032
 18197 100460 103704
 18198 100462 107254
 18199 100464 105050
 18200 100466 102436
 18201 100470 107254
 18202 100472 103766
 18203 100474 104004
 18204 100476 104070
 18205 100500 102436
 18206 100502 000000
 18207 100504 000000
 18208 100506 000000
 18209 100510 000000
 18210 100512 000000
 18211 100514 000000
 18212 100516 000000

:ENTRY 73 - INSTRUCTION UNDER TEST = CMPN

```

:ICMPN3: .WORD 14          ;INST = CMPN
          .WORD 1          ;TYPE = 1
          .WORD T211       ;IP1 - SRC1.LEN
          .WORD T2         ;IP2 - SRC1.ADR
          .WORD T212       ;IP3 - SRC2.LEN
          .WORD T46        ;IP4 - SRC2.ADR
          .WORD T5         ;IP5 - UNUSED PORTION OF REG. 4
          .WORD T47        ;IP6 - SRC1.DATA
          .WORD T20        ;IP7 - SRC1.SURR.DATA
          .WORD T0         ;IP10 - SRC1.SURR.LEN
          .WORD T47        ;IP11 - SRC2.DATA
          .WORD T7         ;IP12 - SRC2.SURR.DATA
          .WORD T10        ;IP13 - SRC2.SURR.LEN
          .WORD T14        ;IP14 - SEPARATION CONSTANT
          .WORD T0         ;IP15 - SPECIAL HANDLING
          .WORD 0          ;IP16
          .WORD 0          ;IP17
          .WORD 0          ;IP20
          .WORD 0          ;IP21
          .WORD 0          ;IP22
          .WORD 0          ;IP23
          .WORD 0          ;IP24
  
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
:                   - STRINGS ADJACENT
:                   - STRINGS PARTIALLY OVERLAP
:                   - STRINGS COMPLETELY OVERLAP
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 8; SIGN +; HIGH NIBBLE = 17
:                 - ALL DIGITS IDENTICAL = 0; SIGN +; HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 8
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 8; SIGN +; HIGH NIBBLE = 17
:                 - ALL DIGITS IDENTICAL = 0; SIGN +; HIGH NIBBLE = 1
:                 - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 8
  
```

```

: TOTAL # OF TEST CONDITIONS = 256
: TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE) = 832
:
  
```

18213
 18214
 18215
 18216
 18217
 18218
 18219
 18220
 18221
 18222
 18223
 18224
 18225
 18226
 18227
 18228
 18229
 18230
 18231
 18232
 18233
 18234

18236
 18237
 18238 100520 000014
 18239 100522 000003
 18240 100524 102462
 18241 100526 103022
 18242 100530 102462
 18243 100532 107004
 18244 100534 103704
 18245 100536 106340
 18246 100540 102436
 18247 100542 102436
 18248 100544 106340
 18249 100546 102436
 18250 100550 102436
 18251 100552 104070
 18252 100554 102706
 18253 100556 000000
 18254 100560 000000
 18255 100562 000000
 18256 100564 000000
 18257 100566 000000
 18258 100570 000000
 18259 100572 000000

:ENTRY 74 - INSTRUCTION UNDER TEST = CMPN

```

:
: CMPN4: .WORD 14          ;INST = CMPN
:         .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
:         .WORD T1A        ;IP1 - SRC1.LEN
:         .WORD T2A        ;IP2 - SRC1.ADR
:         .WORD T1A        ;IP3 - SRC2.LEN
:         .WORD T46A       ;IP4 - SRC2.ADR
:         .WORD T5         ;IP5 - UNUSED PORTION OF REG. 4
:         .WORD TZ19       ;IP6 - SRC1.DATA
:         .WORD T0         ;IP7 - SRC1.SURR.DATA
:         .WORD T0         ;IP10 - SRC1.SURR.LEN
:         .WORD TZ19       ;IP11 - SRC2.DATA
:         .WORD T0         ;IP12 - SRC2.SURR.DATA
:         .WORD T0         ;IP13 - SRC2.SURR.LEN
:         .WORD T14        ;IP14 - SEPARATION CONSTANT
:         .WORD TSPA       ;IP15 - SPECIAL HANDLING
:         .WORD 0          ;IP16
:         .WORD 0          ;IP17
:         .WORD 0          ;IP20
:         .WORD 0          ;IP21
:         .WORD 0          ;IP22
:         .WORD 0          ;IP23
:         .WORD 0          ;IP24

```

18260
 18261
 18262
 18263
 18264
 18265
 18266
 18267
 18268
 18269
 18270
 18271
 18272

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - NO OVERLAP OF STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +

```

```

: TOTAL # OF TEST CONDITIONS = 64
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)64 = 64
:

```

18274
18275
18276
18277 100574 000030
18278 100576 000001
18279 100600 102712
18280 100602 103012
18281 100604 107310
18282 100606 102732
18283 100610 105634
18284 100612 106072
18285 100614 105050
18286 100616 102436
18287 100620 104020
18288 100622 104036
18289 100624 104054
18290 100626 106546
18291 100630 102436
18292 100632 000000
18293 100634 000000
18294 100636 000000
18295 100640 000000
18296 100642 000000
18297 100644 000000
18298 100646 000000
18299
18300
18301
18302
18303
18304
18305
18306
18307
18308
18309
18310
18311
18312
18313
18314
18315
18316
18317
18318
18319
18320
18321

```

.SBTTL          ASHP TABLES
:ENTRY 75 - INSTRUCTION UNDER TEST = ASHP
:
:ASHP:  .WORD 30          :INST=ASHP
        .WORD 1          :TYPE = 1
        .WORD T111       :IP1 - SRC.LEN
        .WORD T2         :IP2 - SRC.ADR
        .WORD T50        :IP3 - RND.DGT,SHFT.CNT
        .WORD T112       :IP4 - DST.LEN
        .WORD T34        :IP5 - DST.ADR
        .WORD T35B       :IP6 - SRC DATA
        .WORD T20        :IP7 - SRC SURR DATA
        .WORD T0         :IP10 - SRC SURR LEN
        .WORD T11        :IP11 - DST DATA
        .WORD T12        :IP12 - DST SURR DATA
        .WORD T13        :IP13 - DST SURR LEN
        .WORD T43        :IP14 - SEP CONST
        .WORD T0         :IP15 - SPECIAL HANDLING
        .WORD 0          :IP16
        .WORD 0          :IP17
        .WORD 0          :IP20
        .WORD 0          :IP21
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24

```

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE LENGTH - 0,1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE ADDRESS - 200
: DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
:                       - STRINGS ADJACENT
: SOURCE DATA - ALL DIGITS IDENTICAL = 8; SIGN -
:               - ALL DIGITS IDENTICAL = 0; SIGN -
:               - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:               - DIGITS FROM STRING = 000888-
:               - DIGITS FROM STRING = 40000000000000000000000000000000-
: ROUND DIGIT, SHIFT COUNT - 0,0
:                               - 5,-2
:                               - 9,2
:                               - 3,-3
:                               - 1,-3
:                               - 0,5

```

```

: TOTAL # OF TEST CONDITIONS = 540
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)540 = 1620
:

```

18323
18324
18325 100650 000030
18326 100652 000001
18327 100654 102462
18328 100656 103022
18329 100660 107354
18330 100662 102462
18331 100664 105602
18332 100666 106000
18333 100670 102436
18334 100672 102436
18335 100674 102436
18336 100676 102436
18337 100700 102436
18338 100702 104070
18339 100704 102706
18340 100706 000000
18341 100710 000000
18342 100712 000000
18343 100714 000000
18344 100716 000000
18345 100720 000000
18346 100722 000000
18347
18348
18349
18350
18351
18352
18353
18354
18355
18356
18357
18358
18359
18360

:ENTRY 76 - INSTRUCTION UNDER TEST = ASHP

```
ASHP1: .WORD 30          ;INST=ASHP
        .WORD 1          ;TYPE = 1
        .WORD T1A       ;IP1 - SRC.LEN
        .WORD T2A       ;IP2 - SRC.ADR
        .WORD T50C      ;IP3 - RND.DGT,SHFT.CNT
        .WORD T1A       ;IP4 - DST.LEN
        .WORD T34A      ;IP5 - DST.ADR
        .WORD TP19      ;IP6 - SRC DATA
        .WORD T0        ;IP7 - SRC SURR DATA
        .WORD T0        ;IP10 - SRC SURR LEN
        .WORD T0        ;IP11 - DST DATA
        .WORD T0        ;IP12 - DST SURR DATA
        .WORD T0        ;IP13 - DST SURR LEN
        .WORD T14       ;IP14 - SEP CONST
        .WORD TSPA      ;IP15 - SPECIAL HANDLING
        .WORD 0         ;IP16
        .WORD 0         ;IP17
        .WORD 0         ;IP20
        .WORD 0         ;IP21
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
```

:THIS TABLE EXERCISE ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
.....
SOURCE LENGTH - 0,1,2,3,4,5,11,20
DESTINATION LENGTH - 0,1,2,3,4,5,11,20
SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
DESTINATION ADDRESS - NO OVERLAP
SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
ROUND DIGIT, SHIFT COUNT - 7,-1
                           7,-2
                           1, 0
                           8, 3
```

:TOTAL # OF TESTS = 256

18362
18363
18364 100724 000030
18365 100726 000003
18366 100730 102550
18367 100732 103022
18368 100734 107370
18369 100736 102606
18370 100740 105602
18371 100742 106064
18372 100744 102436
18373 100746 102436
18374 100750 102436
18375 100752 102436
18376 100754 102436
18377 100756 104070
18378 100760 102436
18379 100762 000000
18380 100764 000000
18381 100766 000000
18382 100770 000000
18383 100772 000000
18384 100774 000000
18385 100776 000000
18386
18387
18388
18389
18390
18391
18392
18393
18394
18395
18396
18397
18398

:ENTRY 76A - INSTRUCTION UNDER TEST = ASHP

