

KE44-A

CIS INSTR EXER  
CZKEEBO

AH-F227B-MC  
FICHE 1 OF 3

FEB 1981  
COPYRIGHT © 79-80  
MADE IN USA





KE44-A

CIS INSTR EXER  
CZKEEBO

AH-F227B-MC  
FICHE 2 OF 3

FEB 1981  
COPYRIGHT © 79-80  
MADE IN USA





KE44-A

CIS INSTR EXER  
CZKEEBO

AH-F227B-MC  
FICHE 3 OF 3

FEB 1981  
COPYRIGHT © 79-80  
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  
5178

.REM 8

IDENTIFICATION  
-----

PRODUCT CODE: AC-F226B-MC  
PRODUCT NAME: CZKEEBO PDP-11 CIS INST EXERCISER  
DATE: OCTOBER, 1980  
MAINTAINER: BASE SYSTEMS DIAGNOSTIC ENGINEERING

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, 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	



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  
5209

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



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  
5264

## 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



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  
5318

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  
-----



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  
5372

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) 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 A LINE TIME CLOCK (KW11-L TYPE).

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
MOVTC	354
SKPC	30



5373  
5374  
5375  
5376  
5377  
5378  
5379  
5380  
5381  
5382  
5383  
5384  
5385  
5386  
5387  
5388  
5389  
5390  
5391  
5392  
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  
5426

MATC	904
SCANC	126
SPANC	126
CVTPN	226
CVTNP	568
CVTLP	170
CVTLN	323
CVTPL	53
CVTNL	99
ADDP	1970
ADDN	3872
SUBP	1970
SUBN	3746
CMPP	502
CMPN	1089
ASHP	1972
ASHN	3872
MULP	1993
DIVP	1973

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

CZKEEBO 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: APPROX. 30 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)

THE INSTRUCTION COUNT DISPLAYED AT THE START OF TESTING FOR  
EACH INSTRUCTION IS CUMULATIVE FROM THE FIRST L2D0 CIS INSTRUCTION



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  
5480

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:

CZKEEBO 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, TEST OPERANDS AND STRING DATA FOR EACH CIS INSTRUCTION TEST WILL BE DERIVED USING A RANDOM NUMBER GENERATOR. A



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  
5534

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% COVE AGE 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 V'RIFY PASS TIME: LESS THAN 5 MINUTES  
L2D0 INST CT: XX XXXXX



J 1  
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 INSTRUCTION TEST CASES.

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  
5588



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.

SWITCH REGISTER OPTIONS:

NOTE: ON MACHINES WITH NO HARDWARE SWITCH REGISTER  
(11/24) LOCATION 176 IS THE SOFTWARE SWITCH REGISTER.

- BIT 0 - STOP ON TEST; PROGRAM WILL QUERY USER FOR TEST  
(DECIMAL) TO STOP ON AND DISPLAY.

2.4 EXECUTION TIMES

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

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  
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  
5696

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

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  RO-R6,CC/ 003760 111241 000054 111046 000344 155555 053444 1111
EXP OUT RO-R6,CC/ 003704 000000 000000 000000 000344 155555 053444 0000
ACT OUT RO-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  RO-R6,CC/ 070000 110200 070000 110206 000005 155555 053444 0100
EXP OUT RO-R6,CC/ 000000 000000 070000 000206 000000 155555 053444 1011
ACT OUT RO-R6,CC/
SRC          0+  SIGN BYTE=OF
EXP RESULT  0+  SIGN BYTE=OF
ACT RESULT  0+  SIGN BYTE=OF
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.



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  
5750

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), 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 REPT...C  
-----

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  
5777  
5778  
5779  
5780  
5781  
5782  
5783  
5784

4.1 PERFORMANCE REPORTS  
-----

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  
-----

REVISION B - MADE COMPATIBLE WITH 11/23 AND 11/24 CIS  
- ADDED STOP ON TEST FEATURE; SEE SECTION 2.3

6.0 PROGRAM TABLE OF CONTENTS  
-----



```

6235 .ENABL ABS,AMA
6351 .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
6356 .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 CRL 200 ;:CODE FOR CARRIAGE RETURN-LINE FEED
(1)
(1) : *GENERAL PURPOSE REGISTER DEFINITIONS
(1) 000000 R0= X0 ;:GENERAL REGISTER
(1) 000001 R1= X1 ;:GENERAL REGISTER
(1) 000002 R2= X2 ;:GENERAL REGISTER
(1) 000003 R3= X3 ;:GENERAL REGISTER
(1) 000004 R4= X4 ;:GENERAL REGISTER
(1) 000005 R5= X5 ;:GENERAL REGISTER
(1) 000006 R6= X6 ;:GENERAL REGISTER
(1) 000007 R7= X7 ;: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)      177766      CPUERR = 177766 ;; SYSTEM ID REGISTER
(1)      177766      CPUERR = 177766 ;; CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
(1)      177766      CPUERR = 177766 ;; THE TRAP TO ERRVEC (000004)

```

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



(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)	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)	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)	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)	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)	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

;\*KERNEL 'D' PAGE DESCRIPTOR REGISTERS

(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

;\*KERNEL 'I' PAGE ADDRESS REGISTERS

(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

;\*KERNEL 'D' PAGE ADDRESS REGISTERS

(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

.SBTTL UNIBUS MAP REGISTER DEFINITIONS

;\*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'  
;\*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

(1) 170200 MAPL00 = 170200  
(1) 170202 MAPH00 = 170202  
(1) 170204 MAPL01 = 170204  
(1) 170206 MAPH01 = 170206

(1)	170210	MAP_02 = 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)		
(1)		
6357		:
6358	076175	DIVPI==076175
6359	170000	BS0==170000
6360	100000	BS00==100000
6361	174000	BS2==174000
6362	176000	BS4==176000
6363	177000	BS8==177000
6364	177400	BS16==177400
6365	177600	BS32==177600
6366	177700	BS64==177700
6367	177740	BS128==177740
6368	177760	BS256==177760
6369	177770	BS512==177770
6370	177777	BSNULL==177777
6371	177400	BY==177400
6372	006000	MP06000==006000
6373	004000	MP04000==004000
6374	000000	WD==0
6375	125252	EOT==125252
6376	000020	MP020==000020
6377	000070	MP070==000070
6378	000100	MP0100==000100
6379	000140	MP0140==000140
6380	002000	MP02000==002000
6381	004040	MP04040==004040
6382	152525	DSCPTR==152525
6383	022000	MP02200==022000
6384		

: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)
(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 036760 JMP @#START ;; JUMP TO STARTING ADDRESS OF PROGRAM
6386 000204 000137 037154 JMP DVTST
6387 000210 000137 036766 JMP QVST
6388 000214 000137 036750 JMP SEEDST
6395
6397 ; *****
(1)
(1) .SBTTL ACT11 HOOKS
(1) ; HOOKS REQUIRED BY ACT11
(1) 000220 000220 $SVPC= . ; SAVE PC
(1) 000046 000046 . = 46 ; ; 1) SET LOC.46 TO ADDRESS OF ENDAD
(1) 000052 000052 ENDAD ; ; 2) SET LOC.52 TO ZERO
(1) 000220 000220 . = 52 ; ; RESTORE PC
(1) 001100 . = $SVPC
6398 . = 1100
6400 ; *****
(1)
(1) .SBTTL APT PARAMETER BLOCK
(1) ; SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2) ; *****
(1) 001100 . $X= . ;; 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 TIME OF LONGEST TEST
(1) 001106 000454 $PASTM: .WORD 300. ;; 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)
6405 ; *****
(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 $DEVCi: .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          $ETABLE: ;; APT ENVIRONMENT TABLE
(1) 001134          $ENV:  .BYTE  AENV  ;; ENVIRONMENT BYTE
(1) 001135          $ENVM: .BYTE  AENVM ;; ENVIRONMENT MODE BITS
(1) 001136 000000 $SWREG: .WORD  ASWREG ;; APT SWITCH REGISTER
(1) 001140 000000 $USWR:  .WORD  AUSWR  ;; USER SWITCHES
(1) 001142 000000 $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          $MAMS1: .BYTE  AMAMS1 ;; HIGH ADDRESS, M.S. BYTE
(1) 001145          $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
6406 *****
(1)
(1) .SBTTL APT COMMUNICATIONS ROUTINE
(1) 001150 112737 000001 001414 $ATY1: MOVB #1,$FFLG ;TO REPORT FATAL ERROR
(1) 001156 112737 000001 001412 $ATY3: MOVB #1,$MFLG ;TO TYPE A MESSAGE
(1) 001164 000403          BR $ATYC
(1) 001166 112737 000001 001414 $ATY4: MOVB #1,$FFLG ;TO ONLY REPORT FATAL ERROR
(2) 001174          $ATYC:
(3) 001174 010046          MOV RO,-(SP) ;; PUSH RO 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,$ENVM ;SHOULD SPOOL MESSAGE?
(1) 001224 001425          BEQ 3$ ;IF NOT: BR
(1) 001226 017600 000004          MOV @4(SP),RO ;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 RO,$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  $MSGAD,R0    ;SUB START OF MESSAGE
(1) 001262 006200          ASR  R0           ;GET MESSAGE LNTH IN WORDS
(1) 001264 010037 001132  MOV  R0,$MSGGLT  ;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 110640  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
6407 ;*****

```

6409  
6410  
6411  
6412  
6413  
6414  
6415  
6416  
6417  
6418  
6419

.SBTTL GLOBAL DATA SECTION

		.SBTTL		TEST COUNTS	
TOTTCH:	.WORD	0			:TEST COUNT MULTIPLIER
TOTTC:	.WORD	0			:TOTAL TEST COUNT
INVTC:	.WORD	0			:INVALID TEST COUNT
REDTC:	.WORD	0			:REDUNDANT TEST COUNT

001416	000000
001420	000000
001422	000000
001424	000000

6421			.SBTTL	INPUT TABLE ENTRY TYPE DISPATCH TABLE
6422			:INPUT	TABLE ENTRY TYPE DISPATCH TABLE.
6423			:	
6424	001426		!TYPE:	
6425	001426	041062		.WORD TYPE0
6426	001430	041232		.WORD TYPE1
6427	001432	000000		.WORD 0
6428	001434	000000		.WORD 0
6429				



6431  
6432  
6433  
6434 001436  
6435 001436 000000  
6436 001440 006334  
6437 001442 006334  
6438 001444 006414  
6439 001446 006776  
6440 001450 006776  
6441 001452 007134  
6442 001454 007134  
6443 001456 006476  
6444 001460 006560  
6445 001462 007044  
6446 001464 007044  
6447 001466 006642  
6448 001470 007344  
6449 001472 007216  
6450 001474 007216  
6451 001476 006716  
6452 001500 007274  
6453 001502 007044  
6454 001504 007044  
6455 001506 006642  
6456 001510 007344  
6457 001512 007044  
6458 001514 007044  
6459 001516 006716  
6460 001520 007274  
6461 001522 007406  
6462 001524 007406  
6463

.SBTTL POINTERS TO CIS INST FLOW TABLES  
:POINTERS TO CIS INSTRUCTION FLOW TABLES.

:  
:INO:

.WORD 0  
.WORD XMOVC  
.WORD XMOVRC  
.WORD XMOVTC  
.WORD XLOCC  
.WORD XSKPC  
.WORD XSCANC  
.WORD XSPANC  
.WORD XCMPC  
.WORD XMATCHC  
.WORD XADDN  
.WORD XSUBN  
.WORD XCMPP  
.WORD XCVTNL  
.WORD XCVTPN  
.WORD XCVTNP  
.WORD XASHN  
.WORD XCVTLN  
.WORD XADDP  
.WORD XSUBP  
.WORD XCMPP  
.WORD XCVTPL  
.WORD XMULP  
.WORD XDIVP  
.WORD XASHP  
.WORD XCVTLP  
.WORD XL2D  
.WORD XL3D

6465  
6466  
5467  
6468 001526  
6469 001526 053564  
6470 001530 042052  
6471 001532 042262  
6472 001534 042362  
6473 001536 042432  
6474 001540 042540  
6475 001542 042646  
6476 001544 042756  
6477 001546 045742  
6478 001550 045776  
6479 001552 046174  
6480 001554 050226  
6481 001556 051326  
6482 001560 054234  
6483 001562 053564  
6484 001564 053564  
6485  
6486

.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 16,8 17 ARE UNUSED.

6488  
6489  
6490  
6491 001566  
6492 001566 000000  
6493 001570 000000  
6494 001572 000000  
6495 001574 000000  
6496 001576 000000  
6497 001600 000000  
6498 001602 000000  
6499 001604 000000  
6500 001606 000000  
6501 001610 000000  
6502 001612 000000  
6503 001614 000000  
6504 001616 000000  
6505 001620 000000  
6506 001622 000000  
6507 001624 000000  
6508 001626 000000  
6509 001630 000000  
6510 001632 000000  
6511 001634 000000  
6512 001636 000000  
6513

.SBTTL PARAMETER TABLE POINTERS  
;PARAMETER TABLE POINTERS  
;  
;PTP:  
PTP01: .WORD 0  
PTP02: .WORD 0  
PTP03: .WORD 0  
PTP04: .WORD 0  
PTP05: .WORD 0  
PTP06: .WORD 0  
PTP07: .WORD 0  
PTP10: .WORD 0  
PTP11: .WORD 0  
PTP12: .WORD 0  
PTP13: .WORD 0  
PTP14: .WORD 0  
PTP15: .WORD 0  
PTP16: .WORD 0  
PTP17: .WORD 0  
PTP20: .WORD 0  
PTP21: .WORD 0  
PTP22: .WORD 0  
PTP23: .WORD 0  
PTP24: .WORD 0



6515			.SBTTL	MISCELLANEOUS CONSTANTS	
6516			;MISCELLANEOUS CONSTANTS		
6517			:		
6519	001640	130000	TBADR: .WORD	130000	;TEST BUFFER STARTING ADDRESS
6520	001642	002000	TBLEN: .WORD	2000	;TEST BUFFER LENGTH
6521	001644	010000	RTBLEN: .WORD	10000	;RANDOM MODE TEST BUFFER LENGTH
6522	001646	140000	EBADR: .WORD	140000	;EMULATION BUFFER STARTING ADDRESS
6523	001650	002000	EBLEN: .WORD	2000	;EMULATION BUFFER LENGTH
6524	001652	020000	TPERP: .WORD	20000	;# OF TESTS PER APT PASS IN RANDOM MODE
6525	001654	002000	PSEED: .WORD	2000	;# OF TESTS TO EXECUTE IN RANDOM MODE
6526					
6527					
6528					
6542	001656	000024	IPNU: .WORD	24	;# OF INPUT PARAMETERS PER INPUT
6543					; TABLE ENTRY.
6544	001660	000001	INCSQ1: .WORD	1	;INCREMENT SEQUENCE WORD 1
6545	001662	000002	INCSQ2: .WORD	2	;INCREMENT SEQUENCE WORD 2
6546	001664	177776	TPSW: .WORD	177776	;PSW
6547	001666	177560	TKS:	177560	
6548	001670	000240	KNOP:	NOP	
6549	001672	000000	KHALT:	HALT	
6550	001674	000405	KBR5: .WORD	405	
6551	001676	000403	KBR3: .WORD	403	
6552	001700	000402	KBR2: .WORD	402	
6553	001702	123321	NOTREG: .WORD	123321	;PATTERN LOADED INTO REG SET NO SELECTED FOR TEST
6554	001704	000002	EL74: .WORD	2	;11/74 PROCESSOR TYPE
6555	001706	000001	EL44: .WORD	1	;11/44 PROCESSOR TYPE
6556	001710	000003	EL2324: .WORD	3	;11/23/24 PROCESSOR TYPE
6557					; BETWEEN PRINTING OF RNG SEED.
6558					;NOTE: ONLY 1 BIT IS ALLOWED TO BE SET IN PSEED.
6559	001712	177562	TKB:	177562	
6560	001714	177564	TPS:	177564	
6561	001716	177566	TPB:	177566	
6562		000015	CR=	15	
6563		000012	LF=	12	
6564		000057	SL=	57	
6565		000010	BS=	10	
6566		000007	MFPT=	7	
6567	001720	177777	PAT0: .WORD	177777	
6568	001722	111111	PAT1: .WORD	111111	
6569	001724	122222	PAT2: .WORD	122222	
6570	001726	133333	PAT3: .WORD	133333	
6571	001730	144444	PAT4: .WORD	144444	
6572	001732	155555	PAT5: .WORD	155555	
6573	001734	120606	IXLTB1: .WORD	XLTLB1	
6574	001736	177570	SWR: .WORD	DSWR	;ADDRESS OF SWITCH REGISTER
6575	001740	177570	DISPLAY: .WORD	DDISP	;ADDRESS OF DISPLAY REGISTER

6577				
6578			.SBTTL	MISCELLANEOUS VARIABLES
6579			:MISCELLANEOUS VARIABLES	
6580			:	
6581	001742	000000	FLOPTR:	.WORD 0
6582	001744	000000	ESEED:	.WORD 0
6583	001746	000000	DENS:	.WORD 0
6584	001750	000000	N200M:	.WORD 0
6585	001752	000000	TSTPSW:	.WORD 0
6586	001754	000000	TBEND:	.WORD 0
6587	001756	000000	SBR:	.WORD 0
6588	001760	000000	RANDOM:	.WORD 0
6589	001762	000000	CTACT:	.WORD 0
6590	001764	000000	PMASK:	.WORD 0
6591	001766	177417	ZMSK:	.WORD 177417
6592	001770	000000	RANDTA:	.WORD 0
6593	001772	000000	LIMSTG:	.WORD 0
6594	001774	000000	RNIB:	.WORD 0
6595	001776	000000	STRNC:	.WORD 0
6596	002000	000000	STRP1:	.WORD 0
6597	002002	000000	STRP2:	.WORD 0
6598	002004	000000	SXRNC:	.WORD 0
6599	002006	000000	SXRP1:	.WORD 0
6600	002010	000000	SXRP2:	.WORD 0
6601	002012	000000	SYRNC:	.WORD 0
6602	002014	000000	SYRP1:	.WORD 0
6603	002016	000000	SYRP2:	.WORD 0
6604	002020	000000	SWRNC:	.WORD 0
6605	002022	000000	SWRP1:	.WORD 0
6606	002024	000000	SWRP2:	.WORD 0
6607	002026	000000	SVRNC:	.WORD 0
6608	002030	000000	SVRP1:	.WORD 0
6609	002032	000000	SVRP2:	.WORD 0
6610	002034	000000	ZCCR:	.WORD 0
6611	002036	000000	SPCV:	.WORD 0
6612	002040	000000	TPRECS:	.WORD 0
6613	002042	000000	NOERDS:	.WORD 0
6614	002044	000000	PROGD:	.WORD 0
6615	002046	000000	SURLEN:	.WORD 0
6616	002050	000000	SURADR:	.WORD 0
6617	002052	000000	REGSET:	.WORD 0
6618	002054	000000	ERRCT:	.WORD 0
6619	002056	000000	ERRS:	.WORD 0
6620	002060	000000	TWOSETS:	.WORD 0
6621	002062	000000	FAVRS:	.WORD 0
6622	002064	000000	FAVR4:	.WORD 0
6623	002066	000000	FAVR3:	.WORD 0
6624	002070	000000	FAVR2:	.WORD 0
6625	002072	000000	FAVR1:	.WORD 0
6626	002074	000000	FSRUN:	.WORD 0
6627	002076	000000	MSEED:	.WORD 0
6628	002100	000000	NBLKS:	.WORD 0
6629	002102	000000	TRA:	.WORD 0
6630	002104	000000	TRL:	.WORD 0

6631	002106	000000	STGDS1:	.WORD	0
6632	002110	000000	STGDS2:	.WORD	0
6633	002112	000000	STGLN:	.WORD	0
6634	002114	000000	STGAD:	.WORD	0
6635	002116	000000	SAVSL:	.WORD	0
6636	002120	000000	SAVSA:	.WORD	0
6637	002122	000000	SAVSGL:	.WORD	0
6638	002124	000000	SIGN:	.WORD	0
6639	002126	000000	VIP:	.WORD	0
6640	002130	000000	PTW1:	.WORD	0
6641	002132	000000	FATAL:	.WORD	0
6642	002134	000000	INPTP:	.WORD	0
6643	002136	000000	PTPTR:	.WORD	0
6644	002140	000000	SPHAND:	.WORD	0
6645	002142	000000	EMPTR:	.WORD	0
6646	002144	000000	ERRCC:	.WORD	0
6647	002146	000000	ERRREG:	.WORD	0
6648	002150	000000	ERRBUF:	.WORD	0
6649	002152	000000	ERRSTK:	.WORD	0
6650	002154	000000	PT34:	.WORD	0
6651	002156	000000	MMFLG:	.WORD	0
6652	002160	000000	MODE:	.WORD	0
6653					
6654	002162	000000	DEN:	.WORD	0
6655	002164	000000	NMODES:	.WORD	0
6656					
6657	002166	000000	TRPLOC:	.WORD	0
6658	002170	000000	HLTLOC:	.WORD	0
6659	002172	000000	IRXLT:	.WORD	0
6660	002174	000000	AADR:	.WORD	0
6661	002176	000000	AEDTA:	.WORD	0
6662	002200	000000	EMADR:	.WORD	0
6663	002202	000000	EMDTA:	.WORD	0
6664	002204	000000	QRYFLG:	.WORD	0
6665	002206	000000	QVMODE:	.WORD	0
6666	002210	000000	MMSTAT:	.WORD	0
6667	002212	000000	PTQV:	.WORD	0
6668	002214	000000	ICMPC:	.WORD	0
6669	002216	000000	STOPTF:	.WORD	0
6670	002220	000000	STOPT:	.WORD	0
6671	002222	000000	RLL:	.WORD	0
6672	002224	000000	RUL:	.WORD	0
6673	002226	000000	BAD:	.WORD	0
6674	002230	000000	TSP:	.WORD	0
6675	002232	000000	NXFLD:	.WORD	0
6676	002234	000000	TTR0:	.WORD	0
6677	002236	000000	TTR1:	.WORD	0
6678	002240	000000	TTR2:	.WORD	0
6679	002242	000000	TTR3:	.WORD	0
6680	002244	000000	TTR4:	.WORD	0
6681	002246	000000	TTR5:	.WORD	0
6682	002250	000000	TTR6:	.WORD	0
6683	002252	000000	TEROR:	.WORD	0
6684	002254	000000	TER1R:	.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)

6685	002256	000000	TER2R:	.WORD	0	
6686	002260	000000	TER3R:	.WORD	0	
6687	002262	000000	TER4R:	.WORD	0	
6688	002264	000000	TER5R:	.WORD	0	
6689	002266	000000	TER6R:	.WORD	0	
6690	002270	000000	TERR:	.WORD	0	
6691	002272	000000	RPTFLG:	.WORD	0	
6692	002274	000000	FILLS2:	.WORD	0	
6693	002276	000000	OCTIC:	.WORD	0	
6694	002300	C00000	TW1:	.WORD	0	
6695	002302	000000	TW2:	.WORD	0	
6696	002304	000000	PRTSGN:	.WORD	0	
6697	002306	000000	STGDIG:	.WORD	0	
6698	002310	000000	STGTYP:	.WORD	0	
6699	002312	044072	TYPTAB:	.WORD	TYPSZ	:SIGNED ZONED
6700	002314	044124		.WORD	TYPUZ	:UNSIGNED ZONED
6701	002316	044126		.WORD	TYPTO	:TRAILING OVERPUNCH
6702	002320	044234		.WORD	TYPLO	:LEADING OVERPUNCH
6703	002322	044244		.WORD	TYPTS	:TRAILING SEPARATE
6704	002324	044310		.WORD	TYPLS	:LEADING SEPARATE
6705	002326	002332		.WORD	TYPSP	:RESERVED
6706	002330	002332		.WORD	TYPSP	:RESERVED
6707	002332	000000	TYPSP:	HALT		
6708	002334	175	NEGTAB:	.BYTE	175	:-0
6709	002335	112		.BYTE	112	:-1
6710	002336	113		.BYTE	113	:-2
6711	002337	114		.BYTE	114	:-3
6712	002340	115		.BYTE	115	:-4
6713	002341	116		.BYTE	116	:-5
6714	002342	117		.BYTE	117	:-6
6715	002343	120		.BYTE	120	:-7
6716	002344	121		.BYTE	121	:-8
6717	002345	122		.BYTE	122	:-9
6718	002346	135	NEGTB1:	.BYTE	135	:-0
6719	002347	112		.BYTE	112	:-1
6720	002350	113		.BYTE	113	:-2
6721	002351	114		.BYTE	114	:-3
6722	002352	115		.BYTE	115	:-4
6723	002353	116		.BYTE	116	:-5
6724	002354	117		.BYTE	117	:-6
6725	002355	120		.BYTE	120	:-7
6726	002356	121		.BYTE	121	:-8
6727	002357	122		.BYTE	122	:-9
6728	002360	173	POSTAB:	.BYTE	173	:+0
6729	002361	101		.BYTE	101	:+1
6730	002362	102		.BYTE	102	:+2
6731	002363	103		.BYTE	103	:+3
6732	002364	104		.BYTE	104	:+4
6733	002365	105		.BYTE	105	:+5
6734	002366	106		.BYTE	106	:+6
6735	002367	107		.BYTE	107	:+7
6736	002370	110		.BYTE	110	:+8
6737	002371	111		.BYTE	111	:+9
6738	002372	060	POSTB1:	.BYTE	060	:+0



6739	002373	061	.BYTE 061	:+1
6740	002374	062	.BYTE 062	:+2
6741	002375	063	.BYTE 063	:+3
6742	002376	064	.BYTE 064	:+4
6743	002377	065	.BYTE 065	:+5
6744	002400	066	.BYTE 066	:+6
6745	002401	067	.BYTE 067	:+7
6746	002402	070	.BYTE 070	:+8
6747	002403	071	.BYTE 071	:+9
6748	002404	133	POSTB2: .BYTE 133	:+0
6749	002405	061	.BYTE 061	:+1
6750	002406	062	.BYTE 062	:+2
6751	002407	063	.BYTE 063	:+3
6752	002410	064	.BYTE 064	:+4
6753	002411	065	.BYTE 065	:+5
6754	002412	066	.BYTE 066	:+6
6755	002413	067	.BYTE 067	:+7
6756	002414	070	.BYTE 070	:+8
6757	002415	071	.BYTE 071	:+9
6758	002416	045244	PTYPTA: .WORD PTYPSZ	:SIGNED ZONED
6759	002420	045740	.WORD EISTG	:UNSIGNED ZONED
6760	002422	045106	.WORD PTYPTO	:TRAILING OVERPUNCH
6761	002424	045254	.WORD PTYPLO	:LEADING OVERPUNCH
6762	002426	045264	.WORD PTYPTS	:TRAILING SEPARATE
6763	002430	045300	.WORD PTYPLS	:LEADING SEPARATE
6764	002432	002332	.WORD TYPSP	:RESERVED
6765	002434	002332	.WORD TYPSP	:RESERVED
6766	002436	000000	ONEBEY: .WORD 0	
6767	002440	000000	DECINS: .WORD 0	
6768	002442	000006	S1TYPE: .WORD 6	
6769	002444	000006	S2TYPE: .WORD 6	
6770	002446	000006	S3TYPE: .WORD 6	
6771	002450	070000	TYPFLD: .WORD 070000	
6772	002452	000000	MIXTYP: .WORD 0	: MANUALLY SET TO ANY NONZERO VALUE
6773				: TO CAUSE TESTING OF MIXED DATA TYPES
6774				: WITHIN INST. NOTE: THIS WILL
6775				: GREATLY INCREASE RUN TIME!!!
6776	002454	000000	PKPTW: .WORD 0	
6777	002456	000000	NDESC: .WORD 0	
6778	002460	000000	ZPM: .WORD 0	
6779	002462	000000	ACINST: .WORD 0	
6780	002464	000000	.WORD 0	
6781	002466	000000	.WORD 0	
6782	002470	000000	ONEINS: .WORD 0	
6783	002472	000000	PZCODE: .WORD 0	
6784	002474	000000	SAVPTR: .WORD 0	
6785	002476	000000	SAVSRF: .WORD 0	
6786	002500	000000	INSRC1: .WORD 0	:SRC1 STRING SAVE BUFFER DESCRIPTOR
6787	002502	003041	.WORD BUFSR1	
6788	002504	000000	INSRC2: .WORD 0	:SRC2 STRING SAVE BUFFFFR DESCRIPTOR
6789	002506	003102	.WORD BUFSR2	
6790				
6791	002510	120000	RANDSC: .WORD 120000	
6792	002512	000000	.WORD 0	