```

:ASHP2: .WORD 30          ;INST=ASHP
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1F       ;IP1 - SRC.LEN
        .WORD T2A       ;IP2 - SRC.ADR
        .WORD T50D      ;IP3 - RND.DGT,SHFT.CNT
        .WORD T1K       ;IP4 - DST.LEN
        .WORD T34A      ;IP5 - DST.ADR
        .WORD TP99      ;IP6 - SRC DATA
        .WORD T0        ;IP7 - SRC SURR DATA
        .WORD T0        ;IP10 - SRC SURR LEN
        .WORD T0        ;IP11 - DST DATA
        .WORD T0        ;IP12 - DST SURR DATA
        .WORD T0        ;IP13 - DST SURR LEN
        .WORD T14       ;IP14 - SEP CONST
        .WORD T0        ;IP15 - SPECIAL HANDLING
        .WORD 0         ;IP16
        .WORD 0         ;IP17
        .WORD 0         ;IP20
        .WORD 0         ;IP21
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24

```

:THIS TABLE EXERCISE ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE LENGTH - 20,16,17,3
: DESTINATION LENGTH - 20,16,17,1
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP
: SOURCE DATA - ALL DIGITS IDENTICAL = 9; SIGN -
: ROUND DIGIT, SHIFT COUNT - 2,-2
:                               9,-3

```

```

:TOTAL # OF TEST CONDITIONS = 32
:TOTAL # OF TESTS = (2 DATA TYPES + 1 IN-LINE) 32 = 96

```

18400
18401
18402
18403 101000 000020
18404 101002 000001
18405 101004 102712
18406 101006 103012
18407 101010 107310
18408 101012 102732
18409 101014 106170
18410 101016 106350
18411 101020 105050
18412 101022 102436
18413 101024 104020
18414 101026 104036
18415 101030 104054
18416 101032 106546
18417 101034 102436
18418 101036 000000
18419 101040 000000
18420 101042 000000
18421 101044 000000
18422 101046 000000
18423 101050 000000
18424 101052 000000

```

.SBTTL          ASHN TABLES
:ENTRY 77 - INSTRUCTION UNDER TEST = ASHN
:
:ASHN:  .WORD 20          :INST=ASHN
        .WORD 1          :TYPE = 1
        .WORD T111       :IP1 - SRC.LEN
        .WORD T2         :IP2 - SRC.ADR
        .WORD T50        :IP3 - RND.DGT,SHFT.CNT
        .WORD T112       :IP4 - DST.LEN
        .WORD T36        :IP5 - DST.ADR
        .WORD T37B       :IP6 - SRC DATA
        .WORD T20        :IP7 - SRC SURR DATA
        .WORD T0         :IP10 - SRC SURR LEN
        .WORD T11        :IP11 - DST DATA
        .WORD T12        :IP12 - DST SURR DATA
        .WORD T13        :IP13 - DST SURR LEN
        .WORD T43        :IP14 - SEP CONST
        .WORD T0         :IP15 - SPECIAL HANDLING
        .WORD 0          :IP16
        .WORD 0          :IP17
        .WORD 0          :IP20
        .WORD 0          :IP21
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24

```

18425
18426
18427
18428
18429
18430
18431
18432
18433
18434
18435
18436
18437
18438
18439
18440
18441
18442
18443
18444
18445
18446
18447

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE LENGTH - 0,1,37
:DESTINATION LENGTH - 0,1,37
:SOURCE ADDRESS - 200
:DESTINATION ADDRESS - NO OVERLAP OF SOURCE & DESTINATION STRINGS
:                      - STRINGS ADJACENT
:SOURCE DATA - ALL DIGITS IDENTICAL = 8; SIGN -; HIGH NIBBLE = 8
:              - ALL DIGITS IDENTICAL = 0; SIGN -; HIGH NIBBLE = 1
:              - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:              - DIGITS FROM STRING = 000888-
:              - DIGITS FROM STRING = 40000000000000000000000000000000-
:ROUND DIGIT, SHIFT COUNT - 0,0
:                          - 5,-2
:                          - 9,2
:                          - 3,-3
:                          - 1,-3
:                          - 0,5
:
:TOTAL # OF TEST CONDITIONS = 540
:TOTAL # OF TESTS = (6 DATA TYPES + 1 INLINE)540 = 3780
:

```


18449
18450
18451 101054 000020
18452 101056 000003
18453 101060 102462
18454 101062 103022
18455 101064 107340
18456 101066 102462
18457 101070 106142
18458 101072 106340
18459 101074 102436
18460 101076 102436
18461 101100 102436
18462 101102 102436
18463 101104 102436
18464 101106 104070
18465 101110 102706
18466 101112 000000
18467 101114 000000
18468 101116 000000
18469 101120 000000
18470 101122 000000
18471 101124 000000
18472 101126 000000
18473
18474
18475
18476
18477
18478
18479
18480
18481
18482
18483

:ENTRY 78 - INSTRUCTION UNDER TEST = ASHN

```
ASHN1: .WORD 20          ;INST=ASHN
        .WORD 3          ;TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1A       ;IP1 - SRC.LEN
        .WORD T2A       ;IP2 - SRC.ADR
        .WORD T50A      ;IP3 - RND.DGT,SHFT.CNT
        .WORD T1A       ;IP4 - DST.LEN
        .WORD T36A      ;IP5 - DST.ADR
        .WORD TZ19      ;IP6 - SRC DATA
        .WORD T0        ;IP7 - SRC SURR DATA
        .WORD T0        ;IP10 - SRC SURR LEN
        .WORD T0        ;IP11 - DST DATA
        .WORD T0        ;IP12 - DST SURR DATA
        .WORD T0        ;IP13 - DST SURR LEN
        .WORD T14       ;IP14 - SEP CONST
        .WORD TSPA      ;IP15 - SPECIAL HANDLING
        .WORD 0         ;IP16
        .WORD 0         ;IP17
        .WORD 0         ;IP20
        .WORD 0         ;IP21
        .WORD 0         ;IP22
        .WORD 0         ;IP23
        .WORD 0         ;IP24
```

:THIS TABLE EXERCISE ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```
:
: SOURCE LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP
: SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: ROUND DIGIT, SHIFT COUNT - 5,-1
```

:TOTAL # OF TESTS = 64

18485
18486
18487 101130 000020
18488 101132 000003
18489 101134 102532
18490 101136 103022
18491 101140 107344
18492 101142 102542
18493 101144 106142
18494 101146 106340
18495 101150 102436
18496 101152 102436
18497 101154 102436
18498 101156 102436
18499 101160 102436
18500 101162 104070
18501 101164 102436
18502 101166 000000
18503 101170 000000
18504 101172 000000
18505 101174 000000
18506 101176 000000
18507 101200 000000
18508 101202 000000

```
:ENTRY 78A - INSTRUCTION UNDER TEST = ASHN
:
:ASHN2: .WORD 20          :INST=ASHN
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1D       :IP1 - SRC.LEN
        .WORD T2A       :IP2 - SRC.ADR
        .WORD T50B      :IP3 - RND.DGT,SHFT.CNT
        .WORD T1E       :IP4 - DST.LEN
        .WORD T36A      :IP5 - DST.ADR
        .WORD TZ19      :IP6 - SRC DATA
        .WORD T0        :IP7 - SRC SURR DATA
        .WORD T0        :IP10 - SRC SURR LEN
        .WORD T0        :IP11 - DST DATA
        .WORD T0        :IP12 - DST SURR DATA
        .WORD T0        :IP13 - DST SURR LEN
        .WORD T14       :IP14 - SEP CONST
        .WORD T0        :IP15 - SPECIAL HANDLING
        .WORD 0         :IP16
        .WORD 0         :IP17
        .WORD 0         :IP20
        .WORD 0         :IP21
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24
```