```

6794          .SBTTL          PROGRAMMABLE CLOCK CONSTANTS
6795          ;PROGRAMMABLE CLOCK CONSTANTS
6796 002514 000104          PCLK1V: .WORD 104          ;P-CLK 1
6797 002516 000106          PCLK1P: .WORD 106
6798 002520 172540          PC1CSR: .WORD 172540
6799 002522 172542          PC1CSB: .WORD 172542
6800 002524 172544          PC1CTR: .WORD 172544
6801 002526 000000          PCLK2V: .WORD 000          ;P-CLK 2
6802 002530 172550          PC2CSR: .WORD 172550
6803 002532 172552          PC2CSB: .WORD 172552
6804 002534 172554          PC2CTR: .WORD 172554
6805 002536 000004          TIMEOUT: .WORD 4
6806
6807 002540 000000          PROGCT: .WORD 0
6808 002542 000000          LATCT: .WORD 0
6809 002544 000000          LATEN: .WORD 0
6810 002546 000000          INTCT: .WORD 0
6811 002550 000240          KNOP1: .WORD NOP
6812 002552 000240          KNOP2: .WORD NOP
6813 002554 000240          KNOP3: .WORD NOP
6814 002556 000000          INTRVL: .WORD 0
6815 002560 000000          STOPLA: .WORD 0          ;IF NONZERO & LATENCY EXCEEDS THIS VALUE PROGRAM HALTS
6816 002562 001000          MAXIVL: .WORD 1000      ;USER DEFINED MAXIMUM INTERVAL ALLOWED
6817
6818 002564 000000          STATPS: .WORD 0
6819 002566 000000          STATR0: .WORD 0
6820 002570 000000          STATR1: .WORD 0
6821 002572 000000          STATR2: .WORD 0
6822 002574 000000          STATR3: .WORD 0
6823 002576 000000          STATR4: .WORD 0
6824 002600 000000          STATR5: .WORD 0
6825 002602 000000          STATR6: .WORD 0
6826 002604 000100          .BLKW ^D64
6827 003004          SCSTK:
6828
6829 003004 000000          SGPRO: .WORD 0
6830 003006 000000          SGPR1: .WORD 0
6831 003010 000000          SGPR2: .WORD 0
6832 003012 000000          SGPR3: .WORD 0
6833 003014 000000          SGPR4: .WORD 0
6834 003016 000000          SGPR5: .WORD 0
6835 003020 000000          SGPR6: .WORD 0
6836
6837

```

6839  
6840  
6841 003022 000100  
6842 003024 000102  
6843 003026 177546  
6844 003030 000000  
6845 003032 000000  
6846 003034 000240  
6847 003036 000000  
6848  
6849

.SBTTL LINE TIME CLOCK CONSTANTS  
;LINE TIME CLOCK CONSTANTS  
LTCIV: .WORD 100  
LTCIP: .WORD 102  
LKS: .WORD 177546  
LCNT: .WORD 0  
VLCNT: .WORD 0  
KNOP4: .WORD 240  
LTCPLY: .WORD 0

6851					
6852			.SBTTL	SOURCE STPING STORAGE BUFFER	
6853			;SOURCE	STRING STORAGE BUFFER - USED BY ERROR PRINTOUT ROUTINES	
6854			:		
6855	003040	000			
6856	003041	000040	BUFSR1:	.BLKB ^D32	:S1 BUFFER
6857	003101	000			
6858	003102	000040	BUFSR2:	.BLKB ^D32	:S2 BUFFER
6859					
6860					
6861					
6862	003142	000	PB0:	.BYTE	0 :DISPLAY BUFFER
6863	003143	000	PB1:	.BYTE	0
6864	003144	000	PB2:	.BYTE	0
6865	003145	000	PB3:	.BYTE	0
6866	003146	000	PB4:	.BYTE	0
6867	003147	000	PB5:	.BYTE	0
6868	003150	000	PB6:	.BYTE	0
6869	003151	000	PB7:	.BYTE	0
6870					



6872  
6873  
6874  
6875 003152 000012  
6876 003154 003250  
6877 003156 000020  
6878 003160 003162  
6879 003162 000031  
6880  
6881

; DESCRIPTORS AND DESTINATION BUFFER FOR INTERRUPTABILITY SERVICE  
; ROUTINE DIVPI INST.  
;  
DIVDS: .WORD 12 ;SOURCE 1 & 2 DESC  
; .WORD SSTG2  
DI:J. .WORD 20 ;DESTINATION DESC  
; .WORD DESTBUF  
DESTBUF: .BLKW 31

			.SBTTL	PRE-SPECIFIED STRINGS	
			:PRE-SPECIFIED STRINGS		
6883					
6884					
6885					
6886	003244	001	SSTG1:	.BYTE 001	
6887	003245	001		.BYTE 001	
6888	003246	007		.BYTE 007	
6889	003247	000		.BYTE 000	
6890					
6891	003250	012002	SSTG2:	.WORD 012002	;PACKED 2,147,483,648 +MAX+1
6892	003252	101564		.WORD 101564	
6893	003254	106144		.WORD 106144	
6894					
6895	003256	024404	SSTG2A:	.WORD 024404	;PACKED 4,294,967,294 +MAX * 2
6896	003260	063511		.WORD 063511	
6897	003262	046051		.WORD 046051	
6898					
6899	003264	112102	SSTG2B:	.WORD 112102	;PACKED 42,949,672,940 +MAX * 20
6900	003266	071226		.WORD 071226	
6901	003270	006224		.WORD 006224	
6902					
6903	003272	000402	SSTG3:	.WORD 000402	;ZONED 2,147,483,648 +MAX+1
6904	003274	003404		.WORD 003404	
6905	003276	034064		.WORD 034064	
6906	003300	003003		.WORD 003003	
6907	003302	034164		.WORD 034164	
6908					
6909	003304	012002	SSTG4:	.WORD 012002	;PACKED 2,147,483,647 +MAX
6910	003306	101564		.WORD 101564	
6911	003310	076144		.WORD 076144	
6912					
6913	003312	000402	SSTG5:	.WORD 000402	;ZONED 2,147,483,647 +MAX
6914	003314	003404		.WORD 003404	
6915	003316	034064		.WORD 034064	
6916	003320	003003		.WORD 003003	
6917	003322	033564		.WORD 033564	
6918					
6919	003324	012002	SSTG6:	.WORD 012002	;PACKED -2,147,483,648 -MAX
6920	003326	101564		.WORD 101564	
6921	003330	106544		.WORD 106544	
6922					
6923	003332	000402	SSTG7:	.WORD 000402	;ZONED -2,147,483,648 -MAX
6924	003334	003404		.WORD 003404	
6925	003336	034064		.WORD 034064	
6926	003340	003003		.WORD 003003	
6927	003342	074164		.WORD 074164	
6928					
6929	003344	012002	SSTG10:	.WORD 012002	;PACKED -2,147,483,649 -MAX-1
6930	003346	101564		.WORD 101564	
6931	003350	116544		.WORD 116544	
6932					
6933	003352	000402	SSTG11:	.WORD 000402	;ZONED -2,147,483,649 -MAX-1
6934	003354	003404		.WORD 003404	
6935	003356	034064		.WORD 034064	
6936	003360	003003		.WORD 003003	

6937	003362	074564	.WORD	074564	
6938					
6939	003364	032022	SSTG12: .WORD	032022	:PACKED STRING
6940	003366	074126	.WORD	074126	: 1234567891234567891234000891233
6941	003370	021621	.WORD	021621	
6942	003372	063505	.WORD	063505	
6943	003374	011211	.WORD	011211	
6944	003376	000064	.WORD	000064	
6945	003400	110410	.WORD	110410	
6946	003402	036043	.WORD	036043	
6947					
6948	003404	000000	STG12B: .WORD	000000	:PACKED STRING
6949	003406	000000	.WORD	000000	
6950	003410	000000	.WORD	000000	
6951	003412	000000	.WORD	000000	
6952	003414	000000	.WORD	000000	
6953	003416	000000	.WORD	000000	
6954	003420	000000	.WORD	000000	
6955	003422	106610	.WORD	106610	: 000888-
6956					
6957	003424	000100	STG12C: .WORD	000100	:PACKED STRING
6958	003426	000000	.WORD	000000	: 40000000000000000000000000000000-
6959	003430	000000	.WORD	000000	
6960	003432	000000	.WORD	000000	
6961	003434	000000	.WORD	000000	
6962	003436	000000	.WORD	000000	
6963	003440	000000	.WORD	000000	
6964	003442	006400	.WORD	006400	
6965					
6966	003444	001001	SSTG13: .WORD	001001	:ZONED STRING
6967	003446	002003	.WORD	002003	: 1234567891234567891234000891233
6968	003450	003005	.WORD	003005	
6969	003452	004007	.WORD	004007	
6970	003454	000411	.WORD	000411	
6971	003456	101602	.WORD	101602	
6972	003460	072564	.WORD	072564	
6973	003462	073566	.WORD	073566	
6974	003464	034470	.WORD	034470	
6975	003466	031061	.WORD	031061	
6976	003470	002003	.WORD	002003	
6977	003472	000000	.WORD	000000	
6978	003474	004000	.WORD	004000	
6979	003476	000411	.WORD	000411	
6980	003500	001402	.WORD	001402	
6981	003502	000063	.WORD	000063	
6982					
6983	003504	000000	STG13B: .WORD	000000	:ZONED STRING
6984	003506	000000	.WORD	000000	
6985	003510	000000	.WORD	000000	
6986	003512	000000	.WORD	000000	
6987	003514	000000	.WORD	000000	
6988	003516	000000	.WORD	000000	
6989	003520	000000	.WORD	000000	
6990	003522	000000	.WORD	000000	

6991 003524 000000  
6992 003526 000000  
6993 003530 000000  
6994 003532 000000  
6995 003534 000000  
6996 003536 000000  
6997 003540 004000  
6998 003542 074010  
6999  
7000 003544 000004  
7001 003546 000000  
7002 003550 000000  
7003 003552 000000  
7004 003554 000000  
7005 003556 000000  
7006 003560 000000  
7007 003562 000000  
7008 003564 000000  
7009 003566 000000  
7010 003570 000000  
7011 003572 000000  
7012 003574 000000  
7013 003576 000000  
7014 003600 000000  
7015 003602 000160  
7016  
7017 003604 042006  
7018 003606 050044  
7019 003610 046224  
7020  
7021 003612 042006  
7022 003614 050044  
7023 003616 046624  
7024  
7025 003620 000000  
7026 003622 000000  
7027 003624 000000  
7028 003626 036063  
7029

.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 004000  
.WORD 074010  
  
STG13C: .WORD 000004  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 000160  
  
SSTG14: .WORD 042006  
.WORD 050044  
.WORD 046224  
  
SSTG15: .WORD 42006  
.WORD 50044  
.WORD 46624  
  
SSTG16: .WORD 000000  
.WORD 000000  
.WORD 000000  
.WORD 036063

: 000888-  
  
: ZONED STRING  
: 40000000000000000000000000000000-  
  
: PACKED STRING 3X2\*\*31  
: 6442450944 +  
  
: PACKED STRING 3X2\*\*31 -  
: 6442450944 -  
  
: PACKED STRING  
: 000000000000333+



```
7031          .SBTTL          TEST AND EMULATION OPERANDS
7032          ;TEST OPERANDS
7033          ;
7034 003630     TRN:
7035 003630     TR0:          .WORD 0
7036 003632     TR1:          .WORD 0
7037 003634     TR2:          .WORD 0
7038 003636     TR3:          .WORD 0
7039 003640     TR4:          .WORD 0
7040 003642     TP5:          .WORD 0
7041 003644     TR6:          .WORD 0
7042 003646     TCC:          .WORD 0
7043
7044          ;TEST RESULTS - REGISTERS
7045          ;
7046 003650     TRNR:
7047 003650     TR0R:         .WORD 0
7048 003652     TR1R:         .WORD 0
7049 003654     TR2R:         .WORD 0
7050 003656     TR3R:         .WORD 0
7051 003660     TR4R:         .WORD 0
7052 003662     TR5R:         .WORD 0
7053 003664     TR6R:         .WORD 0
7054
7055          ;TEST RESULTS - CONDITION CODES
7056          ;
7057 003666     TCCR:         .WORD 0
7058
7059          ;EMULATION OPERANDS
7060          ;
7061 003670     ERN:
7062 003670     ER0:          .WORD 0
7063 003672     ER1:          .WORD 0
7064 003674     ER2:          .WORD 0
7065 003676     ER3:          .WORD 0
7066 003700     ER4:          .WORD 0
7067 003702     ER5:          .WORD 0
7068 003704     ER6:          .WORD 0
7069
7070          ;EMULATION RESULTS - REGISTERS
7071          ;
7072 003706     ERNR:
7073 003706     ER0R:         .WORD 0
7074 003710     ER1R:         .WORD 0
7075 003712     ER2R:         .WORD 0
7076 003714     ER3R:         .WORD 0
7077 003716     ER4R:         .WORD 0
7078 003720     ER5R:         .WORD 0
7079 003722     ER6R:         .WORD 0
7080
7081          ;EMULATION RESULTS - CONDITION CODES
7082          ;
7083 003724     ECCR:         .WORD 0
```

7085  
7086  
7087  
7088 003726 00000Q  
7089 003730 076030  
7090 003732 076031  
7091 003734 076032  
7092 003736 076040  
7093 003740 076041  
7094 003742 076042  
7095 003744 076043  
7096 003746 076044  
7097 003750 076045  
7098 003752 076050  
7099 003754 076051  
7100 003756 076052  
7101 003760 076053  
7102 003762 076054  
7103 003764 076055  
7104 003766 076056  
7105 003770 076057  
7106 003772 076070  
7107 003774 076071  
7108 003776 076072  
7109 004000 076073  
7110 004002 076074  
7111 004004 076075  
7112 004006 076076  
7113 004010 076077  
7114 004012 076020  
7115 004014 076060

.SBTTL OCTAL CODING FOR EACH CIS INSTRUCTION  
:OCTAL CODING FOR EACH CIS INSTRUCTION  
:OINST: .WORD 0 : 0 - UNASSIGNED  
: .WORD 76030 : 1 - MOVC  
: .WORD 76031 : 2 - MOVRC  
: .WORD 76032 : 3 - MOVTC  
: .WORD 76040 : 4 - LOCC  
: .WORD 76041 : 5 - SKPC  
: .WORD 76042 : 6 - SCANC  
: .WORD 76043 : 7 - SPANC  
: .WORD 76044 :10 - CMPC  
: .WORD 76045 :11 - MATCHC  
: .WORD 76050 :12 - ADDN  
: .WORD 76051 :13 - SUBN  
: .WORD 76052 :14 - CMPN  
: .WORD 76053 :15 - CVTNL  
: .WORD 76054 :16 - CVTPN  
: .WORD 76055 :17 - CVTNP  
: .WORD 76056 :20 - ASHN  
: .WORD 76057 :21 - CVTLN  
: .WORD 76070 :22 - ADDP  
: .WORD 76071 :23 - SUBP  
: .WORD 76072 :24 - CMPP  
: .WORD 76073 :25 - CVTPL  
: .WORD 76074 :26 - MULP  
: .WORD 76075 :27 - DIVP  
: .WORD 76076 :30 - ASHP  
: .WORD 76077 :31 - CVTLP  
: .WORD 76020 :32 - L2DR  
: .WORD 76060 :33 - L3DR

7117  
7118  
7119  
7120 004016 000000  
7121 004020 000000  
7122 004022 000000  
7123 004024 000000  
7124 004026 000003  
7125 004030 000003  
7126 004032 000003  
7127 004034 000003  
7128 004036 000000  
7129 004040 000003  
7130 004042 000001  
7131 004044 000001  
7132 004046 000003  
7133 004050 000000  
7134 004052 000001  
7135 004054 000001  
7136 004056 000001  
7137 004060 000001  
7138 004062 000001  
7139 004064 000001  
7140 004066 000003  
7141 004070 000000  
7142 004072 000001  
7143 004074 000000  
7144 004076 000001  
7145 004100 000001  
7146 004102 000000  
7147 004104 000000  
7148

.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 - MOVCL  
.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 - CMPPL  
.WORD 0 :25 - CVTPL  
.WORD 1 :26 - MULP  
.WORD 0 :27 - DIVP  
.WORD 1 :30 - ASHP  
.WORD 1 :31 - CVTLP  
.WORD 0 :32 - L2DR  
.WORD 0 :33 - L3DR

7150  
7151  
7152  
7153 004106 000000  
7154 004110 000000  
7155 004112 000000  
7156 004114 000000  
7157 004116 000000  
7158 004120 000000  
7159 004122 000000  
7160 004124 000000  
7161 004126 000000  
7162 004130 000000  
7163 004132 000000  
7164 004134 000000  
7165 004136 000000  
7166 004140 000000  
7167 004142 000000  
7168 004144 000000  
7169 004146 000000  
7170 004150 000000  
7171 004152 000000  
7172 004154 000000  
7173 004156 000000  
7174 004160 000000  
7175 004162 000000  
7176 004164 000000  
7177 004166 000000  
7178 004170 000000  
7179 004172 000000  
7180 004174 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



7182  
7183  
7184  
7185 004176 000000  
7186 004200 006070  
7187 004202 006070  
7188 004204 006104  
7189 004206 006140  
7190 004210 006140  
7191 004212 006150  
7192 004214 006150  
7193 004216 006070  
7194 004220 006164  
7195 004222 006244  
7196 004224 006244  
7197 004226 006272  
7198 004230 006226  
7199 004232 006200  
7200 004234 006200  
7201 004236 006316  
7202 004240 006214  
7203 004242 006244  
7204 004244 006244  
7205 004246 006272  
7206 004250 006226  
7207 004252 006244  
7208 004254 006244  
7209 004256 006316  
7210 004260 006214

.SBTTL RANDOM EXERCISE MODE MASK TABLE POINTERS  
;RANDOM EXERCISE MODE MASK TABLE POINTERS  
;MINST: .WORD 0 : 0 - UNASSIGNED  
          .WORD MMOV C : 1 - MOV C  
          .WORD MMOVRC : 2 - MOVRC  
          .WORD MMOVTC : 3 - MOVTC  
          .WORD MLOCC : 4 - LOCC  
          .WORD MSKPC : 5 - SKPC  
          .WORD MSCANC : 6 - SCANC  
          .WORD MSPANC : 7 - SPANC  
          .WORD MCMPC :10 - CMPC  
          .WORD MMTCHC :11 - MATCHC  
          .WORD MADDN :12 - ADDN  
          .WORD MSUBN :13 - SUBN  
          .WORD MCMPN :14 - CMPN  
          .WORD MCVTNL :15 - CVTNL  
          .WORD MCVTPN :16 - CVTPN  
          .WORD MCVTNP :17 - CVTNP  
          .WORD MASHN :20 - ASHN  
          .WORD MCVTLN :21 - CVTLN  
          .WORD MADDP :22 - ADDP  
          .WORD MSUBP :23 - SUBP  
          .WORD MCMPP :24 - CMPP  
          .WORD MCVTPL :25 - CVTPL  
          .WORD MMULP :26 - MULP  
          .WORD MDIVP :27 - DIVP  
          .WORD MASHP :30 - ASHP  
          .WORD MCVTLP :31 - CVTLP

			.SBTTL	DECIMAL INST DATA TYPE CONTROL WORDS		
			:DECIMAL	INSTRUCTION DATA TYPE CONTROL WORDS		
7212						
7213						
7214						
7215	004262	004214	DECTYP:	.WORD DECTTB-50		
7216						
7217	004264	001400	DECTTB:	.WORD 1400	:ADDN	NDESC,PKPTW
7218	004266	002000		.WORD 2000	:	ZPM,SXTYPE
7219	004270	001400		.WORD 1400	:SUBN	
7220	004272	002000		.WORD 2000		
7221	004274	001000		.WORD 1000	:CMPN	
7222	004276	002000		.WORD 2000		
7223	004300	000400		.WORD 0400	:CVTNL	
7224	004302	002000		.WORD 2000		
7225	004304	001006		.WORD 1006	:CVTPN	
7226	004306	004006		.WORD 4006		
7227	004310	001060		.WORD 1060	:CVTNP	
7228	004312	003060		.WORD 3060		
7229	004314	001000		.WORD 1000	:ASHN	
7230	004316	002000		.WORD 2000		
7231	004320	000400		.WORD 0400	:CVTLN	
7232	004322	002000		.WORD 2000		
7233	004324	001400		.WORD 1400	:ADDP	
7234	004326	001777		.WORD 1777		
7235	004330	001400		.WORD 1400	:SUBP	
7236	004332	001777		.WORD 1777		
7237	004334	001000		.WORD 1000	:CMPP	
7238	004336	001777		.WORD 1777		
7239	004340	000400		.WORD 0400	:CVTPL	
7240	004342	001777		.WORD 1777		
7241	004344	001400		.WORD 1400	:MULP	
7242	004346	001777		.WORD 1777		
7243	004350	001400		.WORD 1400	:DIVP	
7244	004352	001777		.WORD 1777		
7245	004354	001000		.WORD 1000	:ASHP	
7246	004356	001777		.WORD 1777		
7247	004360	000400		.WORD 0400	:CVTLP	
7248	004362	001777		.WORD 1777		

			.SBTTL	TYPE 0 INDIRECTLY SPECIFIED PARAMETERS	
			:TYPE 0 ENTRY	- INDIRECTLY SPECIFIED PARAMETERS	
7250			INSTID:	.WORD 0	:UNASSIGNED
7251				.BYTE 6	:MOVC
7252				.BYTE 0	
7253	004364	000000		.BYTE 6	:MOVRC
7254	004366	006		.BYTE 0	
7255	004367	000		.BYTE 6	:MOVTC
7256	004370	006		.BYTE 0	
7257	004371	000		.BYTE 6	:MOVTC
7258	004372	006		.BYTE 15	
7259	004373	015		.BYTE 4	:LOCC
7260	004374	004		.BYTE 0	
7261	004375	000		.BYTE 4	:SKPC
7262	004376	004		.BYTE 0	
7263	004377	000		.BYTE 6	:SCANC
7264	004400	006		.BYTE 11	
7265	004401	011		.BYTE 6	:SPANC
7266	004402	006		.BYTE 11	
7267	004403	011		.BYTE 6	:CMPC
7268	004404	006		.BYTE 11	
7269	004405	011		.BYTE 6	:MATCHC
7270	004406	006		.BYTE 11	
7271	004407	011		.BYTE 7	:ADDN
7272	004410	007		.BYTE 12	
7273	004411	012		.BYTE 7	:SUBN
7274	004412	007		.BYTE 12	
7275	004413	012		.BYTE 6	:CMPN
7276	004414	006		.BYTE 11	
7277	004415	011		.BYTE 3	:CVTNL
7278	004416	003		.BYTE 0	
7279	004417	000		.BYTE 5	:CVTPN
7280	004420	005		.BYTE 0	
7281	004421	000		.BYTE 5	:CVTNP
7282	004422	005		.BYTE 0	
7283	004423	000		.BYTE 6	:ASHN
7284	004424	006		.BYTE 0	
7285	004425	000		.BYTE 0	:CVTLN
7286	004426	000		.BYTE 0	
7287	004427	000		.BYTE 7	:ADDP
7288	004430	007		.BYTE 12	
7289	004431	012		.BYTE 7	:SUBP
7290	004432	007		.BYTE 12	
7291	004433	012		.BYTE 6	:CMPP
7292	004434	006		.BYTE 11	
7293	004435	011		.BYTE 3	:CVTPL
7294	004436	003		.BYTE 0	
7295	004437	000		.BYTE 7	:MULP
7296	004440	007		.BYTE 12	
7297	004441	012		.BYTE 7	:DIVP
7298	004442	007		.BYTE 12	
7299	004443	012		.BYTE 6	:ASHP
7300	004444	006		.BYTE 0	
7301	004445	000		.BYTE 0	:CVTLP
7302	004446	000		.BYTE 0	
7303	004447	000		.BYTE 0	

7305				
7306				
7307				
7308				
7309	004450			
7311	004450	047515	041526	000000
7312	004456	072352		
7313	004460	047515	041526	000061
7314	004466	072426		
7315	004470	047515	041526	000062
7316	004476	072502		
7317	004500	047514	041503	000000
7318	004506	072556		
7319	004510	047514	041503	000061
7320	004516	072632		
7321	004520	047514	041503	000062
7322	004526	072706		
7323	004530	046503	041520	000000
7324	004536	072762		
7325	004540	046503	041520	000061
7326	004546	073036		
7327	004550	046503	041520	000062
7328	004556	073112		
7329	004560	047515	051126	000103
7330	004566	073166		
7331	004570	047515	051126	030503
7332	004576	073242		
7333	004600	047515	051126	031103
7334	004606	073316		
7335	004610	047515	052126	000103
7336	004616	073372		
7337	004620	047515	052126	030503
7338	004626	073446		
7339	004630	047515	052126	031103
7340	004636	073522		
7341	004640	045523	041520	000000
7342	004646	073576		
7343	004650	045523	041520	000061
7344	004656	073652		
7345	004660	045523	041520	000062
7346	004666	073726		
7347	004670	040515	041524	000000
7348	004676	074002		
7349	004700	040515	041524	000061
7350	004706	074056		
7351	004710	040515	041524	000062
7352	004716	074132		
7353	004720	041523	047101	000103
7354	004726	074206		
7355	004730	041523	047101	030503
7356	004736	074262		
7357	004740	041523	047101	031103
7358	004746	074336		
7359	004750	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>
```

7360	004756	074412			.WORD ISPAN
7361	004760	050123	047101	030503	.ASCII /SPANC1/
7362	004766	074466			.WORD ISPAN1
7363	004770	050123	047101	031103	.ASCII /SPANC2/
7364	004776	074542			.WORD ISPAN2
7365	005000	053103	050124	000116	.ASCII /CVTPN/<0>
7366	005006	074616			.WORD ICPZ
7367	005010	053103	050124	030516	.ASCII /CVTPN1/
7368	005016	074672			.WORD ICPZ1
7369	005020	053103	047124	000120	.ASCII /CVTNP/<0>
7370	005026	074746			.WORD ICZP
7371	005030	053103	047124	030520	.ASCII /CVTNP1/
737	005036	075022			.WORD ICZP1
737	005040	053103	047124	031120	.ASCII /CVTNP2/
737	005046	075076			.WORD ICZP2
7375	005050	053103	046124	000120	.ASCII /CVTLP/<0>
7376	005056	075152			.WORD ICLP
7377	005060	053103	046124	030520	.ASCII /CVTLP1/
7378	005066	075226			.WORD ICLP1
7379	005070	053103	046124	031120	.ASCII /CVTLP2/
7380	0C 076	075302			.WORD ICLP2
7381	005100	053103	046124	000116	.ASCII /CVTLN/<0>
7382	005106	075356			.WORD ICLZ
7383	005110	053103	046124	030516	.ASCII /CVTLN1/
7384	005116	075432			.WORD ICLZ1
7385	005120	053103	050124	000114	.ASCII /CVTPL/<0>
7386	005126	075506			.WORD ICPL
7387	005130	053103	050124	030514	.ASCII /CVTPL1/
7388	005136	075562			.WORD ICPL1
7389	005140	053103	050124	031114	.ASCII /CVTPL2/
7390	005146	075636			.WORD ICPL2
7391	005150	053103	050124	031514	.ASCII /CVTPL3/
7392	005156	075712			.WORD ICPL3
7393	005160	053103	047124	000114	.ASCII /CVTNL/<0>
7394	005166	075766			.WORD ICZL
7395	005170	053103	047124	030514	.ASCII /CVTNL1/
7396	005176	076042			.WORD ICZL1
7397	005200	053103	047124	031114	.ASCII /CVTNL2/
7398	005206	076116			.WORD ICZL2
7399	005210	042101	050104	000000	.ASCII /ADDP/<0><0>
7400	005216	076172			.WORD IADDP
7401	005220	042101	050104	000061	.ASCII /ADDP1/<0>
7402	005226	076246			.WORD IADDP1
7403	005230	042101	050104	000062	.ASCII /ADDP2/<0>
7404	005236	076322			.WORD IADDP2
7405	005240	042101	050104	000063	.ASCII /ADDP3/<0>
7406	005246	076376			.WORD IADDP3
7407	005250	042101	050104	000064	.ASCII /ADDP4/<0>
7408	005256	076452			.WORD IADDP4
7409	005260	042101	047104	000000	.ASCII /ADDN/<0><0>
7410	005266	076526			.WORD IADDN
7411	005270	042101	047104	000061	.ASCII /ADDN1/<0>
7412	005276	076602			.WORD IADDN1
7413	005300	042101	047104	000062	.ASCII /ADDN2/<0>

7414	005306	076656			.WORD IADDN2
7415	005310	042101	047104	000063	.ASCII /ADDN3/<0>
7416	005316	076732			.WORD IADDN3
7417	005320	042101	047104	000064	.ASCII /ADDN4/<0>
7418	005326	077006			.WORD IADDN4
7419	005330	052523	050102	000000	.ASCII /SUBP/<0><0>
7420	005336	077136			.WORD ISUBP
7421	005340	052523	050102	000061	.ASCII /SUBP1/<0>
7422	005346	077212			.WORD ISUBP1
7423	005350	052523	050102	000062	.ASCII /SUBP2/<0>
7424	005356	077266			.WORD ISUBP2
7425	005360	052523	050102	000063	.ASCII /SUBP3/<0>
7426	005366	077342			.WORD ISUBP3
7427	005370	052523	050102	000064	.ASCII /SUBP4/<0>
7428	005376	077416			.WORD ISUBP4
7429	005400	052523	047102	000000	.ASCII /SUBN/<0><0>
7430	005406	077472			.WORD ISUBN
7431	005410	052523	047102	000061	.ASCII /SUBN1/<0>
7432	005416	077546			.WORD ISUBN1
7433	005420	052523	047102	000062	.ASCII /SUBN2/<0>
7434	005426	077622			.WORD ISUBN2
7435	005430	052523	047102	000063	.ASCII /SUBN3/<0>
7436	005436	077676			.WORD ISUBN3
7437	005440	052523	047102	000064	.ASCII /SUBN4/<0>
7438	005446	077752			.WORD ISUBN4
7439	005450	046503	050120	000000	.ASCII /CMPP/<0><0>
7440	005456	100026			.WORD ICMPP
7441	005460	046503	050120	000061	.ASCII /CMPP1/<0>
7442	005466	100102			.WORD ICMPP1
7443	005470	046503	050120	000062	.ASCII /CMPP2/<0>
7444	005476	100156			.WORD ICMPP2
7445	005500	046503	050120	000063	.ASCII /CMPP3/<0>
7446	005506	100232			.WORD ICMPP3
7447	005510	046503	050120	000064	.ASCII /CMPP4/<0>
7448	005516	100306			.WORD ICMPP4
7449	005520	046503	047120	000000	.ASCII /CMPN/<0><0>
7450	005526	100362			.WORD ICMPN
7451	005530	046503	047120	000061	.ASCII /CMPN1/<0>
7452	005536	100436			.WORD ICMPN1
7453	005540	046503	047120	000062	.ASCII /CMPN2/<0>
7454	005546	100512			.WORD ICMPN2
7455	005550	046503	047120	000063	.ASCII /CMPN3/<0>
7456	005556	100566			.WORD ICMPN3
7457	005560	046503	047120	000064	.ASCII /CMPN4/<0>
7458	005566	100642			.WORD ICMPN4
7459	005570	051501	050110	000000	.ASCII /ASHP/<0><0>
7460	005576	100716			.WORD IASHP
7461	005600	051501	050110	000061	.ASCII /ASHP1/<0>
7462	005606	100772			.WORD IASHP1
7463	005610	051501	050110	000062	.ASCII /ASHP2/<0>
7464	005616	101046			.WORD IASHP2
7465	005620	051501	047110	000000	.ASCII /ASHN/<0><0>
7466	005626	101122			.WORD IASHN
7467	005630	051501	047110	000061	.ASCII /ASHN1/<0>



7468	005636	101176			.WORD IASHN1
7469	005640	051501	047110	000062	.ASCII /ASHN2/<0>
7470	005646	101252			.WORD IASHN2
7471	005650	052515	050114	000000	.ASCII /MULP/<0><0>
7472	005656	101326			.WORD IMULP
7473	005660	052515	050114	000061	.ASCII /MULP1/<0>
7474	005666	101402			.WORD IMULP1
7475	005670	052515	050114	000062	.ASCII /MULP2/<0>
7476	005676	101456			.WORD IMULP2
7477	005700	052515	050114	000063	.ASCII /MULP3/<0>
7478	005706	101532			.WORD IMULP3
7479	005710	052515	050114	000064	.ASCII /MULP4/<0>
7480	005716	101606			.WORD IMULP4
7481	005720	052515	050114	000065	.ASCII /MULP5/<0>
7482	005726	101662			.WORD IMULP5
7483	005730	052515	050114	000066	.ASCII /MULP6/<0>
7484	005736	101736			.WORD IMULP6
7485	005740	052515	050114	000067	.ASCII /MULP7/<0>
7486	005746	102012			.WORD IMULP7
7487	005750	044504	050126	000000	.ASCII /DIVP/<0><0>
7488	005756	102066			.WORD IDIVP
7489	005760	044504	050126	000061	.ASCII /DIVP1/<0>
7490	005766	102142			.WORD IDIVP1
7491	005770	044504	050126	000062	.ASCII /DIVP2/<0>
7492	005776	102216			.WORD IDIVP2
7493	006000	044504	050126	000063	.ASCII /DIVP3/<0>
7494	006006	102272			.WORD IDIVP3
7495	006010	044504	050126	000064	.ASCII /DIVP4/<0>
7496	006016	102346			.WORD IDIVP4
7497	006020	044504	050126	000065	.ASCII /DIVP5/<0>
7498	006026	102422			.WORD IDIVP5
7499	006030	044504	050126	000066	.ASCII /DIVP6/<0>
7500	006036	102476			.WORD IDIVP6
7502	006040	031114	000104	000000	.ASCII /L2D/<0><0><0>
7503	006046	072222			.WORD IL2D
7504	006050	031514	000104	000000	.ASCII /L3D/<0><0><0>
7505	006056	072276			.WORD IL3D
7506	006060	000000			.WORD 0
7507	006062	000000			.WORD 0
7508	006064	000000			.WORD 0
7509	006066	000000			.WORD 0

```
7511          .SBTTL          RANDOM EXERCISE MASK TABLES
7512          :
7513          :RANDOM EXERCISING MASK TABLES
7514          :
7515          :
7516 006070    MMOVC:
7517 006070    MCMPC:
7518 006070    MMOVRC:
7520 006070 174000 .WORD BS2          :IP1 MASK          (LEN)
7521 006072 176000 .WORD BS4          :IP2          (ADR)
7522 006074 174000 .WORD BS2          :IP3          (LEN)
7523 006076 174000 .WORD BS2          :IP4          (ADR)
7531 006100 177400 .WORD BY           :IP5          (FILL)
7532 006102 125252 .WORD EOT
7533
7534
7535
7536
7537 006104    MMOVTC:
7539 006104 174000 .WORD BS2          :IP1 MASK          (LEN)
7540 006106 176000 .WORD BS4          :IP2          (ADR)
7541 006110 174000 .WORD BS2          :IP3          (LEN)
7542 006112 002000 .WORD MPO2000     :IP4          (ADR)
7550 006114 177400 .WORD BY           :IP5          (FILL)
7551 006116 177777 .WORD BSNULL      :IP6
7552 006120 177777 .WORD BSNULL      :IP7
7553 006122 177777 .WORD BSNULL      :IP10
7554 006124 177777 .WORD BSNULL      :IP11
7555 006126 177777 .WORD BSNULL      :IP12
7556 006130 177777 .WORD BSNULL      :IP13
7557 006132 177777 .WORD BSNULL      :IP14
7558 006134 022000 .WORD MPO2200     :IP15          (TABLE ADR)
7559 006136 125252 .WORD EOT
7560
```

```
7562  
7563  
7564  
7565  
7566 006140  
7567 006140  
7569 006140 174000  
7570 006142 174000  
7576 006144 177400  
7577 006146 125252  
7578  
7579  
7580  
7581 006150  
7582 006150  
7584 006150 174000  
7585 006152 174000  
7591 006154 177400  
7592 006156 177400  
7594 006160 174000  
7599 006162 125252  
7600  
7601  
7602 006164  
7604 006164 177000  
7605 006166 176000  
7606 006170 174000  
7607 006172 174000  
7615 006174 177400  
7616 006176 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

```

7618
7619
7620      ;RANDOM EXERCISE MODE MASK TABLES (CONTINUED)
7621      ;
7622
7623      006200      MCVTNP:
7624      006200      177740      MCVTPN: .WORD BS128      ;IP1 MASK      (LEN)
7626      006202      176000      .WORD BS4      ;IP2      (ADR)
7631      006204      177740      .WORD BS128      ;IP3      (LEN)
7633      006206      004000      .WORD MPO4000      ;IP4      (ADR)
7638      006210      152525      .WORD DSCPTR      ;IP5      (DESC POINTER)
7639      006212      125252      .WORD EOT
7640
7641
7642      006214      MCVTLP:
7643      006214      000000      MCVTLN: .WORD WD      ;IP1 MASK      (LONG-HIGH)
7644      006216      000000      .WORD WD      ;IP2      (LONG-LOW)
7645      006220      177740      .WORD BS128      ;IP3      (LEN)
7647      006222      174000      .WORD BS2      ;IP4      (ADR)
7652      006224      125252      .WORD EOT
7653
7654
7655      006226      MCVTPL:
7656      006226      177740      MCVTNL: .WORD BS128      ;IP1 MASK      (LEN)
7658      006230      174000      .WORD BS2      ;IP2      (ADR)
7663      006232      152525      .WORD DSCPTR      ;IP3      (DESC POINTER)
7664      006234      177777      .WORD BSNULL      ;IP4
7665      006236      177777      .WORD BSNULL      ;IP5
7666      006240      177400      .WORD BY      ;IP6      (DATA)
7667      006242      125252      .WORD EOT

```