18509
18510
18511
18512
18513
18514
18515
18516
18517
18518
18519
18520
18521

```
: THIS TABLE EXERCISE ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE LENGTH - 1,2
: DESTINATION LENGTH - 2,3
: SOURCE ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: DESTINATION ADDRESS - NO OVERLAP
: SOURCE DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: ROUND DIGIT, SHIFT COUNT - 7,-1
:                               7, 2
:
: TOTAL # OF TEST CONDITIONS = 8
: TOTAL # OF TESTS (6 DATA TYPES + 1 IN-LINE)8 = 56
```

```

18523      .SBTTL      MULP TABLES
18524      :ENTRY 79 - INSTRUCTION UNDER TEST = MULP
18525      :
18526      101204 000026      :IMULP: .WORD 26      :INST=MULP
18527      101206 000003      :.WORD 3      :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
18528      101210 102436      :.WORD T0      :IP1 - SRC1.LEN
18529      101212 103012      :.WORD T2      :IP2 - SRC1.ADR
18530      101214 102436      :.WORD T0      :IP3 - SRC2.LEN
18531      101216 104442      :.WORD T161     :IP4 - SRC2.ADR
18532      101220 102442      :.WORD XT1      :IP5 - DST.LEN
18533      101222 102436      :.WORD T0      :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
18534      101224 105176      :.WORD T22      :IP7 - SRC1 DATA
18535      101226 103766      :.WORD T7       :IP10 - SRC1 SURR DATA
18536      101230 104004      :.WORD T10      :IP11 - SRC1 SURR LEN
18537      101232 105176      :.WORD T22      :IP12 - SRC2 DATA
18538      101234 105050      :.WORD T20      :IP13 - SRC2 SURR DATA
18539      101236 102436      :.WORD T0       :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
18540      :          :AS NOT TO DESTROY ANY OF SRC1)
18541      101240 104020      :.WORD T11      :IP15 - DST DATA
18542      101242 104036      :.WORD T12      :IP16 - DST SURR DATA
18543      101244 104054      :.WORD T13      :IP17 - DST SURR LEN
18544      101246 104070      :.WORD T14      :IP20 - SEPARATION CONSTRAANT
18545      101250 102436      :.WORD T0       :IP21 -SPECIAL HANDLING
18546      101252 000000      :.WORD 0        :IP22
18547      101254 000000      :.WORD 0        :IP23
18548      101256 000000      :.WORD 0        :IP24
    
```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE 1 LENGTH - 0
:SOURCE 2 LENGTH - 0
:DESTINATION LENGTH - 0,1,5
:SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ADJACENT WITH DESTINATION STRING
:SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 6
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)6 = 18
:
    
```

18549
18550
18551
18552
18553
18554
18555
18556
18557
18558
18559
18560
18561
18562
18563

18565
18566
18567
18568
18569 101260 000026
18570 101262 000001
18571 101264 102436
18572 101266 103012
18573 101270 105066
18574 101272 104442
18575 101274 102442
18576 101276 102436
18577 101300 105176
18578 101302 103766
18579 101304 104004
18580 101306 104774
18581 101310 105050
18582 101312 102436
18583
18584 101314 104020
18585 101316 104036
18586 101320 104054
18587 101322 104070
18588 101324 102436
18589 101326 000000
18590 101330 000000
18591 101332 000000
18592
18593
18594
18595
18596
18597
18598
18599
18600
18601
18602
18603
18604
18605
18606
18607
18608
18609

```

:ENTRY 80 - INSTRUCTION UNDER TEST = MULP
:MULP1: .WORD 26 ;INST=MULP
        .WORD 1 ;TYPE = 1
        .WORD T0 ;IP1 - SRC1.LEN
        .WORD T2 ;IP2 - SRC1.ADR
        .WORD T21 ;IP3 - SRC2.LEN
        .WORD T161 ;IP4 - SRC2.ADR
        .WORD XT1 ;IP5 - DST.LEN
        .WORD T0 ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22 ;IP7 - SRC1 DATA
        .WORD T7 ;IP10 - SRC1 SURR DATA
        .WORD T10 ;IP11 - SRC1 SURR LEN
        .WORD T17 ;IP12 - SRC2 DATA
        .WORD T20 ;IP13 - SRC2 SURR DATA
        .WORD T0 ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11 ;IP15 - DST DATA
        .WORD T12 ;IP16 - DST SURR DATA
        .WORD T13 ;IP17 - DST SURR LEN
        .WORD T14 ;IP20 - SEPARATION CONSTRAINT
        .WORD T0 ;IP21 -SPECIAL HANDLING
        .WORD 0 ;IP22
        .WORD 0 ;IP23
        .WORD 0 ;IP24
    
```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE 1 LENGTH - 0
:SOURCE 2 LENGTH - 1,5
:DESTINATION LENGTH - 0,1,5
:SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ADJACENT WITH DESTINATION STRIN
:SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:SOURCE 2 DATA - DIGTIS FROM STRING = 1234567891234567891234000891233; SIGN +
:               - ALL DIGITS IDENTICAL = 5; SIGN +
:               - ALL DIGITS IDENTICAL =3; SIGN -
:               - ALL DIGITS IDENTICAL = 0; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 48
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144
:
    
```

18611
18612
18613 101334 000026
18614 101336 000001
18615 101340 105066
18616 101342 103012
18617 101344 102436
18618 101346 104442
18619 101350 102442
18620 101352 102436
18621 101354 104774
18622 101356 103766
18623 101360 104004
18624 101362 105176
18625 101364 105050
18626 101366 102436
18627
18628 101370 104020
18629 101372 104036
18630 101374 104054
18631 101376 104070
18632 101400 102436
18633 101402 000000
18634 101404 000000
18635 101406 000000
18636
18637
18638
18639
18640
18641
18642
18643
18644
18645
18646
18647
18648
18649
18650
18651
18652
18653

:ENTRY 81 - INSTRUCTION UNDER TEST = MULP

```

:IMULP2: .WORD 26          ;INST=MULP
          .WORD 1          ;TYPE = 1
          .WORD T21       ;IP1 - SRC1.LEN
          .WORD T2        ;IP2 - SRC1.ADR
          .WORD T0        ;IP3 - SRC2.LEN
          .WORD T161      ;IP4 - SRC2.ADR
          .WORD XT1       ;IP5 - DST.LEN
          .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD T17       ;IP7 - SRC1 DATA
          .WORD T7        ;IP10 - SRC1 SURR DATA
          .WORD T10       ;IP11 - SRC1 SURR LEN
          .WORD T22       ;IP12 - SRC2 DATA
          .WORD T20       ;IP13 - SRC2 SURR DATA
          .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                          ; AS NOT TO DESTROY ANY OF SRC1)
          .WORD T11      ;IP15 - DST DATA
          .WORD T12      ;IP16 - DST SURR DATA
          .WORD T13      ;IP17 - DST SURR LEN
          .WORD T14      ;IP20 - SEPARATION CONSTRAINT
          .WORD T0        ;IP21 - SPECIAL HANDLING
          .WORD 0         ;IP22
          .WORD 0         ;IP23
          .WORD 0         ;IP24
    
```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ADJACENT WITH DESTINATION STRIN
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                               - ALL DIGITS IDENTICAL = 5; SIGN +
:                               - ALL DIGITS IDENTICAL = 3; SIGN -
:                               - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:
: TOTAL # OF TEST CONDITIONS =48
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144
:
    
```

18655
18656
18657 101410 000026
18658 101412 000001
18659 101414 105102
18660 101416 103012
18661 101420 105120
18662 101422 104442
18663 101424 102642
18664 101426 102436
18665 101430 104774
18666 101432 103766
18667 101434 104004
18668 101436 104774
18669 101440 105050
18670 101442 102436
18671
18672 101444 104020
18673 101446 104036
18674 101450 104054
18675 101452 104070
18676 101454 102436
18677 101456 000000
18678 101460 000000
18679 101462 000000
18680
18681
18682
18683
18684
18685
18686
18687
18688
18689
18690
18691
18692
18693
18694
18695
18696
18697
18698
18699
18700

:ENTRY 82 - INSTRUCTION UNDER TEST = MULP

```

:IMULP3: .WORD 26          ;INST=MULP
          .WORD 1          ;TYPE = 1
          .WORD T211       ;IP1 - SRC1.LEN
          .WORD T2         ;IP2 - SRC1.ADR
          .WORD T212       ;IP3 - SRC2.LEN
          .WORD T161       ;IP4 - SRC2.ADR
          .WORD T1P        ;IP5 - DST.LEN
          .WORD T0         ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD T17       ;IP7 - SRC1 DATA
          .WORD T7         ;IP10 - SRC1 SURR DATA
          .WORD T10        ;IP11 - SRC1 SURR LEN
          .WORD T17        ;IP12 - SRC2 DATA
          .WORD T20        ;IP13 - SRC2 SURR DATA
          .WORD T0         ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
                          ; AS NOT TO DESTROY ANY OF SRC1)
          .WORD T11        ;IP15 - DST DATA
          .WORD T12        ;IP16 - DST SURR DATA
          .WORD T13        ;IP17 - DST SURR LEN
          .WORD T14        ;IP20 - SEPARATION CONSTRAINT
          .WORD T0         ;IP21 - SPECIAL HANDLING
          .WORD 0          ;IP22
          .WORD 0          ;IP23
          .WORD 0          ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,37
: SOURCE 2 LENGTH - 1,37
: DESTINATION LENGTH - 0,1,37
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ADJACENT WITH DEST STRING
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                   - ALL DIGITS IDENTICAL = 5; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                   - ALL DIGITS IDENTICAL = 5; SIGN +
:                   - ALL DIGITS IDENTICAL = 3; SIGN -
:                   - ALL DIGITS IDENTICAL = 0; SIGN +

```

```

: TOTAL # OF TEST CONDITIONS = 384
: TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)384 = 1152
:

```

18702
18703
18704 101464 000026
18705 101466 000001
18706 101470 102462
18707 101472 103022
18708 101474 102462
18709 101476 104142
18710 101500 102462
18711 101502 102436
18712 101504 106000
18713 101506 102436
18714 101510 102436
18715 101512 106000
18716 101514 102436
18717 101516 102436
18718
18719 101520 102436
18720 101522 102436
18721 101524 102436
18722 101526 104070
18723 101530 102706
18724 101532 000000
18725 101534 000000
18726 101536 000000
18727
18728
18729
18730
18731
18732
18733
18734
18735
18736
18737
18738
18739
18740

```
:ENTRY 83 - INSTRUCTION UNDER TEST = MULP
:
:IMULP4: .WORD 26          :INST=MULP
          .WORD 1          :TYPE = 1
          .WORD T1A        :IP1 - SRC1.LEN
          .WORD T2A        :IP2 - SRC1.ADR
          .WORD T1A        :IP3 - SRC2.LEN
          .WORD T16A       :IP4 - SRC2.ADR
          .WORD T1A        :IP5 - DST.LEN
          .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD TP19       :IP7 - SRC1 DATA
          .WORD T0         :IP10 - SRC1 SURR DATA
          .WORD T0         :IP11 - SRC1 SURR LEN
          .WORD TP19       :IP12 - SRC2 DATA
          .WORD T0         :IP13 - SRC2 SURR DATA
          .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
          : AS NOT TO DESTROY ANY OF SRC1)
          .WORD T0         :IP15 - DST DATA
          .WORD T0         :IP16 - DST SURR DATA
          .WORD T0         :IP17 - DST SURR LEN
          .WORD T14        :IP20 - SEPARATION CONSTRAINT
          .WORD TSPA       :IP21 - SPECIAL HANDLING
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24
```

```
: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 512
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
:
```

```

18742
18743
18744 101540 000026
18745 101542 000003
18746 101544 102572
18747 101546 103012
18748 101550 102572
18749 101552 104142
18750 101554 102620
18751 101556 102436
18752 101560 106000
18753 101562 102436
18754 101564 102436
18755 101566 106000
18756 101570 102436
18757 101572 102436
18758
18759 101574 102436
18760 101576 102436
18761 101600 102436
18762 101602 104070
18763 101604 102706
18764 101606 000000
18765 101610 000000
18766 101612 000000
18767
18768
18769
18770
18771
18772
18773
18774
18775
18776
18777
18778
18779
18780

```

:ENTRY 83A - INSTRUCTION UNDER TEST = MULP

```

:IMULP5: .WORD 26          :INST=MULP
          .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1H        :IP1 - SRC1.LEN
          .WORD T2         :IP2 - SRC1.ADR
          .WORD T1H        :IP3 - SRC2.LEN
          .WORD T16A       :IP4 - SRC2.ADR
          .WORD T1L        :IP5 - DST.LEN
          .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD TP19       :IP7 - SRC1 DATA
          .WORD T0         :IP10 - SRC1 SURR DATA
          .WORD T0         :IP11 - SRC1 SURR LEN
          .WORD TP19       :IP12 - SRC2 DATA
          .WORD T0         :IP13 - SRC2 SURR DATA
          .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
          .WORD T0         :AS NOT TO DESTROY ANY OF SRC1)
          .WORD T0         :IP15 - DST DATA
          .WORD T0         :IP16 - DST SURR DATA
          .WORD T0         :IP17 - DST SURR LEN
          .WORD T14        :IP20 - SEPARATION CONSTRAINT
          .WORD TSPA       :IP21 - SPECIAL HANDLING
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 12,13,14,15
: SOURCE 2 LENGTH - 12,13,14,15
: DESTINATION LENGTH - 35
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:

```

```

:TOTAL # OF TEST CONDITIONS = 16
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)16 = 16
:

```


18782
18783
18784 101614 000026
18785 101616 000003
18786 101620 102634
18787 101622 103012
18788 101624 102662
18789 101626 104142
18790 101630 102620
18791 101632 102436
18792 101634 106036
18793 101636 102436
18794 101640 102436
18795 101642 106050
18796 101644 102436
18797 101646 102436
18798
18799 101650 102436
18800 101652 102436
18801 101654 102436
18802 101656 104070
18803 101660 102706
18804 101662 000000
18805 101664 000000
18806 101666 000000
18807
18808
18809
18810
18811
18812
18813
18814
18815
18816
18817
18818
18819
18820

```

:ENTRY 83B - INSTRUCTION UNDER TEST = MULP
:
:IMULP6: .WORD 26          :INST=MULP
          .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
          .WORD T1N        :IP1 - SRC1.LEN
          .WORD T2         :IP2 - SRC1.ADR
          .WORD T1Q        :IP3 - SRC2.LEN
          .WORD T16A       :IP4 - SRC2.ADR
          .WORD T1L        :IP5 - DST.LEN
          .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD TP19C      :IP7 - SRC1 DATA
          .WORD T0         :IP10 - SRC1 SURR DATA
          .WORD T0         :IP11 - SRC1 SURR LEN
          .WORD TP19D      :IP12 - SRC2 DATA
          .WORD T0         :IP13 - SRC2 SURR DATA
          .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
          : AS NOT TO DESTROY ANY OF SRC1)
          .WORD T0         :IP15 - DST DATA
          .WORD T0         :IP16 - DST SURR DATA
          .WORD T0         :IP17 - DST SURR LEN
          .WORD T14        :IP20 - SEPARATION CONSTRAINT
          .WORD TSPA       :IP21 - SPECIAL HANDLING
          .WORD 0          :IP22
          .WORD 0          :IP23
          .WORD 0          :IP24

```

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
:SOURCE 1 LENGTH - 3
:SOURCE 2 LENGTH - 0,3
:DESTINATION LENGTH - 35
:SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
:SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:SOURCE 1 DATA - ALL DIGITS IDENTICAL = 0; SIGN +
:                  ALL DIGITS IDENTICAL = 4; SIGN +
:SOURCE 2 DATA - ALL DIGITS IDENTICAL = 0; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 4
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)4 = 4

```

18822
18823
18824
18825 101670 000026
18826 101672 000001
18827 101674 102626
18828 101676 103012
18829 101700 102626
18830 101702 104220
18831 101704 104054
18832 101706 102436
18833 101710 106026
18834 101712 102436
18835 101714 102436
18836 101716 106026
18837 101720 102436
18838 101722 102436
18839
18840 101724 102436
18841 101726 102436
18842 101730 102436
18843 101732 102436
18844 101734 102436
18845 101736 000000
18846 101740 000000
18847 101742 000000
18848
18849
18850
18851
18852
18853
18854
18855
18856
18857
18858
18859
18860
18861

: ENTRY 83C - INSTRUCTION UNDER TEST = MULP

:IMULP7: .WORD 26 :INST=MULP
: .WORD 1 :TYPE = 1
: .WORD T1M :IP1 - SRC1.LEN
: .WORD T2 :IP2 - SRC1.ADR
: .WORD T1M :IP3 - SRC2.LEN
: .WORD T16B :IP4 - SRC2.ADR
: .WORD T13 :IP5 - DST.LEN
: .WORD T0 :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
: .WORD TP19B :IP7 - SRC1 DATA
: .WORD T0 :IP10 - SRC1 SURR DATA
: .WORD T0 :IP11 - SRC1 SURR LEN
: .WORD TP19B :IP12 - SRC2 DATA
: .WORD T0 :IP13 - SRC2 SURR DATA
: .WORD T0 :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
: AS NOT TO DESTROY ANY OF SRC1)
: .WORD T0 :IP15 - DST DATA
: .WORD T0 :IP16 - DST SURR DATA
: .WORD T0 :IP17 - DST SURR LEN
: .WORD T0 :IP20 - SEPARATION CONSTRAINT
: .WORD 0 :IP21 - SPECIAL HANDLING
: .WORD 0 :IP22
: .WORD 0 :IP23
: .WORD 0 :IP24

: THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

: SOURCE 1 LENGTH - 17
: SOURCE 2 LENGTH - 17
: DESTINATION LENGTH - 5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS - 200
: DEST ADDRESS - NO OVERLAP WITH SOURCE STRINGS
: SOURCE 1 DATA = SOURCE 2 DATA - 000000000000333+

: TOTAL # OF TEST CONDITIONS = 1
: TOTAL # OF TESTS = (2 DATA TYPES + 1 IN-LINE) 1 = 3

18863
18864
18865
18866 101744 000027
18867 101746 000003
18868 101750 102436
18869 101752 103012
18870 101754 102436
18871 101756 104442
18872 101760 102442
18873 101762 102436
18874 101764 105176
18875 101766 103766
18876 101770 104004
18877 101772 105176
18878 101774 105050
18879 101776 102436
18880
18881 102000 104020
18882 102002 104036
18883 102004 104054
18884 102006 104070
18885 102010 102436
18886 102012 000000
18887 102014 000000
18888 102016 000000
18889
18890
18891
18892
18893
18894
18895
18896
18897
18898
18899
18900
18901
18902
18903

```
.SBTTL          DIVP TABLES
:ENTRY 84 - INSTRUCTION UNDER TEST = DIVP
:
:DIVP:  .WORD 27          :INST=DIVP
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T0         :IP1 - SRC1.LEN
        .WORD T2         :IP2 - SRC1.ADR
        .WORD T0         :IP3 - SRC2.LEN
        .WORD T161       :IP4 - SRC2.ADR
        .WORD XT1        :IP5 - DST.LEN
        .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22        :IP7 - SRC1 DATA
        .WORD T7         :IP10 - SRC1 SURR DATA
        .WORD T10        :IP11 - SRC1 SURR LEN
        .WORD T22        :IP12 - SRC2 DATA
        .WORD T20        :IP13 - SRC2 SURR DATA
        .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11        :IP15 - DST DATA
        .WORD T12        :IP16 - DST SURR DATA
        .WORD T13        :IP17 - DST SURR LEN
        .WORD T14        :IP20 - SEPARATION CONSTRAINT
        .WORD T0         :IP21 - SPECIAL HANDLING
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
```

```
:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE 1 LENGTH - 0
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                               - SOURCE 2 STRING ADJACENT WITH DESTINATION STRING
:
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 6
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)6 = 18
:
```