```

7669
7670
7671      ;RANDOM EXERCISE MODE MASK TABLES (CONTINUED)
7672      ;
7673
7674 006244      MADDP:
7675 006244      MADDN:
7676 006244      MSUBP:
7677 006244      MSUBN:
7678 006244      MMULP:
7679 006244      MDIVP:
7680 006244 177740      .WORD BS128      ;IP1 MASK      (LEN)
7682 006246 177000      .WORD BS8        ;IP2            (ADR)
7683 006250 177740      .WORD BS128      ;IP3            (LEN)
7684 006252 002000      .WORD MPO2000    ;IP4            (ADR)
7685 006254 177740      .WORD BS128      ;IP5            (LEN)
7686 006256 004040      .WORD MPO4040    ;IP6            (ADR)
7695 006260 152525      .WORD DSCPTR     ;IP7            (DESC POINTER)
7696 006262 177777      .WORD BSNUL      ;IP10           (LEN)
7697 006264 177777      .WORD BSNUL      ;IP11           (ADR)
7698 006266 152525      .WORD DSCPTR     ;IP12           (DESC POINTER)
7699 006270 125252      .WORD EOT
7700
7701
7702 006272      MCMPN:
7703 006272      MCMPP:
7704 006272 177740      .WORD BS128      ;IP1 MASK      (LEN)
7706 006274 176000      .WORD BS4        ;IP2            (ADR)
7707 006276 177740      .WORD BS128      ;IP3            (LEN)
7708 006300 004000      .WORD MPO4000    ;IP4            (ADR)
7715 006302 177400      .WORD BY         ;IP5            (DATA)
7716 006304 152525      .WORD DSCPTR     ;IP6            (DESC POINTER)
7717 006306 177777      .WORD BSNUL      ;IP7            (LEN)
7718 006310 177777      .WORD BSNUL      ;IP10           (ADR)
7719 006312 152525      .WORD DSCPTR     ;IP11           (DESC POINTER)
7720 006314 125252      .WORD EOT
  
```

7722  
 7723  
 7724  
 7725  
 7726  
 7727 006316  
 7728 006316  
 7729 006316 177740  
 7731 006320 176000  
 7732 006322 170000  
 7733 006324 177740  
 7734 006326 004000  
 7742 006330 152525  
 7743 006332 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



7773  
7774  
7775  
7776  
7777 006334  
7778 006334  
7779 006334 151600  
7780 006336 010100  
7781 006340 010201  
7782 006342 010302  
7783 006344 010504  
7784 006346 020403  
7785 006350 031001  
7786 006352 031303  
7787 006354 041001  
7788 006356 041323  
7789 006360 051300  
7790 006362 060000  
7791 006364 071123  
7792 006366 001312  
7793 006370 070601  
7794 006372 001007  
7795 006374 100000  
7796 006376 111300  
7797 006400 122400  
7798 006402 001300  
7799 006404 000000  
7800 006406 007000  
7801 006410 137000  
7802 006412 140000  
7803  
7804

.SBTTL CIS INST FLOW TABLES  
:CIS INSTRUCTION FLOW TABLES

:  
XMOVCL  
XMOVRC:  
.WORD 151600  
.WORD 010100  
.WORD 010201  
.WORD 010302  
.WORD 010504  
.WORD 020403  
.WORD 031001  
.WORD 031303  
.WORD 041001  
.WORD 041323  
.WORD 051300  
.WORD 060000  
.WORD 071123  
.WORD 001312  
.WORD 070601  
.WORD 001007  
.WORD 100000  
.WORD 111300  
.WORD 122400  
.WORD 001300  
.WORD 000000  
.WORD 007000  
.WORD 137000  
.WORD 140000

:MOVCL FLOW TABLE S1=SRC, S2=DST  
: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 < 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.

7806  
7807 006414  
7808 006414 151600  
7809 006416 010100  
7810 006420 010201  
7811 006422 010302  
7812 006424 010504  
7813 006426 011505  
7814 006430 020403  
7815 006432 031001  
7816 006434 031303  
7817 006436 041001  
7818 006440 041323  
7819 006442 051300  
7820 006444 060000  
7821 006446 071123  
7822 006450 001312  
7823 006452 070601  
7824 006454 001007  
7825 006456 100000  
7826 006460 111300  
7827 006462 122450  
7828 006464 001300  
7829 006466 000000  
7830 006470 007000  
7831 006472 137000  
7832 006474 140000  
7833  
7834

:  
XMOVTC:  
.WORD 151600  
.WORD 010100  
.WORD 010201  
.WORD 010302  
.WORD 010504  
.WORD 011505  
.WORD 020403  
.WORD 031001  
.WORD 031303  
.WORD 041001  
.WORD 041323  
.WORD 051300  
.WORD 060000  
.WORD 071123  
.WORD 001312  
.WORD 070601  
.WORD 001007  
.WORD 100000  
.WORD 111300  
.WORD 122450  
.WORD 001300  
.WORD 000000  
.WORD 007000  
.WORD 137000  
.WORD 140000

: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  
:LOAD TR5 FROM PTP15 (TRANSLATION TABLE)  
: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.

7836  
7837  
7838 006476  
7839 006476 151500  
7840 006500 010100  
7841 006502 010201  
7842 006504 010302  
7843 006506 010504  
7844 006510 020403  
7845 006512 031001  
7846 006514 031303  
7847 006516 041001  
7848 006520 041323  
7849 006522 051300  
7850 006524 060000  
7851 006526 071123  
7852 006530 001312  
7853 006532 070601  
7854 006534 001007  
7855 006536 100000  
7856 006540 111300  
7857 006542 122400  
7858 006544 001300  
7859 006546 000000  
7860 006550 007000  
7861 006552 131370  
7862 006554 007000  
7863 006556 140000  
7864

XCMPC:

```
.WORD 151500
.WORD 010100
.WORD 010201
.WORD 010302
.WORD 010504
.WORD 020403
.WORD 031001
.WORD 031303
.WORD 041001
.WORD 041323
.WORD 051300
.WORD 060000
.WORD 071123
.WORD 001312
.WORD 070601
.WORD 001007
.WORD 100000
.WORD 111300
.WORD 122400
.WORD 001300
.WORD 000000
.WORD 007000
.WORD 131370
.WORD 007000
.WORD 140000

;CMPC FLOW TABLE S1=SRC1, S2=SRC2
;LOAD SPECIAL HANDLING WORD 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.
```

7866  
7867 006560  
7868 006560 151500  
7869 006562 010300  
7870 006564 010102  
7871 006566 010203  
7872 006570 010504  
7873 006572 020401  
7874 006574 031301  
7875 006576 031003  
7876 006600 041301  
7877 006602 041023  
7878 006604 051300  
7879 006606 060000  
7880 006610 070623  
7881 006612 001007  
7882 006614 071101  
7883 006616 001312  
7884 006620 100000  
7885 006622 111300  
7886 006624 122000  
7887 006626 001300  
7888 006630 000000  
7889 006632 000170  
7890 006634 131370  
7891 006636 001700  
7892 006640 140000  
7893

XMATCHC:

.WORD 151500  
.WORD 010300  
.WORD 010102  
.WORD 010203  
.WORD 010504  
.WORD 020401  
.WORD 031301  
.WORD 031003  
.WORD 041301  
.WORD 041023  
.WORD 051300  
.WORD 060000  
.WORD 070623  
.WORD 001007  
.WORD 071101  
.WORD 001312  
.WORD 100000  
.WORD 111300  
.WORD 122000  
.WORD 001300  
.WORD 000000  
.WORD 000170  
.WORD 131370  
.WORD 001700  
.WORD 140000

:MATCHC FLOW TABLE S1=SRC, S2=OBJ  
:LOAD SPECIAL HANDLING WORD FROM PTP15  
:LOAD TR0 FROM PTP03  
:LOAD TR2 FROM PTP01  
:LOAD TR3 FROM PTP02  
:LOAD TR4 FROM PTP05  
:GENERATE TR1 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.

7895  
7896  
7897 006642  
7898 006642  
7899 006642 151500  
7900 006644 010110  
7901 006646 010201  
7902 006650 010312  
7903 006652 010504  
7904 006654 020403  
7905 006656 031001  
7906 006660 031303  
7907 006662 041001  
7908 006664 041323  
7909 006666 051300  
7910 006670 060000  
7911 006672 075123  
7912 006674 074601  
7913 006676 100000  
7914 006700 111300  
7915 006702 122000  
7916 006704 001300  
7917 006706 000000  
7918 006710 007000  
7919 006712 137000  
7920 006714 140000  
7921

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

:CMPN 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.

7923  
7924 006716  
7925 006716  
7926 006716 151500  
7927 006720 010110  
7928 006722 010201  
7929 006724 010422  
7930 006726 010304  
7931 006730 020503  
7932 006732 031001  
7933 006734 031303  
7934 006736 041001  
7935 006740 041323  
7936 006742 051300  
7937 006744 060000  
7938 006746 071123  
7939 006750 001312  
7940 006752 074601  
7941 006754 100000  
7942 006756 111300  
7943 006760 122400  
7944 006762 001300  
7945 006764 000000  
7946 006766 007000  
7947 006770 133700  
7948 006772 007000  
7949 006774 140000  
7950  
7951

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



7953  
7954  
7955  
7956 006776  
7957 006776  
7958 006776 150700  
7959 007000 010100  
7960 007002 010201  
7961 007004 010304  
7962 007006 030601  
7963 007010 040601  
7964 007012 051000  
7965 007014 060000  
7966 007016 070401  
7967 007020 000605  
7968 007022 100000  
7969 007024 111000  
7970 007026 121400  
7971 007030 001000  
7972 007032 000000  
7973 007034 000170  
7974 007036 131700  
7975 007040 001700  
7976 007042 140000  
7977  
7978

XLOCC:  
XSKPC:  
.WORD 150700  
.WORD 010100  
.WORD 010201  
.WORD 010304  
.WORD 030601  
.WORD 040601  
.WORD 051000  
.WORD 060000  
.WORD 070401  
.WORD 000605  
.WORD 100000  
.WORD 111000  
.WORD 121400  
.WORD 001000  
.WORD 000000  
.WORD 000170  
.WORD 131700  
.WORD 001700  
.WORD 140000

:LOCC FLOW TABLE  
:SKPC FLOW TABLE  
:LOAD SPECIAL HANDLING WORD FROM PTP07  
:LOAD TR0 FROM PTP01  
:LOAD TR1 FROM PTP02  
:LOAD TR4 FROM PTP03  
:VERIFY THAT SRC.ADR-SRC.SURR.LEN >=20  
:VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN < TBLEN  
:ADD TEST BUFFER ADDRESS TO TR1  
:INITIALIZE TEST BUFFER  
:INSERT SRC & SRC SURR STRINGS IN BUFFER  
  
:COPY TEST BUFFER INTO EMULATION BUFFER  
:SETUP EMULATION OPERANDS & EMULATE INST.  
:SETUP CC & REGS AND EXECUTE CIS INST  
  
:COMPARE RESULTS  
  
:UPDATE POINTERS AND RETURN FOR NEXT  
: TEST CONDITION.

7980				
7981	007044		XADDP:	;ADDP FLOW TABLE
7982	007044		XADDN:	;ADDN FLOW TABLE
7983	007044		XSUBP:	;SUBP FLOW TABLE
7984	007044		XSUBN:	;SUBN FLOW TABLE
7985	007044		XMULP:	;MULP FLOW TABLE
7986	007044		XDIVP:	;DIVP FLOW TABLE
7987	007044	152100	.WORD 152100	;LOAD SPECIAL HANDLING FROM PTP21
7988	007046	010110	.WORD 010110	;LOAD TR0 FROM PTP01
7989	007050	010201	.WORD 010201	;LOAD TR1 FROM PTP02
7990	007052	010322	.WORD 010322	;LOAD TR2 FROM PTP03
7991	007054	010534	.WORD 010534	;LOAD TR4 FROM PTP05
7992	007056	010605	.WORD 010605	;TYPE 0 USE ONLY - LOAD TR5 FROM PTP06
7993	007060	020403	.WORD 020403	;GENERATE TR3 & TR5 FROM PTP04
7994	007062	031101	.WORD 031101	;VERIFY THAT SRC1.ADR-SRC1.SURR.LEN >= 20
7995	007064	031403	.WORD 031403	;VERIFY THAT SRC2.ADR-SRC2.SURR.LEN >= 20
7996	007066	031705	.WORD 031705	;VERIFY THAT DST.ADR-DST.SURR.LEN >=20
7997	007070	041101	.WORD 041101	;VERIFY THAT SRC1.ADR+SRC1.LEN+SRC1.SURR.LEN < TBLLEN
7998	007072	041423	.WORD 041423	;VERIFY THAT SRC2.ADR+SRC2.LEN+SRC2.SURR.LEN < TBLLEN
7999	007074	041745	.WORD 041745	;VERIFY THAT DST.ADR+DST.LEN+DST.SURR.LEN < TBLLEN
8000	007076	051350	.WORD 051350	;ADD TEST BUFFER ADDRESS TO TR1,TR3, & TR5
8001	007100	060000	.WORD 060000	;INITIALIZE TEST BUFFER
8002	007102	071545	.WORD 071545	;INSERT DST & DST.SURR STRINGS IN TEST BUFFER
8003	007104	001716	.WORD 001716	
8004	007106	074701	.WORD 074701	;INSERT SRC1 IN TEST BUFFER
8005	007110	075223	.WORD 075223	;INSERT SRC2 IN TEST BUFFER
8006	007112	100000	.WORD 100000	;COPY TEST BUFFER INTO EMULATION BUFFER
8007	007114	111350	.WORD 111350	;SETUP EMULATION OPERANDS & EMULATE INST
8008	007116	123000	.WORD 123000	;SETUP CC & REGS AND EXECUTE CIS INST.
8009	007120	001350	.WORD 001350	
8010	007122	000000	.WORD 000000	
8011	007124	007000	.WORD 007000	
8012	007126	135700	.WORD 135700	;COMPARE RESULTS
8013	007130	007000	.WORD 007000	
8014	007132	140000	.WORD 140000	;UPDATE POINTERS FOR NEXT TEST CONDITION
8015				; AND RETURN TO START EXECUTING NEXT
8016				; TEST CONDITION

8018  
8019 007134  
8020 007134  
8021 007134 151500  
8022 007136 010100  
8023 007140 010201  
8024 007142 010302  
8025 007144 010303  
8026 007146 010404  
8027 007150 020505  
8028 007152 031001  
8029 007154 031305  
8030 007156 041001  
8031 007160 041325  
8032 007162 051500  
8033 007164 060000  
8034 007166 071125  
8035 007170 070601  
8036 007172 001007  
8037 007174 100000  
8038 007176 111500  
8039 007200 122000  
8040 007202 001500  
8041 007204 000000  
8042 007206 000170  
8043 007210 131570  
8044 007212 001700  
8045 007214 140000  
8046

XSCANC:  
XSPANC:  
.WORD 151500  
.WORD 010100  
.WORD 010201  
.WORD 010302  
.WORD 010303  
.WORD 010404  
.WORD 020505  
.WORD 031001  
.WORD 031305  
.WORD 041001  
.WORD 041325  
.WORD 051500  
.WORD 060000  
.WORD 071125  
.WORD 070601  
.WORD 001007  
.WORD 100000  
.WORD 111500  
.WORD 122000  
.WORD 001500  
.WORD 000000  
.WORD 000170  
.WORD 131570  
.WORD 001700  
.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

8048  
8049  
8050 007216  
8051 007216  
8052 007216 151400  
8053 007220 010110  
8054 007222 010201  
8055 007224 010322  
8056 007226 020403  
8057 007230 030701  
8058 007232 031203  
8059 007234 040701  
8060 007236 041223  
8061 007240 051300  
8062 007242 060000  
8063 007244 071023  
8064 007246 001211  
8065 007250 074501  
8066 007252 100000  
8067 007254 111300  
8068 007256 122000  
8069 007260 001300  
8070 007262 000000  
8071 007264 007000  
8072 007266 133700  
8073 007270 007000  
8074 007272 140000  
8075

XCVTPN:  
XCVTNP:  
.WORD 151400  
.WORD 010110  
.WORD 010201  
.WORD 010322  
.WORD 020403  
.WORD 030701  
.WORD 031203  
.WORD 040701  
.WORD 041223  
.WORD 051300  
.WORD 060000  
.WORD 071023  
.WORD 001211  
.WORD 074501  
.WORD 100000  
.WORD 111300  
.WORD 122000  
.WORD 001300  
.WORD 000000  
.WORD 007000  
.WORD 133700  
.WORD 007000  
.WORD 140000

:CVTPN FLOW TABLE  
:CVTNP FLOW TABLE  
:LOAD SPECIAL HANDLING FROM PTP14  
:LOAD TR0 FROM PTP01  
:LOAD TR1 FROM PTP02  
:LOAD TR2 FROM PTP03  
:GENERATE TR3 FROM PTP04  
: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 & TR3  
:INITIALIZE TEST BUFFER  
:INSERT DST & DST SURR STRINGS IN BUFFER  
  
:INSERT SRC STRING IN BUFFER  
:COPY TEST BUFFER INTO EMULATION BUFFER  
:SETUP EMULATION OPERANDS & EMULAT INST.  
:SETUP CC REGS & EXECUTE CIS INST.  
  
:COMPARE RESULTS  
:UPDATE PTRS FROM NEXT TEST CONDITION

8077  
8078  
8079 007274  
8080 007274  
8081 007274 151000  
8082 007276 010102  
8083 007300 010203  
8084 007302 010310  
8085 007304 010401  
8086 007306 030701  
8087 007310 040701  
8088 007312 051000  
8089 007314 060000  
8090 007316 070501  
8091 007320 000706  
8092 007322 100000  
8093 007324 111000  
8094 007326 122000  
8095 007330 001300  
8096 007332 000000  
8097 007334 007000  
8098 007336 131700  
8099 007340 007000  
8100 007342 140000  
8101  
8102  
8103

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

8105  
8106  
8107 007344  
8108 007344  
8109 007344 150700  
8110 007346 010110  
8111 007350 010201  
8112 007352 010604  
8113 007354 030501  
8114 007356 040501  
8115 007360 051000  
8116 007362 060000  
8117 007364 074301  
8118 007366 100000  
8119 007370 111000  
8120 007372 122000  
8121 007374 001000  
8122 007376 007300  
8123 007400 007000  
8124 007402 137000  
8125 007404 140000  
8126  
8127

XCVTPL:  
XCVTNL:  
.WORD 150700  
.WORD 010110  
.WORD 010201  
.WORD 010604  
.WORD 030501  
.WORD 040501  
.WORD 051000  
.WORD 060000  
.WORD 074301  
.WORD 100000  
.WORD 111000  
.WORD 122000  
.WORD 001000  
.WORD 007300  
.WORD 007000  
.WORD 137000  
.WORD 140000

:CVTPL FLOW TABLE  
:CVTNL FLOW TABLE  
:LOAD SPECIAL HANDLING FROM PTP07  
:LOAD TR0 FROM PTP01  
:LOAD TR1 FROM PTP02  
:LOAD TR4 FROM PTP06  
:VERIFY THAT SRC.ADR-SRC.SURR.LEN.+20  
:VERIFY THAT SRC.ADR+SRC.LEN+SRC.SURR.LEN<TBLEN  
:ADD TEST BUFFER ADDRESS TO TR1  
:INITIALIZE TEST BUFFER  
:INSERT SRC STRING IN TEST 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



8129 007406  
8130 007406  
8131 007406 010100  
8132 007410 010201  
8133 007412 010302  
8134 007414 010403  
8135 007416 010504  
8136 007420 010605  
8137 007422 060000  
8138 007424 020700  
8139 007426 100000  
8140 007430 110000  
8141 007432 120000  
8142 007434 137000  
8143 007436 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.