18905
18906
18907 102020 000027
18908 102022 000001
18909 102024 102436
18910 102026 103012
18911 102030 105066
18912 102032 104442
18913 102034 102442
18914 102036 102436
18915 102040 105176
18916 102042 103766
18917 102044 104004
18918 102046 104774
18919 102050 105050
18920 102052 102436
18921
18922 102054 104020
18923 102056 104036
18924 102060 104054
18925 102062 104070
18926 102064 102436
18927 102066 000000
18928 102070 000000
18929 102072 000000
18930
18931
18932
18933
18934
18935
18936
18937
18938
18939
18940
18941
18942
18943
18944
18945
18946
18947

:ENTRY 85 - INSTRUCTION UNDER TEST = DIVP

```

:DIVP1: .WORD 27          :INST=DIVP
        .WORD 1          :TYPE = 1
        .WORD T0         :IP1 - SRC1.LEN
        .WORD T2         :IP2 - SRC1.ADR
        .WORD T21        :IP3 - SRC2.LEN
        .WORD T161       :IP4 - SRC2.ADR
        .WORD XT1        :IP5 - DST.LEN
        .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T22        :IP7 - SRC1 DATA
        .WORD T7         :IP10 - SRC1 SURR DATA
        .WORD T10        :IP11 - SRC1 SURR LEN
        .WORD T17        :IP12 - SRC2 DATA
        .WORD T20        :IP13 - SRC2 SURR DATA
        .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11        :IP15 - DST DATA
        .WORD T12        :IP16 - DST SURR DATA
        .WORD T13        :IP17 - DST SURR LEN
        .WORD T14        :IP20 - SEPARATION CONSTRAINT
        .WORD T0         :IP21 - SPECIAL HANDLING
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

- : SOURCE 1 LENGTH - 0
- : SOURCE 2 LENGTH - 1,5
- : DESTINATION LENGTH - 0,1,5
- : SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
- : SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
- : - SOURCE 2 STRING ADJACENT WITH DESTINATION STRIN
- : SOURCE 1 DATA - ALL DIGITS IDENTICAL = 3; SIGN +
- : SOURCE 2 DATA - DIGTIS FROM STRING = 1234567891234567891234000891233; SIGN +
- : - ALL DIGITS IDENTICAL = 5; SIGN +
- : - ALL DIGITS IDENTICAL =3; SIGN -
- : - ALL DIGITS IDENTICAL = 0; SIGN +

```

:TOTAL # OF TEST CONDITIONS = 48
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)48 = 144

```

18949
18950
18951
18952 102074 000027
18953 102076 000001
18954 102100 105066
18955 102102 103012
18956 102104 102436
18957 102106 104442
18958 102110 102442
18959 102112 102436
18960 102114 104774
18961 102116 103766
18962 102120 104004
18963 102122 105176
18964 102124 105050
18965 102126 102436
18966
18967 102130 104020
18968 102132 104036
18969 102134 104054
18970 102136 104070
18971 102140 102436
18972 102142 000000
18973 102144 000000
18974 102146 000000
18975
18976
18977
18978
18979
18980
18981
18982
18983
18984
18985
18986
18987
18988
18989
18990
18991
18992

:ENTRY 86 - INSTRUCTION UNDER TEST = DIVP

```

:DIVP2: .WORD 27          ;INST=DIVP
        .WORD 1          ;TYPE = 1
        .WORD T21        ;IP1 - SRC1.LEN
        .WORD T2         ;IP2 - SRC1.ADR
        .WORD T0         ;IP3 - SRC2.LEN
        .WORD T161       ;IP4 - SRC2.ADR
        .WORD XT1        ;IP5 - DST.LEN
        .WORD T0         ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17        ;IP7 - SRC1 DATA
        .WORD T7         ;IP10 - SRC1 SURR DATA
        .WORD T10        ;IP11 - SRC1 SURR LEN
        .WORD T22        ;IP12 - SRC2 DATA
        .WORD T20        ;IP13 - SRC2 SURR DATA
        .WORD T0         ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11        ;IP15 - DST DATA
        .WORD T12        ;IP16 - DST SURR DATA
        .WORD T13        ;IP17 - DST SURR LEN
        .WORD T14        ;IP20 - SEPARATION CONSTRAINT
        .WORD T0         ;IP21 - SPECIAL HANDLING
        .WORD 0          ;IP22
        .WORD 0          ;IP23
        .WORD 0          ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 1,5
: SOURCE 2 LENGTH - 0
: DESTINATION LENGTH - 0,1,5
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
:                                     - SOURCE 2 STRING ADJACENT WITH DESTINATION STRIN
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:                                     - ALL DIGITS IDENTICAL = 5; SIGN +
:                                     - ALL DIGITS IDENTICAL = 3; SIGN -
:                                     - ALL DIGITS IDENTICAL = 0; SIGN +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 3; SIGN +

```

```

: TOTAL # OF TEST CONDITIONS =48
: TOTAL # OF TESTS = (2 DATA TYPES + 1 !NLINE)48 = 144
:

```

18994
18995
18996 102150 000027
18997 102152 000001
18998 102154 105102
18999 102156 103012
19000 102160 105120
19001 102162 104442
19002 102164 102642
19003 102166 102436
19004 102170 104774
19005 102172 103766
19006 102174 104004
19007 102176 104774
19008 102200 105050
19009 102202 102436
19010
19011 102204 104020
19012 102206 104036
19013 102210 104054
19014 102212 104070
19015 102214 102436
19016 102216 000000
19017 102220 000000
19018 102222 000000
19019
19020
19021
19022
19023
19024
19025
19026
19027
19028
19029
19030
19031
19032
19033
19034
19035
19036
19037
19038
19039

:ENTRY 87 - INSTRUCTION UNDER TEST = DIVP

```

:DIVP3: .WORD 27          ;INST=DIVP
        .WORD 1          ;TYPE = 1
        .WORD T211      ;IP1 - SRC1.LEN
        .WORD T2        ;IP2 - SRC1.ADR
        .WORD T212      ;IP3 - SRC2.LEN
        .WORD T161      ;IP4 - SRC2.ADR
        .WORD T1P       ;IP5 - DST.LEN
        .WORD T0        ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17      ;IP7 - SRC1 DATA
        .WORD T7        ;IP10 - SRC1 SURR DATA
        .WORD T10       ;IP11 - SRC1 SURR LEN
        .WORD T17       ;IP12 - SRC2 DATA
        .WORD T20       ;IP13 - SRC2 SURR DATA
        .WORD T0        ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        ; AS NOT TO DESTROY ANY OF SRC1)
        .WORD T11      ;IP15 - DST DATA
        .WORD T12      ;IP16 - DST SURR DATA
        .WORD T13      ;IP17 - DST SURR LEN
        .WORD T14      ;IP20 - SEPARATION CONSTRAINT
        .WORD T0       ;IP21 - SPECIAL HANDLING
        .WORD 0        ;IP22
        .WORD 0        ;IP23
        .WORD 0        ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

SOURCE 1 LENGTH - 1,37
SOURCE 2 LENGTH - 1,37
DESTINATION LENGTH - 0,1,37
SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
SOURCE 2 ADDRESS, DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
                                - SOURCE 2 STRING ADJACENT WITH DEST STRING
SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
                - ALL DIGITS IDENTICAL = 5; SIGN +
                - ALL DIGITS IDENTICAL = 3; SIGN -
                - ALL DIGITS IDENTICAL = 0; SIGN +
SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
                - ALL DIGITS IDENTICAL = 5; SIGN +
                - ALL DIGITS IDENTICAL = 3; SIGN -
                - ALL DIGITS IDENTICAL = 0; SIGN +

```

```

:TOTAL # OF TEST CONDITIONS = 384
:TOTAL # OF TESTS = (2 DATA TYPES + 1 INLINE)384 = 1152

```

19041
19042
19043 102224 000027
19044 102226 000001
19045 102230 102462
19046 102232 103022
19047 102234 102462
19048 102236 104142
19049 102240 102462
19050 102242 102436
19051 102244 106000
19052 102246 102436
19053 102250 102436
19054 102252 106000
19055 102254 102436
19056 102256 102436
19057
19058 102260 102436
19059 102262 102436
19060 102264 102436
19061 102266 104070
19062 102270 102706
19063 102272 000000
19064 102274 000000
19065 102276 000000
19066
19067
19068
19069
19070
19071
19072
19073
19074
19075
19076
19077
19078
19079

:ENTRY 88 - INSTRUCTION UNDER TEST = DIVP

```

:IDIVP4: .WORD 27      ;INST=DIVP
          .WORD 1      ;TYPE = 1
          .WORD T1A    ;IP1 - SRC1.LEN
          .WORD T2A    ;IP2 - SRC1.ADR
          .WORD T1A    ;IP3 - SRC2.LEN
          .WORD T16A   ;IP4 - SRC2.ADR
          .WORD T1A    ;IP5 - DST.LEN
          .WORD T0     ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
          .WORD TP19   ;IP7 - SRC1 DATA
          .WORD T0     ;IP10 - SRC1 SURR DATA
          .WORD T0     ;IP11 - SRC1 SURR LEN
          .WORD TP19   ;IP12 - SRC2 DATA
          .WORD T0     ;IP13 - SRC2 SURR DATA
          .WORD T0     ;IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
          ; AS NOT TO DESTROY ANY OF SRC1)
          .WORD T0     ;IP15 - DST DATA
          .WORD T0     ;IP16 - DST SURR DATA
          .WORD T0     ;IP17 - DST SURR LEN
          .WORD T14    ;IP20 - SEPARATION CONSTRAINT
          .WORD TSPA   ;IP21 - SPECIAL HANDLING
          .WORD 0      ;IP22
          .WORD 0      ;IP23
          .WORD 0      ;IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 2 LENGTH - 0,1,2,3,4,5,11,20
: DESTINATION LENGTH - 0,1,2,3,4,5,11,20
: SOURCE 1 ADDRESS - 201 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
: SOURCE 2 DATA - DIGITS FROM STRING = 1234567891234567891234000891233; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 512
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)512 = 512
:

```

19081
19082
19083 102300 000027
19084 102302 000003
19085 102304 102564
19086 102306 103012
19087 102310 102564
19088 102312 104142
19089 102314 102564
19090 102316 102436
19091 102320 105034
19092 102322 102436
19093 102324 102436
19094 102326 105034
19095 102330 102436
19096 102332 102436
19097
19098 102334 102436
19099 102336 102436
19100 102340 102436
19101 102342 104070
19102 102344 102706
19103 102346 000000
19104 102350 000000
19105 102352 000000
19106
19107
19108
19109
19110
19111
19112
19113
19114
19115
19116
19117
19118
19119

:ENTRY 88A - INSTRUCTION UNDER TEST = DIVP

```

:DIVP5: .WORD 27          :INST=DIVP
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1G       :IP1 - SRC1.LEN
        .WORD T2        :IP2 - SRC1.ADR
        .WORD T1G       :IP3 - SRC2.LEN
        .WORD T16A      :IP4 - SRC2.ADR
        .WORD T1G       :IP5 - DST.LEN
        .WORD T0        :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T17A      :IP7 - SRC1 DATA
        .WORD T0        :IP10 - SRC1 SURR DATA
        .WORD T0        :IP11 - SRC1 SURR LEN
        .WORD T17A      :IP12 - SRC2 DATA
        .WORD T0        :IP13 - SRC2 SURR DATA
        .WORD T0        :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        : AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0        :IP15 - DST DATA
        .WORD T0        :IP16 - DST SURR DATA
        .WORD T0        :IP17 - DST SURR LEN
        .WORD T14       :IP20 - SEPARATION CONSTRAINT
        .WORD TSPA      :IP21 - SPECIAL HANDLING
        .WORD 0         :IP22
        .WORD 0         :IP23
        .WORD 0         :IP24

```

:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:

```

:
: SOURCE 1 LENGTH - 11
: SOURCE 2 LENGTH - 11
: DESTINATION LENGTH - 11
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 7,0; SIGN = +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 7,0; SIGN +
:
: TOTAL # OF TEST CONDITIONS = 4
: TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)4 = 4
:

```


19121
19122
19123 102354 000027
19124 102356 000003
19125 102360 102672
19126 102362 103012
19127 102364 102700
19128 102366 104142
19129 102370 102672
19130 102372 102436
19131 102374 106056
19132 102376 102436
19133 102400 102436
19134 102402 106056
19135 102404 102436
19136 102406 102436
19137
19138 102410 102436
19139 102412 102436
19140 102414 102436
19141 102416 104070
19142 102420 102706
19143 102422 000000
19144 102424 000000
19145 102426 000000
19146
19147
19148
19149
19150
19151
19152
19153
19154
19155
19156
19157
19158
19159

```
:ENTRY 88B - INSTRUCTION UNDER TEST = DIVP
:
:DIVP6: .WORD 27          :INST=DIVP
        .WORD 3          :TYPE = 1(BIT 0);11/44 QV TABLE(BIT 1=1)
        .WORD T1R        :IP1 - SRC1.LEN
        .WORD T2         :IP2 - SRC1.ADR
        .WORD T1S        :IP3 - SRC2.LEN
        .WORD T16A       :IP4 - SRC2.ADR
        .WORD T1R        :IP5 - DST.LEN
        .WORD T0         :IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD TP19E      :IP7 - SRC1 DATA
        .WORD T0         :IP10 - SRC1 SURR DATA
        .WORD T0         :IP11 - SRC1 SURR LEN
        .WORD TP19E      :IP12 - SRC2 DATA
        .WORD T0         :IP13 - SRC2 SURR DATA
        .WORD T0         :IP14 - SRC2 SURR LEN (LENGTH SET TO 0 SO
        .WORD T0         :AS NOT TO DESTROY ANY OF SRC1)
        .WORD T0         :IP15 - DST DATA
        .WORD T0         :IP16 - DST SURR DATA
        .WORD T0         :IP17 - DST SURR LEN
        .WORD T14        :IP20 - SEPARATION CONSTRAINT
        .WORD TSPA       :IP21 - SPECIAL HANDLING
        .WORD 0          :IP22
        .WORD 0          :IP23
        .WORD 0          :IP24
```

```
:THIS TABLE EXERCISES ALL COMBINATIONS OF THE FOLLOWING VARIABLE ASSIGNMENTS:
:
: SOURCE 1 LENGTH - 11
: SOURCE 2 LENGTH - 21
: DESTINATION LENGTH - 11
: SOURCE 1 ADDRESS - 200 (RELATIVE TO START OF TEST BUFFER)
: SOURCE 2 ADDRESS,DEST ADDRESS - NO OVERLAP OF ANY OF THE 3 STRINGS
: SOURCE 1 DATA - ALL DIGITS IDENTICAL = 7; SIGN = +
: SOURCE 2 DATA - ALL DIGITS IDENTICAL = 7; SIGN +
:
:TOTAL # OF TEST CONDITIONS = 1
:TOTAL # OF TESTS = (1 DATA TYPE IN REG MODE)1 = 1
:
```

19163
19164
19165 102430 000000
19166 102432 000000
19167 102434 000000

:ENTRY 89 -
:
:WORD 0
:WORD 0
:WORD 0