```

8145          .SBTTL GLOBAL TEXT SECTION
8146
8147          ; FORMAT STATEMENTS USED IN PRINT CALLS
8148          ;
8149          ;
8150          ;MESSAGES
8151 007440 040445 046440 053117 AMOVC: .ASCIZ /%A MOVC/
      007446 000103
8152 007450 040445 046440 053117 AMOVRC: .ASCIZ /%A MOVRC/
      007456 041522 000
8153 007461 045 020101 047515 AMOVTC: .ASCIZ /%A MOVTC/
      007466 052126 000103
8154 007472 040445 046040 041517 ALOCC: .ASCIZ /%A LOCC/
      007500 000103
8155 007502 040445 051440 050113 ASKPC: .ASCIZ /%A SKPC/
      007510 000103
8156 007512 040445 051440 040503 ASCANC: .ASCIZ /%A SCANC/
      007520 041516 000
8157 007523 045 020101 050123 ASPANC: .ASCIZ /%A SPANC/
      007530 047101 000103
8158 007534 040445 041440 050115 ACMPC: .ASCIZ /%A CMPC/
      007542 000103
8159 007544 040445 046440 052101 AMATCHC: .ASCIZ /%A MATC/
      007552 000103
8160 007554 040445 040440 042104 AADDN: .ASCIZ /%A ADDN/
      007562 000116
8161 007564 040445 051440 041125 ASUBN: .ASCIZ /%A SUBN/
      007572 000116
8162 007574 040445 041440 050115 ACMPN: .ASCIZ /%A CMPN/
      007602 000116
8163 007604 040445 041440 052126 ACVTNL: .ASCIZ /%A CVTNL/
      007612 046116 000
8164 007615 045 020101 053103 ACVTPN: .ASCIZ /%A CVTPN/
      007622 050124 000116
8165 007626 040445 041440 052126 ACVTNP: .ASCIZ /%A CVTNP/
      007634 050116 000
8166 007637 045 020101 051501 AASHN: .ASCIZ /%A ASHN/
      007644 047110 000
8167 007647 045 020101 053103 ACVTLN: .ASCIZ /%A CVTLN/
      007654 046124 000116
8168 007660 040445 040440 042104 AADDP: .ASCIZ /%A ADDP/
      007666 000120
8169 007670 040445 051440 041125 ASUBP: .ASCIZ /%A SUBP/
      007676 000120
8170 007700 040445 041440 050115 ACMPP: .ASCIZ /%A CMPP/
      007706 000120
8171 007710 040445 041440 052126 ACVTPL: .ASCIZ /%A CVTPL/
      007716 046120 000
8172 007721 045 020101 052515 AMULP: .ASCIZ /%A MULP/
      007726 050114 000
8173 007731 045 020101 044504 ADIVP: .ASCIZ /%A DIVP/
      007736 050126 000
8174 007741 045 020101 051501 AASHP: .ASCIZ /%A ASHP/
      007746 050110 000

```

8175	007751	045	020101	053103	ACVTLP:	.ASCIZ	/%A CVTLP/
	007756	046124	000120				
8176	007762	040445	046040	042062	AL2D:	.ASCIZ	/%A L2D%01/
	007770	047445	000061				
8177	007774	040445	046040	042063	AL3D:	.ASCIZ	/%A L3D%01/
	010002	047445	000061				
8178							
8179	010006	051445	031462	040445	FORM1:	.ASCIZ	/%S23%ASL%55%ASA%55%ADL%55%ADA%55%AF%524%ANZVC%N/
	010014	046123	051445	022465			
	010022	051501	022501	032523			
	010030	040445	046104	051445			
	010036	022465	042101	022501			
	010044	032523	040445	022506			
	010052	031123	022464	047101			
	010060	053132	022503	000116			
8180	010066	051445	031462	040445	FORM2:	.ASCIZ	/%S23%ASL%55%ASA%55%ADL%55%ADA%55%AF%56%AT%515%ANZVC%N/
	010074	046123	051445	022465			
	010102	051501	022501	032523			
	010110	040445	046104	051445			
	010116	022465	042101	022501			
	010124	032523	040445	022506			
	010132	033123	040445	022524			
	010140	030523	022465	047101			
	010146	053132	022503	000116			
8181	010154	051445	031462	040445	FORM3:	.ASCIZ	/%S23%ASL%55%ASA%523%ACHAR%521%ANZVC%N/
	010162	046123	051445	022465			
	010170	051501	022501	031123			
	010176	022463	041501	040510			
	010204	022522	031123	022461			
	010212	047101	053132	022503			
	010220	000116					
8182	010222	051445	031462	040445	FORM4:	.ASCIZ	/%S23%ASL%55%ASA%523%AMASK%53%AT%515%ANZVC%N/
	010230	046123	051445	022465			
	010236	051501	022501	031123			
	010244	022463	046501	051501			
	010252	022513	031523	040445			
	010260	022524	030523	022465			
	010266	047101	053132	022503			
	010274	000116					
8183	010276	051445	031462	040445	FORM5:	.ASCIZ	/%S23%AS1L%54%AS1A%54%AS2L%54%AS2A%54%AF%524%ANZVC%N/
	010304	030523	022514	032123			
	010312	040445	030523	022501			
	010320	032123	040445	031123			
	010326	022514	032123	040445			
	010334	031123	022501	032123			
	010342	040445	022506	031123			
	010350	022464	047101	053132			
	010356	022503	000116				
8184	010362	051445	031462	040445	FORM6:	.ASCIZ	/%S23%ASL%55%ASA%55%AOL%55%AOL%532%ANZVC%N/
	010370	046123	051445	022465			
	010376	051501	022501	032523			
	010404	040445	046117	051445			
	010412	022465	047501	022501			
	010420	031523	022462	047101			

8185	010426	053132	022503	000116		
	010434	051445	031462	040445	FORM7:	.ASCIZ /%S23%AS1L%S4%AS1A%S4%AS2L%S4%AS2A%S4%ADL%S5%ADA%S14%ANZVC%N/
	010442	030523	022514	032123		
	010450	040445	030523	022501		
	010456	032123	040445	031123		
	010464	022514	032123	040445		
	010472	031123	022501	032123		
	010500	040445	046104	051445		
	010506	022465	042101	022501		
	010514	030523	022464	047101		
8186	010522	053132	022503	000116	FORM8:	.ASCIZ /%S23%AS1L%S4%AS1A%S4%AS2L%S4%AS2A%S32%ANZVC%N/
	010530	051445	031462	040445		
	010536	030523	022514	032123		
	010544	040445	030523	022501		
	010552	032123	040445	031123		
	010560	022514	032123	040445		
	010566	031123	022501	031523		
	010574	022462	047101	053132		
8187	010602	022503	000116		FORM9:	.ASCIZ /%S23%ASL%S5%ASA%S5%AD.H%S4%AD.L%S32%ANZVC%N/
	010606	051445	031462	040445		
	010614	046123	051445	022465		
	010622	051501	022501	032523		
	010630	040445	027104	022510		
	010636	032123	040445	027104		
	010644	022514	031523	022462		
	010652	047101	053132	022503		
8188	010660	000116			FORM10:	.ASCIZ /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S32%ANZVC%N/
	010662	051445	031462	040445		
	010670	046123	051445	022465		
	010676	051501	022501	032523		
	010704	040445	046104	051445		
	010712	022465	042101	022501		
	010720	031523	022462	047101		
	010726	053132	022503	000116		
8189	010734	051445	031462	040445	FORM11:	.ASCIZ /%S23%ASL%S5%ASA%S5%ADL%S5%ADA%S5%AR,S%S22%ANZVC%N/
	010742	046123	051445	022465		
	010750	051501	022501	032523		
	010756	040445	046104	051445		
	010764	022465	042101	022501		
	010772	032523	040445	026122		
	011000	022523	031123	022462		
	011006	047101	053132	022503		
	011014	000116				
8190	011016	051445	031462	040445	FORM12:	.ASCIZ /%S23%ADL%S5%ADA%S5%AS.H%S4%AS.L%S32%ANZVC%N/
	011024	046104	051445	022465		
	011032	042101	022501	032523		
	011040	040445	027123	022510		
	011046	032123	040445	027123		
	011054	022514	031523	022462		
	011062	047101	053132	022503		
	011070	000116				
8191	011072	040445	044440	050116	INREG:	.ASCIZ +%A INPUT R0-R6,CC/ +
	011100	052125	020040	051040		
	011106	026460	033122	041454		

8192	011114	027503	000040			
	011120	040445	044440	050116	INMEM: .ASCIZ	+%A INPUTS IN MEMORY/ +
	011126	052125	020123	047111		
	011134	046440	046505	051117		
	011142	027531	000040			
8193	011146	040445	042440	050130	EMOUT: .ASCIZ	+%A EXP OUT R0-R6,CC/ +
	011154	047440	052125	051040		
	011162	026460	033122	041454		
	011170	027503	000040			
8194	011174	047445	022466	030523	FORM13: .ASCIZ	/%06%S1%06%S1%06%S1%06%S1%06%S1%06%S1%06%S1%Y4%N/
	011202	047445	022466	030523		
	011210	047445	022466	030523		
	011216	047445	022466	030523		
	011224	047445	022466	030523		
	011232	047445	022466	030523		
	011240	047445	022466	030523		
	011246	054445	022464	000116		
8195	011254	047445	022466	030523	FORM14: .ASCIZ	/%06%S1%06%S1%06%S1%06%S1%06%S1%06%S1%Y4%N/
	011262	047445	022466	030523		
	011270	047445	022466	030523		
	011276	047445	022466	030523		
	011304	047445	022466	030523		
	011312	047445	022466	030523		
	011320	047445	022466	030523		
	011326	054445	022464	000116		
8196	011334	040445	040440	052103	ACOUT: .ASCIZ	+%A ACT OUT R0-R6,CC/ +
	011342	047440	052125	051040		
	011350	026460	033122	041454		
	011356	027503	000040			
8197	011362	047445	022466	030523	FORM15: .ASCIZ	/%06%S1/
	011370	000				
8198	011371	045	033523	000	FORM16: .ASCIZ	/%S7/
8199	011375	045	033123	022461	FORM17: .ASCIZ	/%S61%Y4/
	011402	032131	000			
8200	011405	045	032131	000	FORM18: .ASCIZ	/%Y4/
8201	011411	045	022516	020101	ERUFO: .ASCIZ	+%N%A EXP BUFFER %06%A/ %03%N+
	011416	054105	020120	052502		
	011424	043106	051105	022440		
	011432	033117	040445	020057		
	011440	047445	022463	000116		
8202	011446	040445	040440	052103	ABUFO: .ASCIZ	+%A ACT BUFFER %06%A/ %03+
	011454	041040	043125	042506		
	011462	020122	047445	022466		
	011470	027501	022440	031517		
	011476	000				
8203	011477	015	041412	041475	QDISP: .ASCIZ	<CR><LF>/C=CONT.;R=REPEAT TEST;S=RESTART;D=DISPLAY MEMORY;H=REPEAT & HAL
	011504	047117	027124	051073		
	011512	051075	050105	040505		
	011520	020124	042524	052123		
	011526	051473	051075	051505		
	011534	040524	052122	042073		
	011542	042075	051511	046120		
	011550	054501	046440	046505		
	011556	051117	035531	036510		

	011564	042522	042520	052101	
	011572	023040	044040	046101	
	011600	020124	052101	041440	
	011606	051511	044440	051516	
8204	011614	037524	000		
	011617	045	022516	041501	STKM1: .ASCIZ /%N%ACIS INST EXECUTION USED MORE THAN 64 LOCS ON STACK%N/
	011624	051511	044440	051516	
	011632	020124	054105	041505	
	011640	052125	047511	020116	
	011646	051525	042105	046440	
	011654	051117	020105	044124	
	011662	047101	033040	020064	
	011670	047514	051503	047440	
	011676	020116	052123	041501	
8205	011704	022513	000116		
	011710	047045	040445	044503	STKM2: .ASCIZ /%N%ACIS INST EXECUTION DESTROYED CONTENTS OF WORD AT STACK+2%N/
	011716	020123	047111	052123	
	011724	042440	042530	052503	
	011732	044524	047117	042040	
	011740	051505	051124	054517	
	011746	042105	041440	047117	
	011754	04252	052116	020123	
	011762	043117	053440	051117	
	011770	020104	052101	051440	
	011776	040524	045503	031053	
8206	012004	047045	000		
	012007	045	022516	052101	CISQ: .ASCIZ /%N%ATRAP TO 10 OCCURRED ON CIS INST IN TEST #1. IS CISP PRESENT?%N/
	012014	040522	020120	047524	
	012022	030440	020060	041517	
	012030	052503	051122	042105	
	012036	047440	020116	044503	
	012044	020123	047111	052123	
	012052	044440	020116	042524	
	012060	052123	021440	027061	
	012066	044440	020123	044503	
	012074	050123	050040	042522	
	012102	042523	052116	022477	
8207	012110	000116			
	012112	047045	040445	051124	MMVMSG: .ASCIZ /%N%ATRAP TO 250/
	012120	050101	052040	020117	
8208	012126	032462	000060		
	012132	047045	040445	051124	TRAP4: .ASCIZ /%N%ATRAP TO 4/
	012140	050101	052040	020117	
8209	012146	000064			
	012150	047045	040445	051124	TRAP10: .ASCIZ /%N%ATRAP TO 10/
	012156	050101	052040	020117	
8210	012164	030061	000		
	012167	045	022516	042501	HLTMSG: .ASCIZ /%N%AERROR HALT/
	012174	051122	051117	044040	
8211	012202	046101	000124		
	012206	047045	040445	044503	NOPROG: .ASCIZ /%N%ACIS INST IS NOT MAKING PROGRESS%N/
	012214	020123	047111	052123	
	012222	044440	020123	047516	
	012230	020124	040515	044513	

	012236	043516	050040	047522	
	012244	051107	051505	022523	
	012252	000116			
8212	012254	047045	040445	040514	LATEXC: .ASCIZ /%N%ALATENCY EXCESSIVE%N/
	012262	042524	041516	020131	
	012270	054105	042503	051523	
	012276	053111	022505	000116	
8213	012304	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/
	012312	022472	033117	040445	
	012320	020040	047515	042504	
	012326	022472	030517	040445	
	012334	020040	026504	047105	
	012342	022472	030502	040445	
	012350	020040	047111	052123	
	012356	022472	033117	040445	
	012364	020040	047111	052123	
	012372	041440	035124	042045	
	012400	022462	030523	042045	
	012406	022465	000116		
8214	012412	047045	040445	047105	ASK: .ASCIZ /%N%ENTER INSTRUCTION TO TEST <ALL> /
	012420	042524	020122	047111	
	012426	052123	052522	052103	
	012434	047511	020116	047524	
	012442	052040	051505	020124	
	012450	040474	046114	020076	
	012456	020040	000		
8215	012461	045	022516	051101	ASKRM: .ASCIZ /%N%RANDOM EXERCISE MODE (Y OR N) ? /
	012466	047101	047504	020115	
	012474	054105	051105	044503	
	012502	042523	046440	042117	
	012510	020105	054450	047440	
	012516	020122	024516	037440	
	012524	020040	000		
8216	012527	045	022516	050101	ASKMOD: .ASCIZ /%N%APROCESSOR TEST MODE (K=KERNEL,S=SUPV,U=USER)? /
	012534	047522	042503	051523	
	012542	051117	052040	051505	
	012550	020124	047515	042504	
	012556	024040	036513	042513	
	012564	047122	046105	051454	
	012572	051475	050125	026126	
	012600	036525	051525	051105	
	012606	037451	020040	000	
8217	012613	045	022516	046501	ASKMM: .ASCIZ /%N%MEMORY MANAGEMENT (N=OFF,D=D-SPACE ENABLED,H=D-SPACE DISABLED)? /
	012620	046505	051117	020131	
	012626	040515	040516	042507	
	012634	042515	052116	024040	
	012642	036516	043117	026106	
	012650	036504	026504	050123	
	012656	041501	020105	047105	
	012664	041101	042514	026104	
	012672	036510	026504	050123	
	012700	041501	020105	044504	
	012706	040523	046102	042105	
	012714	037451	020040	000	



8218	012721	045	022516	052101	ASKINT: .ASCIZ /%N%ATEST INTERRUPTABILITY OF CIS INSTRUCTIONS (KW11-P REQUIRED) (Y OR N
	012726	051505	020124	047111	
	012734	042524	051122	050125	
	012742	040524	044502	044514	
	012750	054524	047440	020106	
	012756	044503	020123	047111	
	012764	052123	052522	052103	
	012772	047511	051516	024040	
	013000	053513	030461	050055	
	013006	051040	050505	044525	
	013014	042522	024504	024040	
	013022	020131	051117	047040	
	013030	020051	000077		
8219	013034	047045	040445	047105	ASKST: .ASCIZ /%N%AENTER STOP TEST NUMBER (DECIMAL):%N/
	013042	042524	020122	052123	
	013050	050117	052040	051505	
	013056	020124	052516	041115	
	013064	051105	024040	042504	
	013072	044503	040515	024514	
	013100	022472	000116		
8220	013104	047045	040445	047111	ASKSRC: .ASCIZ /%N%AINTR SOURCE (R=LTC,N=KW11-P @100KHZ,C=KW11-P @10KHZ,Y=KW11-P EXT OS
	013112	051124	051440	052517	
	013120	041522	020105	051050	
	013126	046075	041524	047054	
	013134	045475	030527	026461	
	013142	020120	030500	030060	
	013150	044113	026132	036503	
	013156	053513	030461	050055	
	013164	040040	030061	044113	
	013172	026132	036531	053513	
	013200	030461	050055	042440	
	013206	052130	047440	041523	
	013214	037451	000		
8221	013217	045	022516	040501	ASKDI: .ASCIZ /%N%AALLOW INTERRUPTING THE CIS INST EXECUTED DURING NORMAL INTR SERVICE
	013224	046114	053517	044440	
	013232	052116	051105	052522	
	013240	052120	047111	020107	
	013246	044124	020105	044503	
	013254	020123	047111	052123	
	013262	042440	042530	052503	
	013270	042524	020104	052504	
	013276	044522	043516	047040	
	013304	051117	040515	020114	
	013312	047111	051124	051440	
	013320	051105	044526	042503	
	013326	000077			
8222	013330	047045	040445	040503	NOLAT: .ASCIZ /%N%ACAN'T TEST LATENCY - NEED 2ND KW11-P/
	013336	023516	020124	042524	
	013344	052123	046040	052101	
	013352	047105	054503	026440	
	013360	047040	042505	020104	
	013366	047062	020104	053513	
	013374	030461	050055	000	
8223	013401	045	022516	041501	NOINT: .ASCIZ /%N%ACAN'T TEST INTERRUPTABILITY - NO CLOCK/

	013406	047101	052047	052040	
	013414	051505	020124	047111	
	013422	042524	051122	050125	
	013430	040524	044502	044514	
	013436	054524	026440	047040	
	013444	020117	046103	041517	
	013452	000113			
8224	013454	047045	040445	053523	SWNG: .ASCII /%N%ASWITCH ON CIS MODULE ACTS LIKE IT IS IN THE INCORRECT POSITION%/
	013462	052111	044103	047440	
	013470	020116	044503	020123	
	013476	047515	052504	042514	
	013504	040440	052103	020123	
	013512	044514	042513	044440	
	013520	020124	051511	044440	
	013526	020116	044124	020105	
	013534	047111	047503	051122	
	013542	041505	020124	047520	
	013550	044523	044524	047117	
	013556	047045	000		
8225	013561	045	022516	041501	NOABO: .ASCII /%N%ACIS INST FAILED TO ABORT%/
	013566	051511	044440	051516	
	013574	020124	040506	046111	
	013602	042105	052040	020117	
	013610	041101	051117	022524	
	013616	000116			
8226	013620	047045	040445	044514	KW11L: .ASCII /%N%ALINE CLOCK WILL BE USED FOR INTERRUPT SOURCE%/
	013626	042516	041440	047514	
	013634	045503	053440	046111	
	013642	020114	042502	052440	
	013650	042523	020104	047506	
	013656	020122	047111	042524	
	013664	051122	050125	020124	
	013672	047523	051125	042503	
	013700	047045	000		
8227	013703	015	040412	042104	AST: .ASCII <CR><LF>/ADDR(S)?/
	013710	024122	024523	000077	
8228	013716	047445	022466	027501	FORM19: .ASCII +%06%A/%S3%03%S2%03%S2%03%S2%03+
	013724	051445	022463	031517	
	013732	051445	022462	031517	
	013740	051445	022462	031517	
	013746	051445	022462	031517	
	013754	000			
8229	013755	045	020101	020040	ADDHDR: .ASCII /%A 0 1 2 3 4 5 6 7%/
	013762	020040	020040	020040	
	013770	020040	020060	020040	
	013776	030440	020040	020040	
	014004	020062	020040	031440	
	014012	020040	020040	020064	
	014020	020040	032440	020040	
	014026	020040	020066	020040	
	014034	033440	047045	000	
8230	014041	045	031123	047445	FORM20: .ASCII /%S2%03%52%03%52%03%52%03%4.
	014046	022463	031123	047445	
	014054	022463	031123	047445	

	014062	022463	031123	047445	
	014070	022463	000115		
8231	014074	047045	000		FORM21: .ASCIZ /%N/
8232	014077	045	022516	020101	FORM22: .ASCIZ /%N%A SRC%S13/
	014104	051123	022503	030523	
	014112	000063			
8233	014114	047045	040445	051440	FORM23: .ASCIZ /%N%A SRC1%S12/
	014122	041522	022461	030523	
	014130	000062			
8234	014132	047045	040445	051440	FORM24: .ASCIZ /%N%A SRC2%S12/
	014140	041522	022462	030523	
	014146	000062			
8235	014150	047045	040445	042440	FORM25: .ASCIZ /%N%A EXP RESULT%S4/
	014156	050130	051040	051505	
	014164	046125	022524	032123	
	014172	000			
8236	014173	045	022516	020101	FORM26: .ASCIZ /%N%A ACT RESULT%S4/
	014200	041501	020124	042522	
	014206	052523	052114	051445	
	014214	000064			
8237	014216	040445	000040		FORM27: .ASCIZ /%A /
8238	014222	051445	022462	044501	FORM30: .ASCIZ /%S2%A INST CNT:%D2%S1%D5/
	014230	051516	020124	047103	
	014236	035124	042045	022462	
	014244	030523	042045	000065	
8239	014252	040445	000111		FORM31: .ASCIZ /%A I/
8240	014256	040445	020040	044523	FORM32: .ASCIZ /%A SIGN BYTE=(/
	014264	047107	041040	052131	
	014272	036505	000050		
8241	014276	040445	000051		FORM33: .ASCIZ /%A)/
8242	014302	040445	025440	000	FORM34: .ASCIZ /%A +/
8243	014307	045	020101	000055	FORM35: .ASCIZ /%A -/
8244	014314	047045	047045	040445	FORM36: .ASCIZ /%N%N%ARANDOM # GENERATOR SEED%S5%06%S2%06%S2%06%N/
	014322	040522	042116	046517	
	014330	021440	043440	047105	
	014336	051105	052101	051117	
	014344	051440	042505	022504	
	014352	032523	047445	022466	
	014360	031123	047445	022466	
	014366	031123	047445	022466	
	014374	000116			
8245	014376	047045	040445	051105	FORM37: .ASCIZ /%N%AERROR #%D6/
	014404	047522	020122	022443	
	014412	033104	000		
8246	014415	045	022516	042501	FORM38: .ASCIZ /%N%AERROR IN UNUSED REGISTER SET: USED SET:%01/
	014422	051122	051117	044440	
	014430	020116	047125	051525	
	014436	042105	051040	043505	
	014444	051511	042524	020122	
	014452	042523	035124	052440	
	014460	042523	020104	042523	
	014466	035124	047445	000061	
8247	014474	047045	040445	054105	FORM39: .ASCIZ /%N%AFXP:%06%A ACT R0-R5 %06%S1%06%S1%06%S1%06%S1%06%S1%06/
	014502	035120	047445	022466	

	014510	020101	040440	052103	
	014516	051040	026460	032522	
	014524	022440	033117	051445	
	014532	022461	033117	051445	
	014540	022461	033117	051445	
	014546	022461	033117	051445	
	014554	022461	033117	051445	
	014562	022461	033117	000	
8248	014567	045	020101	047111	FORM40: .ASCIZ /%A INTR CNT:%D4%A REG SET:%01%A MODE:/
	014574	051124	041440	052116	
	014602	022472	032104	040445	
	014610	051040	043505	051440	
	014616	052105	022472	030517	
	014624	040445	046440	042117	
	014632	035105	000		
8249	014635	045	020101	042040	FRM40A: .ASCIZ /%A D-EN:/
	014642	042455	035116	000	
8250	014647	045	031123	022463	FORM41: .ASCIZ /%S23%AR0%S5%AR1%S5%AR2%S5%AR3%S5%AR4%S5%AR5%S5%AR6%S5%ANZVC%/
	014654	051101	022460	032523	
	014662	040445	030522	051445	
	014670	022465	051101	022462	
	014676	032523	040445	031522	
	014704	051445	022465	051101	
	014712	022464	032523	040445	
	014720	032522	051445	022465	
	014726	051101	022466	032523	
	014734	040445	055116	041526	
	014742	047045	000		
8251	014745	045	022516	041501	FORM42: .ASCIZ /%N%ACIS INSTRUCTION WAS SUSPENDED TO SERVICE INTERRUPT/
	014752	051511	044440	051516	
	014760	051124	041525	044524	
	014766	047117	053440	051501	
	014774	051440	051525	042520	
	015002	042116	042105	052040	
	015010	020117	042523	053122	
	015016	041511	020105	047111	
	015024	042524	051122	050125	
	015032	000124			
8252	015034	047045	040445	051520	FORM43: .ASCIZ /%N%APSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT%/
	015042	020127	044502	020124	
	015050	020070	044123	052517	
	015056	042114	044040	053101	
	015064	020105	042502	047105	
	015072	051440	052105	041040	
	015100	052125	053440	051501	
	015106	047040	052117	047045	
	015114	000			
8253	015115	045	022516	041101	FORM44: .ASCIZ /%N%ABIT 8 OF PSW SET WITH PC < CIS INST PC/
	015122	052111	034040	047440	
	015130	020106	051520	020127	
	015136	042523	020124	044527	
	015144	044124	050040	020103	
	015152	020074	044503	020123	
	015160	047111	052123	050040	

8254	015166	000103			
	015170	047045	040445	052523	FORM45: .ASCIZ /%N%ASUSPECT THAT CIS INST BACKED UP PC TOO FAR/
	015176	050123	041505	020124	
	015204	044124	052101	041440	
	015212	051511	044440	051516	
	015220	020124	040502	045503	
	015226	042105	052440	020120	
	015234	041520	052040	047517	
	015242	043040	051101	000	
8255	015247	045	022516	004501	FORM46: .ASCIZ /%N%A WHEN EXITING TO SERVICE INTERRUPT%N/
	015254	044127	047105	042440	
	015262	044530	044524	043516	
	015270	052040	020117	042523	
	015276	053122	041511	020105	
	015304	047111	042524	051122	
	015312	050125	022524	000116	
8256	015320	047045	040445	044503	FORM47: .ASCIZ /%N%ACIS INST COMPLETED BUT PSW BIT 8 STILL SET%N/
	015326	020123	047111	052123	
	015334	041440	046517	046120	
	015342	052105	042105	041040	
	015350	052125	050040	053523	
	015356	041040	052111	034040	
	015364	051440	044524	046114	
	015372	051440	052105	047045	
	015400	000			
8257	015401	045	022516	044501	FORM48: .ASCIZ /%N%AIN-LINE CIS INSTRUCTION COMPLETED WITH PC/
	015406	026516	044514	042516	
	015414	041440	051511	044440	
	015422	051516	051124	041525	
	015430	044524	047117	041440	
	015436	046517	046120	052105	
	015444	042105	053440	052111	
	015452	020110	041520	000	
8258	015457	045	022516	020101	FORM49: .ASCIZ /%N%A POINTING AT IN-LINE OPERANDS RATHER THAN NEXT INST%N/
	015464	050011	044517	052116	
	015472	047111	020107	052101	
	015500	044440	026516	044514	
	015506	042516	047440	042520	
	015514	040522	042116	020123	
	015522	040522	044124	051105	
	015530	052040	040510	020116	
	015536	042516	052130	044440	
	015544	051516	022524	000116	
8259	015552	006477	051012	042505	QUES: .ASCIZ /?/<CR><LF>/REENTER: /
	015560	052116	051105	000072	
8260	015566	005015	000		XCR LF: .ASCIZ <CR><LF>
8261	015571	057	005015	000	SLCRLF: .ASCIZ <SL><CR><LF>
8263	015575	015	042412	042116	ENDP: .ASCIZ <CR><LF>/END OF PASS (EXECUTION OF TABLED TEST CASES COMPLETE) /
	015602	047440	020106	040520	
	015610	051523	024040	054105	
	015616	041505	052125	047511	
	015624	020116	043117	052040	
	015632	041101	042514	020104	
	015640	042524	052123	041440	

8268	015646	051501	051505	041440	
	015654	046517	046120	052105	
	015662	024505	000		
	015665	015	042412	042116	ENDQP: .ASCIZ <CR><LF>/END OF QUICK VERIFY PASS/
	015672	047440	020106	052521	
	015700	041511	020113	042526	
	015706	044522	054506	050040	
8269	015714	051501	000123		
	015720	005015	047111	052123	FSHDR: .ASCIZ <CR><LF>/INST UNDER TEST WILL BE DISPLAYED AT THE START OF TESTING FOR E
	015726	052440	042116	051105	
	015734	052040	051505	020124	
	015742	044527	046114	041040	
	015750	020105	044504	050123	
	015756	040514	042531	020104	
	015764	052101	052040	042510	
	015772	051440	040524	052122	
	016000	047440	020106	042524	
	016006	052123	047111	020107	
	016014	047506	020122	040505	
	016022	044103	044440	051516	
	016030	000124			
8270	016032	005015	052521	041511	QVHDR: .ASCIZ <CR><LF>/QUICK VERIFY PASS TIME: LESS THAN 5 MINUTES/
	016040	020113	042526	044522	
	016046	054506	050040	051501	
	016054	020123	044524	042515	
	016062	020072	046040	051505	
	016070	020123	044124	047101	
	016076	032440	046440	047111	
	016104	052125	051505	000	
8271	016111	045	022516	042501	ACCSEED: .ASCIZ /%N%ENTER 3 RANDOM NUMBER GEN. SEED CONSTANTS: %N/
	016116	052116	051105	031440	
	016124	051040	047101	047504	
	016132	020115	052516	041115	
	016140	051105	043440	047105	
	016146	020056	042523	042105	
	016154	041440	047117	052123	
	016162	047101	051524	020072	
	016170	022440	000116		
8273	016174	005015	040520	051523	FSHDR1: .ASCIZ <CR><LF>+PASS TIME: APPROX 30 MIN+
	016202	052040	046511	035105	
	016210	040440	050120	047522	
	016216	020130	030063	046440	
	016224	047111	000		
8278	016227	015	042412	052116	FSHDR2: .ASCIZ <CR><LF>/ENTERING RANDOM TEST MODE/
	016234	051105	047111	020107	
	016242	040522	042116	046517	
	016250	052040	051505	020124	
	016256	047515	042504	000	
8279	016263	015	047012	020117	FSHDR3: .ASCIZ <CR><LF>/NO FURTHER END OF PASS MESSAGES WILL BE ISSUED/
	016270	052506	052122	042510	
	016276	020122	047105	020104	
	016304	043117	050040	051501	
	016312	020123	042515	051523	
	016320	043501	051505	053440	

	016326	046111	020114	042502	
	016334	044440	051523	042525	
	016342	000104			
8280	016344	005015	040522	042116	FSHDR4: .ASCIZ <CR><LF>/RANDOM # GENERATOR SEED CONSTANTS WILL BE PRINTED/
	016352	046517	021440	043440	
	016360	047105	051105	052101	
	016366	051117	051440	042505	
	016374	020104	047503	051516	
	016402	040524	052116	020123	
	016410	044527	046114	041040	
	016416	020105	051120	047111	
	016424	042524	000104		
8281	016430	005015	020040	020040	FSHDR5: .ASCIZ <CR><LF>/ EVERY 1024 CIS INSTRUCTION TESTS/
	016436	053105	051105	020131	
	016444	030061	032062	041440	
	016452	051511	044440	051516	
	016460	051124	041525	044524	
	016466	047117	052040	051505	
	016474	051524	000		
8285	016477	045	026501	026455	DASH: .ASCIZ /%A-----%N/
	016504	026455	026455	026455	
	016512	026455	026455	026455	
	016520	026455	026455	026455	
	016526	026455	026455	026455	
	016534	026455	026455	026455	
	016542	026455	026455	026455	
	016550	026455	026455	026455	
	016556	026455	026455	026455	
	016564	026455	026455	026455	
	016572	026455	047045	000	
8286	016577	045	022516	026501	SDASH: .ASCIZ /%N%A-----/
	016604	026455	026455	026455	
	016612	026455	026455	026455	
	016620	000			
8288	016621	015	041412	045532	PNAME: .ASCIZ <CR><LF>/CZKEEBO PDP-11 CIS INSTRUCTION EXERCISER /
	016626	042505	030102	050040	
	016634	050104	030455	020061	
	016642	044503	020123	047111	
	016650	052123	052522	052103	
	016656	047511	020116	054105	
	016664	051105	044503	042523	
	016672	020122	000		
8293	016675	015	046412	046505	MPT34: .ASCIZ <CR><LF>+MEM MGMT HARDWARE AVAILABLE ON SYSTEM UNDER TEST+
	016702	046440	046507	020124	
	016710	040510	042122	040527	
	016716	042522	040440	040526	
	016724	046111	041101	042514	
	016732	047440	020116	054523	
	016740	052123	046505	052440	
	016746	042116	051105	052040	
	016754	051505	000124		
8294	016760	005015	030461	032057	MPT44: .ASCIZ <CR><LF>+11/44 PROCESSOR+
	016766	020064	051120	041517	
	016774	051505	047523	000122	



8295	017002	005015	030461	033457	MPT74: .ASCIZ <CR><LF>+11/74 TYPE PROCESSOR+
	017010	020064	054524	042520	
	017016	050040	047522	042503	
	017024	051523	051117	000	
8296	017031	015	030412	027461	MPT2324: .ASCIZ <CR><LF>+11/23 OR 11/24 PROCESSOR+
	017036	031462	047440	020122	
	017044	030461	031057	020064	
	017052	051120	041517	051505	
	017060	047523	000122		
8297	017064	005015	047516	046440	NMM: .ASCIZ <CR><LF>+NO MEMORY MGMT HARDWARE ON SYSTEM UNDER TEST+
	017072	046505	051117	020131	
	017100	043515	052115	044040	
	017106	051101	053504	051101	
	017114	020105	047117	051440	
	017122	051531	042524	020115	
	017130	047125	042504	020122	
	017136	042524	052123	000	
8298					
8299					