```
19169          .SBTTL PARAMETER TABLES (LENGTHS,ADDRESSES,ETC)
19170          ;
19171          ;PARAMETER TABLES
19172          ;
19174 102436    T0:          ;DUMMY TABLE - USED WHEN AN INPUT PARAMETER
19175          ; IS ALREADY SPECIFIED BY A PRECEDING INPUT PARAMETER T
19176          ; ALSO PROVIDES A SINGLE ENTRY = 0 TABLE
19177 102436 000401      .WORD 401
19178 102440 000000      .WORD 0
19179
19180 102442    XT1:
19181 102442 000403      .WORD 403          ;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES
19182 102444 000000      .WORD 0
19183 102446 000001      .WORD 1
19184 102450 000005      .WORD 5
19185 102452 000000      .WORD 0
19186 102454 000000      .WORD 0
19187 102456 000000      .WORD 0
19188 102460 000000      .WORD 0
19189
19190 102462 000410      T1A: .WORD 410
19191 102464 000000      .WORD 0
19192 102466 000001      .WORD 1
19193 102470 000002      .WORD 2
19194 102472 000003      .WORD 3
19195 102474 000004      .WORD 4
19196 102476 000005      .WORD 5
19197 102500 000011      .WORD 11
19198 102502 000020      .WORD 20
19199 102504 000000      .WORD 0
19200
19201 102506 000403      T1B: .WORD 403
19202 102510 000001      .WORD 1
19203 102512 000002      .WORD 2
19204 102514 000003      .WORD 3
19205 102516 000000      .WORD 0
19206
19207 102520 000403      T1C: .WORD 403
19208 102522 000002      .WORD 2
19209 102524 000004      .WORD 4
19210 102526 000006      .WORD 6
19211 102530 000000      .WORD 0
19212
19213 102532 000402      T1D: .WORD 402
19214 102534 000001      .WORD 1
19215 102536 000002      .WORD 2
19216 102540 000000      .WORD 0
19217
19218 102542 000401      T1E: .WORD 401
19219 102544 000002      .WORD 2
19220 102546 000000      .WORD 0
19221
19222 102550 000404      T1F: .WORD 404
19223 102552 000020      .WORD 20
```

| | | | | |
|-------|--------|--------|------|-----------|
| 19224 | 102554 | 000016 | | .WORD 16 |
| 19225 | 102556 | 000017 | | .WORD 17 |
| 19226 | 102560 | 000003 | | .WORD 3 |
| 19227 | 102562 | 000000 | | .WORD 0 |
| 19228 | | | | |
| 19229 | 102564 | 000401 | T1G: | .WORD 401 |
| 19230 | 102566 | 000011 | | .WORD 11 |
| 19231 | 102570 | 000000 | | .WORD 0 |
| 19232 | | | | |
| 19233 | 102572 | 000404 | T1H: | .WORD 404 |
| 19234 | 102574 | 000012 | | .WORD 12 |
| 19235 | 102576 | 000013 | | .WORD 13 |
| 19236 | 102600 | 000014 | | .WORD 14 |
| 19237 | 102602 | 000015 | | .WORD 15 |
| 19238 | 102604 | 000000 | | .WORD 0 |
| 19239 | | | | |
| 19240 | 102606 | 000404 | T1K: | .WORD 404 |
| 19241 | 102610 | 000020 | | .WORD 20 |
| 19242 | 102612 | 000016 | | .WORD 16 |
| 19243 | 102614 | 000017 | | .WORD 17 |
| 19244 | 102616 | 000001 | | .WORD 1 |
| 19245 | | | | |
| 19246 | 102620 | 000401 | T1L: | .WORD 401 |
| 19247 | 102622 | 000035 | | .WORD 35 |
| 19248 | 102624 | 000000 | | .WORD 0 |
| 19249 | | | | |
| 19250 | 102626 | 000401 | T1M: | .WORD 401 |
| 19251 | 102630 | 000017 | | .WORD 17 |
| 19252 | 102632 | 000000 | | .WORD 0 |
| 19253 | | | | |
| 19254 | 102634 | 000401 | T1N: | .WORD 401 |
| 19255 | 102636 | 000003 | | .WORD 3 |
| 19256 | 102640 | 000000 | | .WORD 0 |
| 19257 | | | | |
| 19258 | 102642 | 000403 | T1P: | .WORD 403 |
| 19259 | 102644 | 000000 | | .WORD 0 |
| 19260 | 102646 | 000001 | | .WORD 1 |
| 19261 | 102650 | 000037 | | .WORD 37 |
| 19262 | 102652 | 000000 | | .WORD 0 |
| 19263 | 102654 | 000000 | | .WORD 0 |
| 19264 | 102656 | 000000 | | .WORD 0 |
| 19265 | 102660 | 000000 | | .WORD 0 |
| 19266 | | | | |
| 19267 | 102662 | 000402 | T1Q: | .WORD 402 |
| 19268 | 102664 | 000000 | | .WORD 0 |
| 19269 | 102666 | 000003 | | .WORD 3 |
| 19270 | 102670 | 000000 | | .WORD 0 |
| 19271 | | | | |
| 19272 | 102672 | 000401 | T1R: | .WORD 401 |
| 19273 | 102674 | 000011 | | .WORD 11 |
| 19274 | 102676 | 000000 | | .WORD 0 |
| 19275 | | | | |
| 19276 | 102700 | 000401 | T1S: | .WORD 401 |
| 19277 | 102702 | 000021 | | .WORD 21 |

| | | | | | |
|-------|--------|--------|-------|--------------|--|
| 19278 | 102704 | 000000 | | .WORD 0 | |
| 19279 | | | | | |
| 19280 | 102706 | | TSPA: | | |
| 19281 | 102706 | 000401 | | .WORD 401 | |
| 19282 | 102710 | 140000 | | .WORD 140000 | |
| 19283 | 102712 | | T111: | | |
| 19284 | 102712 | 000403 | | .WORD 403 | ;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES |
| 19285 | 102714 | 000000 | | .WORD 0 | |
| 19286 | 102716 | 000001 | | .WORD 1 | |
| 19287 | 102720 | 000037 | | .WORD 37 | |
| 19288 | 102722 | 000000 | | .WORD 0 | |
| 19289 | 102724 | 000000 | | .WORD 0 | |
| 19290 | 102726 | 000000 | | .WORD 0 | |
| 19291 | 102730 | 000000 | | .WORD 0 | |
| 19292 | | | | | |
| 19293 | 102732 | | T112: | | |
| 19294 | 102732 | 000403 | | .WORD 403 | ;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES |
| 19295 | 102734 | 000000 | | .WORD 0 | |
| 19296 | 102736 | 000001 | | .WORD 1 | |
| 19297 | 102740 | 000037 | | .WORD 37 | |
| 19298 | 102742 | 000000 | | .WORD 0 | |
| 19299 | 102744 | 000000 | | .WORD 0 | |
| 19300 | 102746 | 000000 | | .WORD 0 | |
| 19301 | 102750 | 000000 | | .WORD 0 | |
| 19302 | | | | | |
| 19303 | 102752 | | TL1C: | | |
| 19304 | 102752 | 000403 | | .WORD 403 | ;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES |
| 19305 | 102754 | 000000 | | .WORD 0 | |
| 19306 | 102756 | 000001 | | .WORD 1 | |
| 19307 | 102760 | 000300 | | .WORD 300 | |
| 19308 | 102762 | 000000 | | .WORD 0 | |
| 19309 | 102764 | 000000 | | .WORD 0 | |
| 19310 | 102766 | 000000 | | .WORD 0 | |
| 19311 | 102770 | 000000 | | .WORD 0 | |
| 19312 | | | | | |
| 19313 | 102772 | | TL2C: | | |
| 19314 | 102772 | 000403 | | .WORD 403 | ;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES |
| 19315 | 102774 | 000000 | | .WORD 0 | |
| 19316 | 102776 | 000001 | | .WORD 1 | |
| 19317 | 103000 | 000005 | | .WORD 5 | |
| 19318 | 103002 | 000000 | | .WORD 0 | |
| 19319 | 103004 | 000000 | | .WORD 0 | |
| 19320 | 103006 | 000000 | | .WORD 0 | |
| 19321 | 103010 | 000000 | | .WORD 0 | |
| 19322 | | | | | |
| 19323 | 103012 | | T2: | | |
| 19324 | 103012 | 000401 | | .WORD 401 | |
| 19325 | 103014 | 000200 | | .WORD 200 | |
| 19326 | 103016 | 000000 | | .WORD 0 | |
| 19327 | 103020 | 000000 | | .WORD 0 | |
| 19328 | | | | | |
| 19329 | 103022 | | T2A: | | |
| 19330 | 103022 | 000401 | | .WORD 401 | |
| 19331 | 103024 | 000201 | | .WORD 201 | |

| | | | | | |
|-------|--------|--------|--------|---------------|---|
| 19332 | 103026 | 000000 | | .WORD 0 | |
| 19333 | | | | | |
| 19334 | 103030 | 000402 | T2AA: | .WORD 402 | |
| 19335 | 103032 | 000200 | | .WORD 200 | |
| 19336 | 103034 | 000201 | | .WORD 201 | |
| 19337 | 103036 | 000000 | | .WORD 0 | |
| 19338 | | | | | |
| 19339 | 103040 | 000401 | T2B: | .WORD 401 | |
| 19340 | 103042 | 000230 | | .WORD 230 | |
| 19341 | 103044 | 000000 | | .WORD 0 | |
| 19342 | | | | | |
| 19343 | 103046 | | T4: | | |
| 19344 | 103046 | 040007 | | .WORD 40007 | :VARIABLE LENGTH ENTRIES:7 ENTRIES |
| 19345 | 103050 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 1; NO OVERLAP; TR3=TR1+TR0+SEP. CONSTANT |
| 19346 | 103054 | 063701 | 003624 | ADD TR0,R1 | |
| 19347 | 103060 | 067701 | 076532 | ADD @PTP14,R1 | |
| 19348 | 103064 | 010137 | 003632 | MOV R1,TR3 | |
| 19349 | 103070 | 000207 | | RTS PC | |
| 19350 | 103072 | 000000 | | 0 | |
| 19351 | 103074 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 2; NO OVERLAP; TR3=TR1-TR2-SEP CONSTANT |
| 19352 | 103100 | 163701 | 003630 | SUB TR2,R1 | |
| 19353 | 103104 | 167701 | 076506 | SUB @PTP14,R1 | |
| 19354 | 103110 | 010137 | 003632 | MOV R1,TR3 | |
| 19355 | 103114 | 000207 | | RTS PC | |
| 19356 | 103116 | 000000 | | 0 | |
| 19357 | 103120 | 005737 | 003624 | TST TR0 | |
| 19358 | 103124 | 001465 | | BEQ 1\$ | |
| 19359 | 103126 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 3; ADJACENT; TR3=TR1+TR0 |
| 19360 | 103132 | 063701 | 003624 | ADD TR0,R1 | :REDUNDANT WITH ENTRY 7 IF TR0=0 |
| 19361 | 103136 | 010137 | 003632 | MOV R1,TR3 | |
| 19362 | 103142 | 000207 | | RTS PC | |
| 19363 | 103144 | 000000 | | 0 | |
| 19364 | 103146 | 005737 | 003630 | TST TR2 | |
| 19365 | 103152 | 001452 | | BEQ 1\$ | |
| 19366 | 103154 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 4; ADJACENT; TR3=TR1-TR2 |
| 19367 | 103160 | 163701 | 003630 | SUB TR2,R1 | :REDUNDANT WITH ENTRY 7 IF TR2=0 |
| 19368 | 103164 | 010137 | 003632 | MOV R1,TR3 | |
| 19369 | 103170 | 000207 | | RTS PC | |
| 19370 | 103172 | 000000 | | 0 | |
| 19371 | 103174 | 013701 | 003624 | MOV TR0,R1 | :ENTRY 5; PARTIAL OVERLAP; TR3=TR1+TR0-(TR2/2) |
| 19372 | 103200 | 013702 | 003630 | MOV TR2,R2 | :REDUNDANT WITH ENTRY 7 IF TR0-(TR2/2)=0 |
| 19373 | 103204 | 006202 | | ASR R2 | |
| 19374 | 103206 | 160201 | | SUB R2,R1 | |
| 19375 | 103210 | 005701 | | TST R1 | |
| 19376 | 103212 | 001432 | | BEQ 1\$ | |
| 19377 | 103214 | 063701 | 003626 | ADD TR1,R1 | |
| 19378 | 103220 | 010137 | 003632 | MOV R1,TR3 | |
| 19379 | 103224 | 000207 | | RTS PC | |
| 19380 | 103226 | 000000 | | 0 | |
| 19381 | 103230 | 005737 | 003624 | TST TR0 | |
| 19382 | 103234 | 001421 | | BEQ 1\$ | |
| 19383 | 103236 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 6; PARTIAL OVERLAP; TR3=TR1-(TR2/2) |
| 19384 | 103242 | 013702 | 003630 | MOV TR2,R2 | :REDUNDANT WITH ENTRY 5 IF TR0=0 |
| 19385 | 103246 | 006202 | | ASR R2 | :REDUNDANT WITH ENTRY 7 IF TR2/2 =0 |

| | | | | | |
|-------|--------|--------|---------------|---------------|--|
| 19386 | 103250 | 005702 | | TST R2 | |
| 19387 | 103252 | 001412 | | BEQ 1\$ | |
| 19388 | 103254 | 160201 | | SUB R2,R1 | |
| 19389 | 103256 | 010137 | 003632 | MOV R1,TR3 | |
| 19390 | 103262 | 000207 | | RTS PC | |
| 19391 | 103264 | 000000 | | 0 | |
| 19392 | 103266 | 013737 | 003626 003632 | MOV TR1,TR3 | :ENTRY 7; COMPLETE OVERLAP; TR3=TR1 |
| 19393 | 103274 | 000207 | | RTS PC | |
| 19394 | 103276 | 000000 | | 0 | |
| 19395 | 103300 | 062706 | 000002 | ADD #2,SP | :FIXUP STACK POINTER |
| 19396 | 103304 | 000137 | 042074 | JMP REDNTC | :SKIP ENTRY TEST CONDITION - REDUNDANT |
| 19397 | | | | | |
| 19398 | 103310 | | | | |
| 19399 | 103310 | 040002 | | .WORD 40002 | :VARIABLE LENGTH ENTRIES; 2 ENTRIES |
| 19400 | 103312 | 013701 | 003624 | MOV TR0,R1 | :ENTRY 1; PARTIAL OVERLAP; TR3=TR1+TR0-(TR2/2) |
| 19401 | 103316 | 013702 | 003630 | MOV TR2,R2 | |
| 19402 | 103322 | 006202 | | ASR R2 | |
| 19403 | 103324 | 160201 | | SUB R2,R1 | |
| 19404 | 103326 | 063701 | 003626 | ADD TR1,R1 | |
| 19405 | 103332 | 010137 | 003632 | MOV R1,TR3 | |
| 19406 | 103336 | 000207 | | RTS PC | |
| 19407 | 103340 | 000000 | | 0 | |
| 19408 | 103342 | 013701 | 003626 | MOV TR1,R1 | :ENTRY 2; PARTIAL OVERLAP; TR3=TR1-(TR2/2) |
| 19409 | 103346 | 013702 | 003630 | MOV TR2,R2 | |
| 19410 | 103352 | 006202 | | ASR R2 | |
| 19411 | 103354 | 160201 | | SUB R2,R1 | |
| 19412 | 103356 | 010137 | 003632 | MOV R1,TR3 | |
| 19413 | 103362 | 000207 | | RTS PC | |
| 19414 | 103364 | 000000 | | 0 | |
| 19415 | 103366 | | | | |
| 19416 | 103366 | 040007 | | .WORD 40007 | :VARIABLE LENGTH ENTRIES; 7 ENTRIES |
| 19417 | 103370 | 013701 | 003632 | MOV TR3,R1 | :ENTRY 1; NO OVERLAP; TR1=TR2+TR3 +SEP. CONSTANT |
| 19418 | 103374 | 063701 | 003630 | ADD TR2,R1 | |
| 19419 | 103400 | 067701 | 076212 | ADD @PTP14,R1 | |
| 19420 | 103404 | 010137 | 003626 | MOV R1,TR1 | |
| 19421 | 103410 | 000207 | | RTS PC | |
| 19422 | 103412 | 000000 | | 0 | |
| 19423 | 103414 | 013701 | 003632 | MOV TR3,R1 | :ENTRY 2; NO OVERLAP; TR1=TR3-TR0-SEP CONSTANT |
| 19424 | 103420 | 163701 | 003624 | SUB TR0,R1 | |
| 19425 | 103424 | 167701 | 076166 | SUB @PTP14,R1 | |
| 19426 | 103430 | 010137 | 003626 | MOV R1,TR1 | |
| 19427 | 103434 | 000207 | | RTS PC | |
| 19428 | 103436 | 000000 | | 0 | |
| 19429 | 103440 | 005737 | 003630 | TST TR2 | |
| 19430 | 103444 | 001465 | | BEQ 1\$ | |
| 19431 | 103446 | 013701 | 003632 | MOV TR3,R1 | :ENTRY 3; ADJACENT; TR1=TR3+TR2 |
| 19432 | 103452 | 063701 | 003630 | ADD TR2,R1 | :REDUNDANT WITH ENTRY 7 IF TR2=0 |
| 19433 | 103456 | 010137 | 003626 | MOV R1,TR1 | |
| 19434 | 103462 | 000207 | | RTS PC | |
| 19435 | 103464 | 000000 | | 0 | |
| 19436 | 103466 | 005737 | 003624 | TST TR0 | |
| 19437 | 103472 | 001452 | | BEQ 1\$ | |
| 19438 | 103474 | 013701 | 003632 | MOV TR3,R1 | :ENTRY 4; ADJACENT; TR1=TR3-TR0 |
| 19439 | 103500 | 163701 | 003624 | SUB TR0,R1 | :REDUNDANT WITH ENTRY 7 IF TR0=0 |

T4A:

T4I:

| | | | | | | |
|-------|--------|--------|--------|--------|---------------|---|
| 19440 | 103504 | 010137 | 003626 | | MOV R1,TR1 | |
| 19441 | 103510 | 000207 | | | RTS PC | |
| 19442 | 103512 | 000000 | | | 0 | |
| 19443 | 103514 | 013701 | 003630 | | MOV TR2,R1 | ;ENTRY 5: PARTIAL OVERLAP; TR1=TR3+TR2-(TR0/2) |
| 19444 | 103520 | 013702 | 003624 | | MOV TR0,R2 | ;REDUNDANT WITH ENTRY 7 IF TR2-(TR0/2)=0 |
| 19445 | 103524 | 006202 | | | ASR R2 | |
| 19446 | 103526 | 160201 | | | SUB R2,R1 | |
| 19447 | 103530 | 005701 | | | TST R1 | |
| 19448 | 103532 | 001432 | | | BEQ 1\$ | |
| 19449 | 103534 | 063701 | 003632 | | ADD TR3,R1 | |
| 19450 | 103540 | 010137 | 003626 | | MOV R1,TR1 | |
| 19451 | 103544 | 000207 | | | RTS PC | |
| 19452 | 103546 | 000000 | | | 0 | |
| 19453 | 103550 | 005737 | 003630 | | TST TR2 | |
| 19454 | 103554 | 001421 | | | BEQ 1\$ | |
| 19455 | 103556 | 013701 | 003632 | | MOV TR3,R1 | ;ENTRY 6: PARTIAL OVERLAP; TR1=TR3-(TR0/2) |
| 19456 | 103562 | 013702 | 003624 | | MOV TR0,R2 | ;REDUNDANT WITH ENTRY 5 IF TR2=0 |
| 19457 | 103566 | 006202 | | | ASR R2 | ;REDUNDANT WITH ENTRY 7 IF TR0/2 =0 |
| 19458 | 103570 | 005702 | | | TST R2 | |
| 19459 | 103572 | 001412 | | | BEQ 1\$ | |
| 19460 | 103574 | 160201 | | | SUB R2,R1 | |
| 19461 | 103576 | 010137 | 003626 | | MOV R1,TR1 | |
| 19462 | 103602 | 000207 | | | RTS PC | |
| 19463 | 103604 | 000000 | | | 0 | |
| 19464 | 103606 | 013737 | 003632 | 003626 | MOV TR3,TR1 | ;ENTRY 7: COMPLETE OVERLAP; TR1=TR3 |
| 19465 | 103614 | 000207 | | | RTS PC | |
| 19466 | 103616 | 000000 | | | 0 | |
| 19467 | 103620 | 062706 | 000002 | 1\$: | ADD #2,SP | ;FIXUP STACK POINTER |
| 19468 | 103624 | 000137 | 042074 | | JMP REDNTC | ;SKIP ENTRY TEST CONDITION - REDUNDANT |
| 19469 | | | | | | |
| 19470 | 103630 | | | T41A: | | |
| 19471 | 103630 | 040001 | | | .WORD 40001 | |
| 19472 | 103632 | 013701 | 003632 | | MOV TR3,R1 | |
| 19473 | 103636 | 063701 | 003630 | | ADD TR2,R1 | |
| 19474 | 103642 | 067701 | 075750 | | ADD @PTP14,R1 | |
| 19475 | 103646 | 010137 | 003626 | | MOV R1,TR1 | |
| 19476 | 103652 | 000207 | | | RTS PC | |
| 19477 | 103654 | 000000 | | | 0 | |
| 19478 | 103656 | | | T411: | | |
| 19479 | 103656 | 040001 | | | .WORD 40001 | |
| 19480 | 103660 | 013701 | 003626 | | MOV TR1,R1 | ;VARIABLE LENGTH ENTRY: 1 ENTRY |
| 19481 | 103664 | 063701 | 003624 | | ADD TR0,R1 | ;ENTRY 1: NO OVERLAP; TR3=TR1+TR0+SEP. CONSTANT |
| 19482 | 103670 | 067701 | 075722 | | ADD @PTP14,R1 | |
| 19483 | 103674 | 010137 | 003632 | | MOV R1,TR3 | |
| 19484 | 103700 | 000207 | | | RTS PC | |