```

8301      :SBTTL GLOBAL SUBROUTINES SECTION
8302
8303      .SBTTL      CIS EMULATOR
8304      : FUNCTIONAL DESCRIPTION:
8305      :   CIS EMULATOR      CONTAINS ROUTINES TO EMULATE EACH OF THE
8306      :                               CIS INSTRUCTIONS USING STANDARD PDP-11 INSTRUCTIONS
8307
8308      : INPUTS:      CIS INSTRUCTION TO EMULATE
8309      :                               CIS INSTRUCTION OPERANDS (LENGTHS,ADDRESSES,ETC)
8310      :                               STARTING ADDRESS FOR REGISTER RESULTS
8311      :                               ADDRESS FOR CONDITION CODE RESULTS
8312
8313      : IMPLICIT INPUTS:      CHARACTER OR DECIMAL STRINGS SETUP IN EMULATION BUFFER
8314
8315      : OUTPUTS:      CONDITION CODES
8316      :                               GENERAL PURPOSE REGISTERS
8317      :                               STRINGS IN EMULATION BUFFER
8318
8319      : IMPLICIT OUTPUTS:
8320
8321      : SUBORDINATE ROUTINES USED:
8322
8323      : FUNCTIONAL SIDE EFFECTS:
8324
8325      : CALLING SEQUENCE:      JSR PC,EMULATE
8326      :                               XXXXXX      ;OCTAL ENCODING OF CIS INST
8327      :                               YYYYYY      ;POINTER TO REGISTER OPERANDS
8328      :                               ZZZZZZ      ;POINTER TO REGISTER RESULTS
8329      :                               WWWWWW      ;POINTER TO CONDITION CODE RESULTS
8330
8331      177776      EPSW = 177776
8332      017144      .EVEN
  
```

```

8334
8335 017144 010037 017336
8336 017150 010137 017340
8337 017154 010237 017342
8338 017160 010337 017344
8339 017164 010437 017346
8340 017170 010537 017350
8341 017174 005037 036210
8342 017200 011600
8343 017202 012037 017354
8344 017206 012037 017356
8345 017212 012037 017360
8346 017216 012037 017362
8347 017222 010016
8348 017224 012700 017404
8349 017230 005710
8350 017232 001001
8351 017234 000207
8352 017236 023720 017354
8353 017242 001372
8354 017244 162700 017406
8355 017250 062700 017530
8356 017254 011037 017352
8357 017260 012700 017364
8358 017264 013701 017356
8359 017270 012120
8360 017272 022700 017402
8361 017276 001374
8362 017300 004777 000046
8363 017304 013700 017336
8364 017310 013701 017340
8365 017314 013702 017342
8366 017320 013703 017344
8367 017324 013704 017346
8368 017330 013705 017350
8369 017334 000207
    
```

EMULAT:

1\$:

2\$:

3\$:

```

.SBTTL
MOV R0,ESTORE
MOV R1,ESTORE+2
MOV R2,ESTORE+4
MOV R3,ESTORE+6
MOV R4,ESTORE+10
MOV R5,ESTORE+12
CLR EZDF
MOV (SP),R0
MOV (R0)+,EINST
MOV (R0)+,EIRSTK
MOV (R0)+,EORSTK
MOV (R0)+,EOPSW
MOV R0,(SP)
MOV #ELISTA,R0
TST (R0)
BNE 2$
RTS PC
CMP EINST,(R0)+
BNE 1$
SUB #ELISTA+2,R0
ADD #ELISTB,R0
MOV (R0),EROUT
MOV #EO,R0
MOV EIRSTK,R1
MOV (R1)+,(R0)+
CMP #E6+2,R0
BNE 3$
JSR PC,@EROUT
MOV ESTORE,R0
MOV ESTORE+2,R1
MOV ESTORE+4,R2
MOV ESTORE+6,R3
MOV ESTORE+10,R4
MOV ESTORE+12,R5
RTS PC
    
```

INSTRUCTION DECODER

;SAVE OLD REGISTER VALUES

;CLEAR ZERO DIVIDE FLAG

;GRAB DATA POINTERS

;CIS INSTRUCTION BEING CALLED

;FAKE INPUT GPRS

;FAKE OUTPUT GPRS

;FAKE PSW

;SUB RETURN ADDRESS

;CIS COMMAND LIST POINTER

;EXIT IF INSTRUCTION CANT DECODE

;ELSE CONTINUE SEARCH

;LOOK AT TABLE FOR MATCH

;KEEP TRYING

;AT LAST, SO FIND HANDLER FOR IT

;HANDLER ADDRESS IN TABLE B

;HOLD ADDRESS WHILE I FIND THE DATA

;POINTER TO EMULATE GPRS

;POINTER TO REGISTER DATA

;BEAM OVER THE DATA

;EXIT LOOP AFTER R6 LOADED

;ELSE LOAD NEXT

;EXECUTE EMULATED CIS INSTRUCTION

;ON RETURN RESTORE REGISTERS

;RETURN TO MAIN PROGRAM

8371  
8372 017336 000006  
8373 017352 000000  
8374 017354 000000  
8375 017356 000000  
8376 017360 000000  
8377 017362 000000  
8378 017364 000000  
8379 017366 000000  
8380 017370 000000  
8381 017372 000000  
8382 017374 000000  
8383 017376 000000  
8384 017400 000000  
8385 017402 000000  
8386 017404 076020  
8387 017406 076021  
8388 017410 076022  
8389 017412 076023  
8390 017414 076024  
8391 017416 076025  
8392 017420 076026  
8393 017422 076027  
8394 017424 076030  
8395 017426 076031  
8396 017430 076032  
8397 017432 076040  
8398 017434 076041  
8399 017436 076042  
8400 017440 076043  
8401 017442 076044  
8402 017444 076045  
8403 017446 076050  
8404 017450 076051  
8405 017452 076052  
8406 017454 076053  
8407 017456 076054  
8408 017460 076055  
8409 017462 076056  
8410 017464 076057  
8411 017466 076060  
8412 017470 076061  
8413 017472 076062  
8414 017474 076063  
8415 017476 076064  
8416 017500 076065  
8417 017502 076066  
8418 017504 076067  
8419 017506 076070  
8420 017510 076071  
8421 017512 076072  
8422 017514 076073  
8423 017516 076074  
8424 017520 076075

.SBTTL  
ESTORE: .BLKW 6  
EROUT: .WORD 0  
EINST: .WORD 0  
EIRSTK: .WORD 0  
EORSTK: .WORD 0  
EOPSW: .WORD 0  
E0: .WORD 0  
E1: .WORD 0  
E2: .WORD 0  
E3: .WORD 0  
E4: .WORD 0  
E5: .WORD 0  
E6: .WORD 0  
TEMP: .WORD 0  
ELISTA: .WORD 076020  
.WORD 076021  
.WORD 076022  
.WORD 076023  
.WORD 076024  
.WORD 076025  
.WORD 076026  
.WORD 076027  
.WORD 76030  
.WORD 76031  
.WORD 76032  
.WORD 76040  
.WORD 76041  
.WORD 76042  
.WORD 76043  
.WORD 76044  
.WORD 76045  
.WORD 76050  
.WORD 76051  
.WORD 76052  
.WORD 76053  
.WORD 76054  
.WORD 76055  
.WORD 76056  
.WORD 76057  
.WORD 076060  
.WORD 076061  
.WORD 076062  
.WORD 076063  
.WORD 076064  
.WORD 076065  
.WORD 076066  
.WORD 076067  
.WORD 76070  
.WORD 76071  
.WORD 76072  
.WORD 76073  
.WORD 76074  
.WORD 76075

DATA STORAGE

8425 017522 076076  
8426 017524 076077  
8427 017526 000000  
8428 017530 036216  
8429 017532 036322  
8430 017534 036330  
8431 017536 036336  
8432 017540 036344  
8433 017542 036360  
8434 017544 036374  
8435 017546 036440  
8436 017550 017652  
8437 017552 020356  
8438 017554 020074  
8439 017556 021246  
8440 017560 021300  
8441 017562 021170  
8442 017564 021112  
8443 017566 020600  
8444 017570 021002  
8445 017572 023412  
8446 017574 023460  
8447 017576 032314  
8448 017600 026766  
8449 017602 027532  
8450 017604 030144  
8451 017606 030716  
8452 017610 025170  
8453 017612 036500  
8454 017614 036554  
8455 017616 036562  
8456 017620 036570  
8457 017622 036576  
8458 017624 036604  
8459 017626 036612  
8460 017630 036672  
8461 017632 023402  
8462 017634 023420  
8463 017636 032240  
8464 017640 026774  
8465 017642 033100  
8466 017644 035142  
8467 017646 030642  
8468 017650 025176

ELISTB:

.WORD 76076  
.WORD 76077  
.WORD 0  
.WORD EL2D0  
.WORD EL2D1  
.WORD EL2D2  
.WORD EL2D3  
.WORD EL2D4  
.WORD EL2D5  
.WORD EL2D6  
.WORD EL2D7  
.WORD EMOVC  
.WORD EMOVRC  
.WORD EMOVTC  
.WORD ELOCC  
.WORD ESKPC  
.WORD ESCNC  
.WORD ESPNC  
.WORD ECMPC  
.WORD EMTCHC  
.WORD EADDN  
.WORD ESUBN  
.WORD ECMPN  
.WORD ECVTNL  
.WORD ECVTPN  
.WORD ECVTNP  
.WORD EASHN  
.WORD ECVTLN  
.WORD EL3D0  
.WORD EL3D1  
.WORD EL3D2  
.WORD EL3D3  
.WORD EL3D4  
.WORD EL3D5  
.WORD EL3D5  
.WORD EL3D6  
.WORD EL3D7  
.WORD EADDP  
.WORD ESUBP  
.WORD ECMPP  
.WORD ECVTPL  
.WORD EMULP  
.WORD EDIVP  
.WORD EASHP  
.WORD ECVTLP

8470					.SBTTL	MOVE STRING
8471	017652	013700	017366		MOV E1,R0	;FIND END OF SOURCE STRING
8472	017656	063700	017364		ADD E0,R0	
8473	017662	013701	017372		MOV E3,R1	;FIND END OF DEST. STRING
8474	017666	063701	017370		ADD E2,R1	
8475	017672	023737	017364	017370	CMP E0,E2	;WICH STRING IS LONGER
8476	017700	101003			BHI 1\$	;SOURCE
8477	017702	105416			BLO 2\$	;DEST.
8478	017704	010102			MOV R1,R2	;THEIR THE SAME
8479	017706	000420			BR 3\$	
8480	017710	013700	017366		1\$: MOV E1,R0	;SHORTEN SOURCE STRING
8481	017714	063700	017370		ADD E2,R0	
8482	017720	010102			MOV R1,R2	;DEST USED END = REAL END
8483	017722	013703	017364		MOV E0,R3	;CALCULATE # OF CHARS
8484	017726	163703	017370		SUB E2,R3	; NOT TRANSFERRED.
8485	017732	010377	177422		MOV R3,@EORSTK	;SAVE RESULT
8486	017736	000406			BR 4\$	
8487	017740	013702	017372		2\$: MOV E3,R2	;USED END < REAL END
8488	017744	063702	017364		ADD E0,R2	
8489	017750	005077	177404		3\$: CLR @EORSTK	;ALL CHAR. TRANSFERED TO DEST.
8490	017754	023737	017366	017372	4\$: CMP E1,E3	;WICH STRING IS IN HIGH CORE
8491	017762	103410			BLO EFORWD	;DEST STRING IS
8492	017764	013703	017366		EBACK: MOV E1,R3	;START ADDRESS OF SOURCE
8493	017770	013704	017372		MOV E3,R4	;START ADDRESS OF DEST.
8494	017774	020402			1\$: CMP R4,R2	;IS TRANSFER COMPLETE ?
8495	017776	001410			BEQ EFILL	;YES
8496	020000	112324			MOVB (R3)+,(R4)+	;XFER CHAR.
8497	020002	000774			BR 1\$	
8498	020004	010203			EFORWD: MOV R2,R3	;DEST STRING POINTER
8499	020006	023700	017366		1\$: CMP E1,R0	;IS XFER COMPLETE ?
8500	020012	001402			BEQ EFILL	;YES
8501	020014	114043			MOVB -(R0),-(R3)	;XFER CHAR.
8502	020016	000773			BR 1\$	
8503	020020	020102			EFILL: CMP R1,R2	;ADD FILL CHARS. TO COMPLETE STRING
8504	020022	001403			BEQ 2\$	
8505	020024	113722	017374		MOVB E4,(R2)+	;XFER FILL
8506	020030	000773			BR EFILL	
8507	020032	013700	017360		2\$: MOV EORSTK,R0	;RETURN CLEAN UP
8508	020036	005720			TST (R0)+	;R0 = R0
8509	020040	005020			CLR (R0)+	;R1 = 0
8510	020042	005020			CLR (R0)+	;R2 = 0
8511	020044	005020			CLR (R0)+	;R3 = 0
8512	020046	013720	017374		MOV E4,(R0)+	;R4 = R4
8513	020052	013710	017376		MOV E5,(R0)	;R5 = R5
8514	020056	023737	017364	017370	CMP E0,E2	;SET PSW CC BITS
8515	020064	013777	177776	177270	MOV EPSW,@EOPSW	;STORE RESULT
8516	020072	000207			RTS PC	

8518					.SBTTL	MOVE STRING TRANSLATE
8519	020074	013700	017366		EMOVTC: MOV E1,R0	;FIND END OF SOURCE
8520	020100	063700	017364		ADD E0,R0	
8521	020104	013701	017372		MOV E3,R1	;FIND END OF DEST.
8522	020110	063701	017370		ADD E2,R1	
8523	020114	023737	017364	017370	CMP E0,E2	;WHICH STRING IS LONGER
8524	020122	101003			BHI 1\$	;SOURCE
8525	020124	103416			BLO 2\$	;DEST.
8526	020126	010102			MOV R1,R2	;SAME
8527	020130	000420			BR 3\$	
8528	020132	013700	017366		1\$: MOV E1,R0	;SHORTEN SOURCE STRING
8529	020136	063700	017370		ADD E2,R0	
8530	020142	010102			MOV R1,R2	
8531	020144	013703	017364		MOV E0,R3	;CALCULATE NO. OF CHARS.
8532	020150	163703	017370		SUB E2,R3	; NOT TRANSFERRED
8533	020154	010377	177200		MOV R3,@EORSTK	;STORE RESULT
8534	020160	000406			BR 4\$	
8535	020162	013702	017372		2\$: MOV E3,R2	;MARK REAL END OF DEST
8536	020166	063702	017364		ADD E0,R2	
8537	020172	005077	177162		3\$: CLR @EORSTK	
8538	020176	023737	017366	017372	4\$: CMP E1,E3	;WHO'S HIGHER IN MEMORY
8539	020204	103420			BLO EMTFRD	;MOVE FORWARD
8540	020206	013703	017366		EMTBCK: MOV E1,R3	;START OF SOURCE
8541	020212	013704	017372		MOV E3,R4	;START OF DEST.
8542	020216	020402			1\$: CMP R4,R2	;XFER COMPLETE YET
8543	020220	001430			BEQ EMTFIL	;YES
8544	020222	112337	017402		MOVB (R3)+,TEMP	;CAL. INDEX INTO TABLE
8545	020226	105037	017403		CLRB TEMP+1	
8546	020232	063737	017376	017402	ADD E5,TEMP	
8547	020240	117724	177136		MOVB @TEMP,(R4)+	;MOVE TABLE VALUE TO DEST.
8548	020244	000764			BR 1\$	
8549	020246	010203			EMTFRD: MOV R2,R3	;DEST MOVE POINTER
8550	020250	023700	017366		1\$: CMP E1,R0	;XFER COMPLETE ?
8551	020254	001412			BEQ EMTFIL	
8552	020256	114037	017402		MOVB -(R0),TEMP	;CAL INDEX INTO TABLE
8553	020262	105037	017403		CLRB TEMP+1	
8554	020266	063737	017376	017402	ADD E5,TEMP	
8555	020274	117743	177102		MOVB @TEMP,-(R3)	;MOVE TABLE VALUE INTO DEST.
8556	020300	000763			BR 1\$	
8557	020302	020102			EMTFIL: CMP R1,R2	;COMPLETE
8558	020304	001403			BEQ 2\$	;YES
8559	020306	113722	017374		MOVB E4,(R2)+	;XFER FILL
8560	020312	000773			BR EMTFIL	
8561	020314	013700	017360		2\$: MOV EORSTK,R0	;RETURN CLEAN UP
8562	020320	005720			TST (R0)+	;R0 = R0
8563	020322	005020			CLR (R0)+	;R1 = 0
8564	020324	005020			CLR (R0)+	;R2 = 0
8565	020326	005020			CLR (R0)+	;R3 = 0
8566	020330	013720	017374		MOV E4,(R0)+	;R4 = R4
8567	020334	013710	017376		MOV E5,(R0)	;R5 = R5
8568	020340	023737	017364	017370	CMP E0,E2	;SET CC BITS
8569	020346	013777	177776	177006	MOV EPSW,@EOPSW	;RETURN STATUS
8570	020354	000207			RTS PC	



8573					.SBTTL	MOVE REVERSE STRING
8574	020356	013700	017366		EMOVRC: MOV E1,R0	;FIND END OF SOURCE
8575	020362	063700	017364		ADD E0,R0	
8576	020366	013701	017372		MOV E3,R1	;FIND END OF DEST.
8577	020372	063701	017370		ADD E2,R1	
8578	020376	023737	017364	017370	CMP E0,E2	;WHICH STRING IS LARGER
8579	020404	101004			BHI 1\$	;SOURCE
8580	020406	103421			BLO 2\$	;DEST.
8581	020410	013702	017372		MOV E3,R2	;SAME
8582	020414	000421			E 3\$	
8583	020416	010037	017366		1\$: MOV R0,E1	;SHORTEN SOURCE
8584	020422	163737	017370	017366	SUB E2,E1	
8585	020430	013702	017372		MOV E3,R2	;DEST. REAL START
8586	020434	013703	017364		MOV E0,R3	;CALCULATE NO OF CHARS
8587	020440	163703	017370		SUB E2,R3	; NOT TRANSFERRED
8588	020444	010377	176710		MOV R3,@FORSTK	;STORE RESULT
8589	020450	000405			BR 4\$	
8590	020452	010102			2\$: MOV R1,R2	;MARK REAL START OF DEST.
8591	020454	163702	017364		SUB E0,R2	
8592	020460	005077	176674		3\$: CLR @EORSTK	;ALL CHARS. TRANSFERED
8593	020464	023702	017366		4\$: CMP E1,R2	;WHO'S IN HIGH MEMORY
8594	020470	103407			BLO EMRFD	;MOVE FORWARD
8595	020472	013703	017366		EMRBKD: MOV E1,R3	;SOURCE START POINTER
8596	020476	010204			MOV R2,R4	;DEST. START POINTER
8597	020500	020401			1\$: CMP R4,R1	;XFER COMPLETE
8598	020502	001406			BEQ EMRFIL	;YES
8599	020504	112324			MOVB (R3)+,(R4)+	;XFER CHAR.
8600	020506	000774			BR 1\$	
8601	020510	020102			EMRFD: CMP R1,R2	;XFER COMPLETE
8602	020512	001402			BEQ EMRFIL	;YES
8603	020514	114041			MOVB -(R0),-(R1)	;XFER CHAR.
8604	020516	000774			BR EMRFD	
8605	020520	013703	017372		EMRFIL: MOV E3,R3	;ADD FILLER
8606	020524	020302			2\$: CMP R3,R2	;FILL COMPLETE
8607	020526	001403			BEQ 1\$	;YES
8608	020530	113723	017374		MOVB E4,(R3)+	;XFER FILL
8609	020534	000773			BR 2\$	
8610	020536	013700	017360		1\$: MOV EORSTK,R0	;RETURN CLEAN UP
8611	020542	005720			TST (R0)+	;R0 = R0
8612	020544	005020			CLR (R0)+	;R1 = 0
8613	020546	005020			CLR (R0)+	;R2 = 0
8614	020550	005020			CLR (R0)+	;R3 = 0
8615	020552	013720	017374		MOV E4,(R0)+	;R4 = R4
8616	020556	013710	017376		MOV E5,(R0)	;R5 = R5
8617	020562	023737	017364	017370	CMP E0,E2	;SET CC BITS
8618	020570	013777	177776	176564	MOV EPSW,@EOPSW	;RETURN TO USER
8619	020576	000207			RTS PC	

8621					.SBTTL	COMPARE STRING
8622	020600	013700	017364	ECMPC:	MOV E0,R0	:CAL. END OF SCR1
8623	020604	063700	017366		ADD E1,R0	
8624	020610	013701	017370		MOV E2,R1	:CAL. END OF SCR2
8625	020614	063701	017372		ADD E3,R1	
8626	020620	013702	017366		MOV E1,R2	:START OF SCR1
8627	020624	013703	017372		MOV E3,R3	:START OF SCR2
8628	020630	020002		1\$:	CMP R0,R2	:END OF SCR1
8629	020632	001427			BEQ ENDA	:YES
8630	020634	020103			CMP R1,R3	:END OF SCR2
8631	020636	001445			BEQ ENDB	:YES
8632	020640	121213			CMPB (R2),(R3)	:SET CC BITS
8633	020642	013777	177776 176512		MOV EPSW,@EOPSW	:STORE STATUS
8634	020650	122322			CMPB (R3)+,(R2)+	:FIND NON MATCHING CHARS.
8635	020652	001766			BEQ 1\$	:KEEP TRYING
8636	020654	005303			DEC R3	:ADJ SCR1 POINTER
8637	020656	005302			DEC R2	:ADJ SCR2 POINTER
8638	020660	160301		ECMOUT:	SUB R3,R1	:NO. OF CHARS. LEFT IN SCR2
8639	020662	160200			SUB R2,R0	:NO. OF CHARS. LEFT IN SCR1
8640	020664	013704	017360	EMATOT:	MOV EORSTK,R4	:REGISTER DUMP POIN'ER
8641	020670	010024			MOV R0,(R4)+	:R0 = SCR1 LEN
8642	020672	010224			MOV R2,(R4)+	:R1 = SCR1 START
8643	020674	010124			MOV R1,(R4)+	:R2 = SCR2 LEN
8644	020676	010324			MOV R3,(R4)+	:R3 = SCR2 START
8645	020700	013724	017374		MOV E4,(R4)+	:R4 = R4
8646	020704	013724	017376		MOV E5,(R4)+	:R5 = R5
8647	020710	000207			RTS PC	
8648	020712	020103		END A:	CMP R1,R3	:END SCR2 ?
8649	020714	001004			BNE 1\$	
8650	020716	013777	177776 176436		MOV EPSW,@EOPSW	:YES - STORE STATUS
8651	020724	000755			BR ECMOUT	
8652	020726	123713	017374	1\$:	CMPB E4,(R5)	:SET CC BITS
8653	020732	013777	177776 176422		MOV EPSW,@EOPSW	:STORE RESULT
8654	020740	123723	017374		CMPB E4,(R3)+	:FIND NON MATCHING CHARS.
8655	020744	001762			BEQ ENDA	:KEEP TRYING
8656	020746	005303			DEC R3	:ADJ SCR2 POINTER
8657	020750	000743			BR ECMOUT	
8658	020752	020002		ENDB:	CMP R0,R2	:END SCR1
8659	020754	001741			BEQ ECMOUT	:YES
8660	020756	121237	017374		CMPB (R2),E4	:SET CC BITS
8661	020762	013777	177776 176372		MOV EPSW,@EOPSW	:SAVE RESULT
8662	020770	123722	017374		CMPB E4,(R2)+	:FIND NON MATCHING CHARS.
8663	020774	001766			BEQ ENDB	:KEEP TRYING
8664	020776	005302			DEC R2	:ADJ POINTER
8665	021000	000727			BR ECMOUT	

8667					.SBTTL		MATCH STRING
8668	021002	013700	017364	EMTCHC:	MOV E0,R0		:CALCULATE END OF SRC
8669	021006	063700	017366		ADD E1,R0		
8670	021012	013701	017370		MOV E2,R1		:CALCULATE END OF OBJ
8671	021016	063701	017372		ADD E3,R1		
8672	021022	013702	017366		MOV E1,R2		:START OF SRC
8673	021026	013703	017372		MOV E3,R3		:START OF OBJ
8674	021032	010204		4\$:	MOV R2,R4		:SAVE START OF SRC
8675	021034	020103		1\$:	CMP R1,R3		:OBJ FOUND IN STRING?
8676	021036	001411			BEQ 3\$		:YES
8677	021040	020200			CMP R2,R0		:END OF STRING AND NO MATCH
8678	021042	001410			BEQ 2\$		:YES
8679	021044	122322			CMPB (R3)+,(R2)+		:DO CHARS. MATCH
8680	021046	001772			BEQ 1\$		:YES
8681	021050	013703	017372		MOV E3,R3		:NO , SO RESET OBJ POINTER
8682	021054	005204			INC R4		:ADVANCE SRC POINTER TO ONE CHAR
8683							: POSITION BEYOND WHERE IT WAS
8684	021056	010402			MOV R4,R2		
8685	021060	000764			BR 4\$		
8686	021062	010402		3\$:	MOV R4,R2		
8687	021064	013703	017372	2\$:	MOV E3,R3		:RESTORE OBJ.LEN AND OBJ.ADR
8688	021070	013701	017370		MOV E2,R1		
8689	021074	160200			SUB R2,R0		:SET CC BITS
8690	021076	005700			TST R0		
8691	021100	013777	177776 176254		MOV EPSW,@EOPSW		:SAVE RESULT
8692	021106	000137	020664		JMP EMAT01		

8694  
8695 021112 013700 017366  
8696 021116 063700 017364  
8697 021122 013701 017366  
8698 021126 020100  
8699 021130 001476  
8700 021132 112137 017402  
8701 021136 105037 017403  
8702 021142 063737 017376 017402  
8703 021150 117702 176226  
8704 021154 133702 017374  
8705 021160 001362  
8706 021162 005301  
8707 021164 000137 021326

ESPNC:  
1\$:  
017402

```
.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
```

8709  
8710 021170 013700 017366  
8711 021174 063700 017364  
8712 021200 013701 017366  
8713 021204 020100  
8714 021206 001447  
8715 021210 112137 017402  
8716 021214 105037 017403  
8717 021220 063737 017376 017402  
8718 021226 117702 176150  
8719 021232 133702 017374  
8720 021236 001762  
8721 021240 005301  
8722 021242 000137 021326

ESCNC:

1\$:

017402

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

SEARCH FOR 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

8724			
8725	021246	013700	017366
8726	021252	010001	
8727	021254	063700	017364
8728	021260	020001	
8729	021262	001421	
8730	021264	122137	017374
8731	021270	001373	
8732	021272	005301	
8733	021274	000137	021326

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

8735				
8736	021300	013700	017366	
8737	021304	010001		
8738	021306	063700	017364	
8739	021312	020001		
8740	021314	001404		
8741	021316	122137	017374	
8742	021322	001773		
8743	021324	005301		
8744	021326	160100		
8745	021330	013702	017360	
8746	021334	010022		
8747	021336	010122		
8748	021340	013722	017370	
8749	021344	013722	017372	
8750	021350	013722	017374	
8751	021354	013722	017376	
8752	021360	005700		
8753	021362	013777	177776	175772
8754	021370	000207		

```
ESKPC:      .SBTTL
             MOV E1,R0
             MOV R0,R1
             ADD E0,R0
1$:          CMP R0,R1
             BEQ ESPND
             CMPB (R1)+,E4
             BEQ 1$
             DEC R1
ESPND:      SUB R1,R0
             MOV EORSTK,R2
             MOV R0,(R2)+
             MOV R1,(R2)+
             MOV E2,(R2)+
             MOV E3,(R2)+
             MOV E4,(R2)+
             MOV E5,(R2)+
             TST R0
             MOV EPSW,@EOPSW
             RTS PC
```

```
SKPC INSTRUCTION
;CALCULATE END ADDRESS
;START ADDRESS

;SEARCH FOR PASS
;FOUND IT
;LOOK FOR NOT CHAR.

;NO. OF CHARS.
;REGISTER DUMP POINTER
;R0 = LEN
;R1 = POS.
;R2 = R2
;R3 = R3
;R4 = R4
;R5 = R5
;SET CC BITS
;STORE RESULTS
```



8756  
8757  
8758  
8759  
8760  
8761  
8762  
8763  
8764  
8765  
8766  
8767  
8768  
8769  
8770  
8771  
8772  
8773  
8774  
8775  
8776  
8777  
8778  
8779  
8780  
8781  
8782  
8783  
8784  
8785  
8786  
8787  
8788  
8789  
8790  
8791  
8792  
8793  
8794  
8795  
8796  
8797  
8798  
8799  
8800  
8801  
8802  
8803  
8804  
8805  
8806  
8807  
8808  
8809

021372 005737 026322  
021376 100403  
021400 004737 021616  
021404 000402  
021406 004737 021414  
021412 000207  
  
021414 010037 032130  
021420 010137 032132  
021424 042701 070000  
021430 005737 025112  
021434 100401  
021436 005201  
021440 010137 025074  
021444 006237 025074  
021450 042737 000001 025074  
021456 060037 025074  
021462 013700 025074  
021466 032701 000002  
021472 001403  
021474 116002 000001  
021500 000401  
021502 111002  
021504 110237 025166  
021510 032701 000001  
021514 001433  
021516 042702 177760  
021522 123737 025130 032132  
021530 001020  
021532 005037 025062

```

.SBTTL                                DIGIT (SIGN) RETRIEVER
*****
ROUTINE TO RETRLIVE 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
*****

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

```

8810	021536	020227	000013			CMP R2,#13	;IS RETRIEVED SIGN NEG (1011)?
8811	021542	001403				BEQ 7\$	;BRANCH IF YES
8812	021544	020227	000015			CMP R2,#15	;IS RETRIEVED SIGN NEG (1101)?
8813	021550	001007				BNE 6\$	;BRANCH IF NO
8814	021552	122737	000160	032133	7\$:	CMPB #160,ETMPR1+1	;IS INST UNSIGNED PACKED?
8815	021560	001403				BEQ 6\$	;BRANCH IF YES
8816	021562	012737	177777	025062		MOV #177777,ERSNEG	;SET NEG FLAG
8817	021570	005002			6\$:	CLR R2	
8818	021572	013700	032130		5\$:	MOV ETMPRO,R0	;RESTORE REGISTERS
8819	021576	013701	032132			MOV ETMPR1,R1	
8820	021602	000207				RTS PC	;RETURN
8821	021604	006202			3\$:	ASR R2	;SELECT UPPER NIBBLE
8822	021606	006202				ASR R2	
8823	021610	006202				ASR R2	
8824	021612	006202				ASR R2	
8825	021614	000740				BR 4\$	
8826							
8827							
8828	021616	010037	032130			:ZONED STRING DIGIT RETRIEVER	
8829	021622	010137	032132			EFNDTZ: MOV R0,ETMPRO	;SAVE REGISTER
8830	021626	013737	025130	025122		MOV R1,ETMPR1	
8831	021634	042701	070000			MOV ELSD,ESLSD	
8832	021640	123737	025130	032132		BIC #070000,R1	
8833	021646	001442				CMPB ELSD,ETMPR1	;REQUEST FOR SIGN?
8834	021650	122737	000040	032133	6\$:	BEQ 1\$	;BRANCH IF YES
8835	021656	001405				CMPB #040,ETMPR1+1	;IS DESC TYPE TRAILING OVERPUNCH?
8836	021660	122737	000060	032133		BEQ 4\$	;BRANCH IF YES
8837	021666	001411				CMPB #060,ETMPR1+1	;IS DESC TYPE LEADING OVERPUNCH?
8838	021670	000415				BEQ 5\$	;BRANCH IF YES
8839						BR 3\$	
8840	021672	005337	025130		4\$:	DEC ELSD	;TYPE = TRAILING OVERPUNCH
8841	021676	123701	025130			CMPB ELSD,R1	;IS DIGIT REQUESTED PART OF THE
8842							; ENCODED SIGN DIGIT?
8843	021702	001010				BNE 3\$	;BRANCH IF NO
8844	021704	004737	022230			JSR PC,DECZO	;DECODE OVERPUNCH BYTE FOR DIGIT
8845	021710	000407				BR 2\$	
8846							
8847	021712				5\$:		;TYPE = LEADING OVERPUNCH
8848	021712	005701				TST R1	;IS DIGIT REQUESTED PART OF ENCODED
8849							; SIGN DIGIT?
8850	021714	001003				BNE 3\$	;BRANCH IF NO
8851	021716	004737	022230			JSR PC,DECZO	;DECODE OVERPUNCH BYTE
8852	021722	000402				BR 2\$	
8853							
8854	021724	060100			3\$:	ADD R1,R0	;BYTE ADDRESS
8855	021726	111002				MOV (R0),R2	;DATA
8856	021730	042702	177760		2\$:	BIC #177760,R2	;MASK OFF JUNK
8857	021734	013700	032130			MOV ETMPRO,R0	;RESTORE REGISTER
8858	021740	013701	032132			MOV ETMPR1,R1	
8859	021744	013737	025122	025130		MOV ESLSD,ELSD	
8860	021752	000207				RTS PC	
8861	021754				1\$:		;SIGN REQUESTED
8862	021754	105737	025130			TSTB ELSD	;IS STRING LEN = 0?
8863	021760	001020				BNE 10\$	;BRANCH IF NO

8864	021762	122737	000100	032133		CMPB #100,ETMPR1+1	:TYPE = TRAILING SEPARATE?
8865	021770	001002				BNE 103\$	:BRANCH IF NO
8866	021772	000137	022154			JMP 14\$	:RETURN SIGN AT 'A'
8867	021776	122737	000120	032133	103\$:	CMPB #120,ETMPR1+1	:IS STRING TYPE LEADING SEPARATE?
8868							:NOTE: SEPARATE TYPE ARE THE ONLY
8869							: 0 LEN ZONED STRING THAT
8870							: THAT OCCUPIES MEMORY.
8871	022004	001002				BNE 102\$	:BRANCH IF NO
8872	022006	000137	022202			JMP 15\$	:RETURN SIGN AT 'A-1'
8873	022012	005037	025166		102\$:	CLR SGNBYT	:CLEAR SIGN BYTE FOR POSSIBLE ERROR PRINT
8874	022016	000137	022054			JMP 100\$	:RETURN + SIGN
8875							
8876	022022	122737	000000	032133	10\$:	CMPB #000,ETMPR1+1	:IS TYPE = SIGNED ZONED?
8877	022030	001022				BNE 11\$	:BRANCH IF NO
8878	022032	060100				ADD R1,R0	:FORM SIGN ADDRESS
8879	022034	114002				MOVB -(R0),P2	
8880	022036	010237	025166			MOV R2,SGNBYT	:SAVE SIGN BYTE
8881	022042	042702	177417			BIC #177417,R2	:LOOK ONLY AT SIGN
8882	022046	020227	000160			CMP R2,#160	:IS IT (0111) NEGATIVE
8883	022052	001404				BEQ 101\$	:BRANCH IF YES
8884	022054	005037	025062		100\$:	CLR ERSNEG	:SET SIGN FLAG TO +
8885	022060	005002				CLR R2	
8886	022062	000722				BR 2\$	
8887							
8888	022064	012737	177777	025062	101\$:	MOV #177777,ERSNEG	:SET SIGN FLAG TO '-'
8889	022072	005002				CLR R2	
8890	022074	000715				BR 2\$	
8891	022076	122737	000020	032133	11\$:	CMPB #020,ETMPR1+1	:IS TYPE = UNSIGNED ZONED?
8892	022104	001004				BNE 12\$	:BRANCH IF NO
8893	022106	060100				ADD R1,R0	
8894	022110	114037	025166			MOVB -(R0),SGNBYT	:SAVE SIGN BYTE
8895	022114	000757				BR 100\$	
8896	022116	122737	000040	032133	12\$:	CMPB #040,ETMPR1+1	:IS TYPE = TRAILING OVERPUNCH?
8897	022124	001005				BNE 13\$	:BRANCH IF NO
8898	022126	005301				DEC R1	
8899	022130	004737	022230		120\$:	JSR PC,DECZO	
8900	022134	005002				CLR R2	
8901	022136	000674				BR 2\$	
8902							
8903	022140	122737	000060	032133	13\$:	CMPB #060,ETMPR1+1	:IS TYPE = LEADING OVERPUNCH?
8904	022146	001002				BNE 14\$	:BRANCH IF NO
8905	022150	005001				CLR R1	
8906	022152	000766				BR 120\$	
8907	022154	122737	000100	032133	14\$:	CMPB #100,ETMPR1+1	:IS TYPE = TRAILING SEPARATE
8908	022162	001007				BNE 15\$	:BRANCH IF NO
8909	022164	060100				ADD R1,R0	:FORM ADDRESS OF SIGN
8910	022166	111037	025166			MOVB (R0),SGNBYT	:SAVE SIGN BYTE
8911	022172	121027	000055			CMPB (R0),#55	:IS SIGN = '-'
8912	022176	001326				BNE 100\$	:BRANCH IF NO
8913	022200	000731				BR 101\$	
8914	022202	122737	000120	032133	15\$:	CMPB #120,ETMPR1+1	:IS TYPE=LEADING SEPARATE
8915	022210	001401				BEQ 115\$	
8916	022212	000000				HALT	:ILLEGAL ZONED DATA TYPE
8917	022214	114037	025166		115\$:	MOVB -(R0),SGNBYT	

```

8918 022220 121027 000055          CMPB (R0),#55          ;IS SIGN = '-' ?
8919 022224 001313          BNE 100$              ;BRANCH IF NO
8920 022226 000716          BR 101$
8921
8922
8923
8924
8925
8926 022230 060100          :SUBROUTINE TO DECODE ZONED OVERPUNCH SIGN DIGIT BYTE
8927          DECZO:          ADD R1,R0          ;FIND DIGIT BY DECODING
8928 022232 111002          MOV B (R0),R2        ;DIGIT RETURNED IN R2; SIGN IN ERSNEG
8929 022234 010237 025166    MOV R2,SGNBYT        ;GET ENCODED BYTE
8930 022240 042702 177417    BIC #177417,R2      ;SAVE SIGN BYTE FOR POSSIBLE ERROR PRINTOUT
8931 022244 020227 000160    CMP R2,#160         ;LOOK AT HIGH NIBBLE
8932 022250 001014          BNE 1$              ;IS HIGH NIBBLE A 7
8933 022252 111002          MOV B (R0),R2        ;DIGIT = 0
8934 022254 032702 000002    5$: BIT #2,R2        ;IS SIGN + OR -
8935 022260 001403          BEQ 2$              ;BRANCH IF -
8936 022262 005037 025062    CLR ERSNEG
8937 022266 000403          BR 3$
8938 022270 012737 177777 025062 2$: MOV #177777,ERSNEG
8939 022276 005002          3$: CLR R2
8940 022300 000207          RTS PC
8941 022302 020227 000120    1$: CMP R2,#120        ;IS HIGH NIBBLE A 5?
8942 022306 001014          BNE 4$              ;BRANCH IF NO
8943 022310 111002          MOV B (R0),R2
8944 022312 032702 000010    BIT #10,R2          ;IS DIGIT = 0?
8945 022316 001356          BNE 5$              ;BRANCH IF YES
8946 022320 012737 177777 025062 5$: MOV #177777,ERSNEG
8947 022326 042702 177760    BIC #177760,R2      ;DIGIT IS NEG
8948 022332 062702 000007    ADD #7,R2
8949 022336 000207          RTS PC
8950 022340 020227 000060    4$: CMP R2,#060        ;IS HIGH NIBBLE A 3?
8951 022344 001006          BNE 6$              ;BRANCH IF NO
8952 022346 005037 025062    CLR ERSNEG          ;SIGN IS POSITIVE
8953 022352 111002          MOV B (R0),R2
8954 022354 042702 177760    BIC #177760,R2      ;DIGIT = LOW NIBBLE OF BYTE
8955 022360 000207          RTS PC
8956 022362 020227 000100    6$: CMP R2,#100        ;IS HIGH NIBBLE A 4?
8957 022366 001404          BEQ 61$
8958 022370 005737 002304    TST PRTSGN          ;DECIMAL PRINTING IN PROGRESS?
8959 022374 001016          BNE 62$              ;BRANCH IF YES (DONT WANT TO HALT IN
8960          ; MIDDLE OF ERROR PRINTOUT)
8961 022376 000000          HALT                ;ILLEGAL ENCODING OF OVERPUNCH DIGIT
8962 022400 111002          61$: MOV B (R0),R2
8963 022402 042702 177760    BIC #177760,R2
8964 022406 020227 000012    CMP R2,#12          ;IS DIGIT POSITIVE?
8965 022412 103407          BLC 62$              ;BRANCH IF YES
8966 022414 012737 177777 025062 62$: MOV #177777,ERSNEG
8967 022422 042702 000010    BIC #10,R2
8968 022426 005302          DEC R2
8969 022430 000207          RTS PC
8970 022432 005037 025062    62$: CLR ERSNEG
8971 022436 000207          RTS PC

```

8973  
8974  
8975  
8976  
8977  
8978  
8979  
8980  
8981  
8982  
8983  
8984  
8985  
8986  
8987  
8988  
8989  
8990  
8991  
8992  
8993  
8994  
8995  
8996 022440 005737 026322  
8997 022444 100403  
8998 022446 004737 022716  
8999 022452 000402  
9000 022454 004737 022462  
9001 022460 000207  
9002  
9003  
9004 022462 010037 032130  
9005 022466 010137 032132  
9006 022472 042701 070000  
9007 022476 010237 032134  
9008 022502 005737 025112  
9009 022506 100401  
9010 022510 005201  
9011 022512 010137 025074  
9012 022516 006237 025074  
9013 022522 063700 025074  
9014 022526 042702 177760  
9015 022532 123737 025130 032132  
9016 022540 001021  
9017 022542 122737 000160 032133  
9018 022550 001005  
9019 022552 112702 000017  
9020 022556 005037 025062  
9021 022562 000410  
9022 022564 005737 025062  
9023 022570 001403  
9024 022572 112702 000015  
9025 022576 000402  
9026 022600 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 = TYPE & 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$: JSR PC,EPUDT              ;PACKED
            2$: 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

```

```

9027 022604 010237 017402          5$:  MOV R2,TEMP          ;HOLD DATA
9028 022610 012737 000017 025106  4$:  MOV #17,EMASK       ;LOW NIBBLE MASK
9029 022616 032701 000001          BIT #1,R1          ;WHICH NIBBLE
9030 022622 001013          BNE 2$           ;LOW
9031 022624 006337 017402          ASL TEMP          ;POSITION FOR HIGH NIBBLE
9032 022630 006337 017402          ASL TEMP
9033 022634 006337 017402          ASL TEMP
9034 022640 006337 017402          ASL TEMP
9035 022644 012737 000360 025106  2$:  MOV #360,EMASK      ;NIBBLE MASK
9036 022652 143710 025106          BICB EMASK,(R0)   ;CLEAR NIBBLE BEFORE LOAD
9037 022656 153710 017402          BISB TEMP,(R0)   ;LOAD NIBBLE
9038 022662 005737 025112          TST EODD         ;IF NUMBER IS EVEN LENGTH
9039 022666 100404          BMI 3$           ;AND THIS IS LAST DIGIT
9040 022670 005037 017402          CLR TEMP
9041 022674 005301          DEC R1
9042 022676 001744          BEQ 4$           ;THEN CLEAR UNUSED NIBBLE
9043 022700 013700 032130          3$:  MOV ETMPRO,R0     ;RESTORE REGISTERS
9044 022704 013701 032132          MOV ETMPR1,R1
9045 022710 013702 032134          MOV ETMPR2,R2
9046 022714 000207          RTS PC
9047
9048          ;ZONED STRING DIGIT PUSHER
9049 022716 010037 032130          EPUDTZ:  MOV R0,ETMPRO     ;SAVE REGISTER
9050 022722 010137 032132          MOV R1,ETMPR1
9051 022726 042701 070000          BIC #070000,R1
9052 022732 010237 032134          MOV R2,ETMPR2
9053 022736 042702 177760          BIC #177760,R2   ;MASK OFF JUNK
9054 022742 052702 000060          BIS #60,R2       ;ADD JUNK
9055 022746 123737 025130 032132  CMPB ELSD,ETMPR1 ;REQUEST TO PUSH SIGN?
9056 022754 001411          BEQ 1$           ;BRANCH IF YES
9057 022756 060100          ADD R1,R0        ;WORD ADDRESS
9058 022760 110210          MOVB R2,(R0)     ;DEPOSIT DATA
9059 022762 013700 032130          2$:  MOV ETMPRO,R0     ;RESTORE REGISTER
9060 022766 013701 032132          MOV ETMPR1,R1
9061 022772 013702 032134          MOV ETMPR2,R2
9062 022776 000207          RTS PC
9063 023000 105737 025130          1$:  TSTB ELSD        ;IS STRING 0 IN LEN
9064 023004 001015          BNE 3$           ;BRANCH IF NO
9065 023006 122737 000100 032133  CMPB #100,ETMPR1+1 ;IS SIGN = TRAILING SEPARATE
9066 023014 001001          BNE 100$
9067 023016 000472          BR 14$          ;PUSH TRAILING SEP SIGN INTO 'A'
9068 023020 122737 000120 032133  100$: CMPB #120,ETMPR1+1 ;IS STRING TYPE = LEADING SEPARATE
9069 023026 001001          BNE 4$           ;BRANCH IF NO
9070 023030 000503          BR 16$          ;PUSH LEADING SEPARATE SIGN INTO 'A-1'
9071 023032 005037 025062          4$:  CLR ERSNEG
9072 023036 000751          BR 2$
9073 023040 122737 000000 032133  3$:  CMPB #000,ETMPR1+1 ;IS TYPE=SIGNED ZONED
9074 023046 001021          BNE 5$           ;BRANCH IF NO
9075 023050 005737 025062          TST ERSNEG       ;WHAT SIGN IS TO BE STORED
9076 023054 001003          BNE 6$
9077 023056 112702 000003          MOVB #3,R2       ;STORE + SIGN
9078 023062 000402          BR 7$
9079 023064 112702 000007          6$:  MOVB #7,R2       ;STORE - SIGN
9080 023070 060100          7$:  ADD R1,R0        ;FORM SIGN BYTE ADDRESS

```

9081	023072	142740	000360			BICB #360,-(R0)	:CLEAR POSITION FOR SIGN
9082	023076	006302				ASL R2	:SHIFT SIGN INTO HIGH NIBBLE POSITION
9083	023100	006302				ASL R2	
9084	023102	006302				ASL R2	
9085	023104	006302				ASL R2	
9086	023106	150210				BISB R2,(R0)	:INSERT SIGN
9087	023110	000724				BR 2\$	
9088	023112	122737	000020	032133	5\$:	CMPB #020,ETMPR1+1	:IS TYPE = UNSIGNED ZONED
9089	023120	001001				BNE 10\$	:BRANCH IF NO
9090	023122	000743				BR 4\$	:SIGN = 3
9091	023124	122737	000040	032133	10\$:	CMPB #040,ETMPR1+1	:IS TYPE = TRAILING OVERPUNCHED?
9092	023132	001016				BNE 11\$	
9093	023134	060100				ADD R1,R0	
9094	023136	114002			15\$:	MOVB -(R0),R2	:PICKUP DIGIT TO BE ENCODED WITH SIGN
9095	023140	042702	000060			BIC #60,R2	:STRIP JUNK THAT WAS PREVIOUSLY ADDED.
9096	023144	005737	025062			TST ERSNEG	:IS SIGN NEGATIVE?
9097	023150	001004				BNE 12\$	:BRANCH IF YES
9098	023152	062702	025142			ADD #OPEPTB,R2	:ENCODE DIGIT WITH + SIGN
9099	023156	111210			13\$:	MOVB (R2),(R0)	:STORE ENCODED BYTE IN STRING.
9100	023160	000700				BR 2\$	
9101							
9102	023162	062702	025154		12\$:	ADD #OPENTB,R2	:ENCODE DIGIT WITH - SIGN
9103	023166	000773				BR 13\$	:STORE ENCODED BYTE IN STRING.
9104	023170	122737	000060	032133	11\$:	CMPB #060,ETMPR1+1	:IS TYPE = LEADING OVERPUNCH?
9105	023176	001002				BNE 14\$	:BRANCH IF NO
9106	023200	005200				INC R0	:ADJUST R0 TO ENABLE USE
9107							: OF TRAILING OVERPUNCH ROUTINE
9108	023202	000755				BR 15\$	
9109							
9110	023204	122737	000100	032133	14\$:	CMPB #100,ETMPR1+1	:IS TYPE=TRAILING SEPARATE?
9111	023212	001012				BNE 16\$	:BRANCH IF NO
9112	023214	060100				ADD R1,R0	
9113	023216	005737	025062		21\$:	TST ERSNEG	:IS SIGN NEGATIVE?
9114	023222	001003				BNE 17\$	:BRANCH IF YES
9115	023224	112710	000053			MOVB #053,(R0)	:STORE + SIGN
9116	023230	000654				BR 2\$	
9117	023232	112710	000055		17\$:	MOVB #055,(R0)	:STORE - SIGN
9118	023236	000651				BR 2\$	
9119							
9120	023240	122737	000120	032133	16\$:	CMPB #120,ETMPR1+1	:IS TYPE = LEADING SEPARATE
9121	023246	001401				BEQ 20\$	:BRANCH IF YES
9122	023250	000000				HALT	:ILLEGAL ZONED DATA TYPE
9123	023252	005300			20\$:	DEC R0	
9124	023254	000760				BR 21\$	
9125							

9127						.SBTTL	DECIMAL ADDER
9128	023256	063737	025070	025072	EDCAD:	ADD EDCOPA,EDCOPB	;ADD TWO SOURCE DIGITS
9129	023264	005737	025066			TST ECARRY	;ANY CARRY OR BORROW
9130	023270	001410				BEQ 1\$	;NO
9131	023272	105737	025067			TSTB ECARRY+1	;WHICH ONE
9132	023276	001403				BEQ 2\$	
9133	023300	005337	025072			DEC EDCOPB	;BORROW
9134	023304	000402				BR 1\$	
9135	023306	005237	025072		2\$:	INC EDCOPB	;CARRY
9136	023312	005037	025066		1\$:	CLR ECARRY	;RESET CARRY/BORROW FLAG
9137	023316	005737	025072			TST EDCOPB	;IS RESULT NEGATIVE
9138	023322	100006				BPL 3\$	;NO
9139	023324	062737	000012	025072		ADD #12,EDCOPB	;MAKE VALUE POSITIVE
9140	023332	112737	000377	025067		MOVB #377,ECARRY+1	;SET BORROW FLAG
9141	023340	023727	025072	000011	3\$:	CMP EDCOPB,#11	;IS RESULT > 9
9142	023346	101411				BLOS 4\$	;NO
9143	023350	062737	000006	025072		ADD #6,EDCOPB	;CONVERT TO DECIMAL
9144	023356	042737	177760	025072		BIC #177760,EDCOPB	
9145	023364	112737	000377	025066		MOVB #377,ECARRY	
9146	023372	053737	025072	025104	4\$:	BIS EDCOPB,EADSUM	;RESULT = 0 INDICATOR
9147	023400	000207				RTS PC	
9148							



9150						.SBTTL	ADDP,ADDN,SUBP,SUBN INSTRUCTIONS
9151	023402	012737	177777	026322	EADDP:	MOV #177777,EPAK	:INDICATE PACKED MODE
9152	023410	000430				BR EADSUB	
9153	023412	005037	026322		EADDN:	CLR EPAK	:INDICATE ZONED MODE
9154	023416	000425				BR EADSUB	
9155	023420	012737	177777	026322	ESUBP:	MOV #177777,EPAK	:INDICATE PACKED MODE
9156	023426	012737	177777	025124		MOV #177777,ESUBF	:SET SUBTRACT FLAG
9157	023434	005037	025112			CLR EODD	
9158	023440	032737	000001	017364		BIT #1,E0	:IS NUMBER ODD LENGTH
9159	023446	001411				BEQ EADSUB	:NO
9160	023450	012737	177777	025112		MOV #177777,EODD	
9161	023456	000405				BR EADSUB	
9162	023460	005037	026322		ESUBN:	CLR EPAK	:INDICATE ZONED MODE
9163	023464	012737	177777	025124		MOV #177777,ESUBF	:SET SUBTRACT FLAG
9164	023472	005037	025114		EADSUB:	CLR EAODD	:ODD SIZE INDICATORS
9165	023476	005037	025116			CLR EBODD	
9166	023502	005037	025120			CLR ECODD	
9167	023506	013737	017364	025134		MOV E0,TE0	:SAVE STRING LEN WORDS
9168	023514	013737	017370	025136		MOV E2,TE2	
9169	023522	013737	017374	025140		MOV E4,TE4	
9170	023530	032737	000001	017364		BIT #1,E0	:IF ODD SIZE SET INDICATOR
9171	023536	001403				BEQ 10\$	:EVEN NO. OF DIGITS
9172	023540	012737	177777	025114		MOV #177777,EAODD	:SHOW ITS ODD
9173	023546	032737	000001	017370	10\$:	BIT #1,E2	:IF ODD SIZE SET INDICATOR
9174	023554	001403				BEQ 11\$	:EVEN NO. OF DIGITS
9175	023556	012737	177777	025116		MOV #177777,EBODD	:SHOW ITS ODD
9176	023564	032737	000001	017374	11\$:	BIT #1,E4	:IS RESULT ODD LENGTH
9177	023572	001403				BEQ 12\$	:NO
9178	023574	012737	177777	025120		MOV #177777,ECODD	:SET ODD INDICATOR
9179	023602	013700	017366		12\$:	MOV E1,R0	:FIND SIGN OF SRC1
9180	023606	013701	017364			MOV E0,R1	
9181	023612	013737	017364	025130		MOV E0,ELSD	
9182	023620	013737	025114	025112		MOV EAODD,EODD	
9183	023626	004737	021372			JSR PC,ESNK	
9184	023632	005037	025052			CLR ES1	
9185	023636	005737	025124			TST ESUBF	:SUBTRACT INST?
9186	023642	001007				BNE 1\$	:BRANCH IF YES
9187	023644	005737	025062			TST ERSNEG	:IS SIGN NEGATIVE
9188	023650	001407				BEQ 2\$	:BRANCH IF NO
9189	023652	012737	177777	025052	3\$:	MOV #177777,ES1	:SET NEGATIVE INDICATOR
9190	023660	000403				BR 2\$	
9191	023662	005737	025062		1\$:	TST ERSNEG	:SUBT INST - IS SIGN NEGATIVE
9192	023666	001771				BEQ 3\$	:BRANCH IF NO TO SET NEG. INDICATOR
9193							:NOTE: FOR SUBT, THE SIGN OF SRC1
9194							: IS INVERTED AND THE ADD IS USED.
9195	023670	013700	017372		2\$:	MOV E3,R0	:FIND SIGN OF SRC2
9196	023674	013701	017370			MOV E2,R1	
9197	023700	013737	017370	025130		MOV E2,ELSD	
9198	023706	013737	025116	025112		MOV EBODD,EODD	
9199	023714	004737	021372			JSR PC,ESNK	
9200	023720	005037	025054			CLR ES2	
9201	023724	005737	025062			TST ERSNEG	:IS SIGN NEGATIVE?
9202	023730	001403				BEQ EADSB1	:BRANCH IF NO
9203	023732	012737	177777	025054		MOV #177777,ES2	:SET NEGATIVE INDICATOR

9204											
9205	023740	005037	026332		EADSB1:	CLR EVTSSV				:RESET RESULT SIZE INDICATOR	
9206	023744	005037	025056			CLR EANEG				:RESET SUBTRACT FLAGS	
9207	023750	005037	025060			CLR EBNEG					
9208	023754	023737	025052	025054		CMP ES1,ES2				:ADD OR SUBTRACT DISPATCH	
9209	023762	001527				BEQ EADAD				:LIKE SIGNS ADD	
9210	023764	113703	017370			MOVB E2,R3				:UNLIKE SIGNS SUBTRACT	
9211	023770	123737	017364	017370		CMPB E0,E2				:WHO IS LONGER	
9212	023776	103402				BLO 1\$					
9213	024000	113703	017364			MOVB E0,R3				:MAXIMUM LENGTH	
9214	024004	113704	017364		1\$:	MOVB E0,R4				:POSITION OF SCR1	
9215	024010	113705	017370			MOVB E2,R5				:POSITION OF SCR2	
9216	024014	160304				SUB R3,R4				:START POSITION	
9217	024016	160305				SUB R3,R5					
9218	024020	123704	017364		6\$:	CMPB E0,R4				:EXIT HERE INDICATES	
9219	024024	001470				BEQ EAIS				:RESULT = 0	
9220	024026	005037	025046			CLR EOPA				:RESET DATA REGS	
9221	024032	005037	025050			CLR EOPB					
9222	024036	005704				TST R4				:IS POSITION OF SCR1 VALID	
9223	024040	100424				BMI 2\$				:NO	
9224	024042	013700	017366			MOV E1,R0				:GIT A OPERAND	
9225	024046	010401				MOV R4,R1					
9226	024050	013737	017364	025130		MOV E0,ELSD					
9227	024056	013737	025114	025112		MOV EAADD,EODD					
9228	024064	013737	017364	025132		MOV E0,TEMPE				:SET TYPE FIELD	
9229	024072	105037	025132			CLRB TEMPE					
9230	024076	053701	025132			BIS TEMPE,R1					
9231	024102	004737	021372			JSR PC,ESNK					
9232	024106	010237	025046			MOV R2,EOPA				:DATA FOR COMPARE	
9233	024112	005705			2\$:	TST R5				:IS POSITION OF SCR2 VALID	
9234	024114	100424				BMI 4\$				:NO	
9235	024116	013700	017372			MOV E3,R0				:GIT B OPERAND	
9236	024122	010501				MOV R5,R1					
9237	024124	013737	017370	025130		MOV E2,LLSD					
9238	024132	013737	025116	025112		MOV EBODD,EGDD					
9239	024140	013737	017370	025132		MOV E2,TEMPE				:SET TYPE FIELD	
9240	024146	105037	025132			CLRB TEMPE					
9241	024152	053701	025132			BIS TEMPE,R1					
9242	024156	004737	021372			JSR PC,ESNK					
9243	024162	010237	025050			MOV R2,EOPB				:DATA FOR COMPARE	
9244	024166	023737	025046	025050	4\$:	CMP EOPA,EOPB				:WHO IS LARGER	
9245	024174	101004				BHI EAIS				:A IS	
9246	024176	103412				BLO EBIS				:B IS	
9247	024200	005204				INC R4				:OH NO THEIR THE SAME	
9248	024202	005205				INC R5					
9249	024204	000705				BR 6\$					
9250	024206	012737	177777	025060	EAIS:	MOV #177777,EBNEG				:MAKE B NEGATIVE	
9251	024214	013737	025052	025062		MOV ES1,ERSNEG				:IF A IS NEGATIVE THEN RESULT IS	
9252	024222	000412				BR EADAD1					
9253	024224	012737	177777	025056	EBIS:	MOV #177777,EANEG				:MAKE A NEGATIVE	
9254	024232	013737	025054	025062		MOV ES2,ERSNEG				:IF B IS NEGATIVE THEN RESULT IS	
9255	024240	000403				BR EADAD1					
9256	024242	013737	025052	025062	EADAD:	MOV ES1,ERSNEG				:SIGN OF RESULT	
9257	024250	005037	025104		EADAD1:	CLR EADSUM				:RESULT =0 INDICATOR	

9258	024254	013737	025062	025126		MOV ERSNEG,SAVSGN	;SAVE SIGN OF RESULT
9259	024262	012700	026126			MOV #EVRTAB+^D40,R0	;CLEAR DATA AREA
9260	024266	005040			21\$:	CLR -(R0)	
9261	024270	020027	026056			CMP R0,#EVRTAB	
9262	024274	001374				BNE 21\$	
9263	024276	012703	026126			MOV #EVRTAB+^D40,R3	;DEST. POINTER
9264	024302	005037	025066			CLR ECARRY	;RESET CARRY
9265	024306	005037	026326			CLR EVTPAS	;RESET PASS COUNTER
9266	024312	005037	025064		20\$:	CLR ENOA	;NO A OPERAND FLAG
9267	024316	105337	017364			DECB E0	;A DIGIT POINTER
9268	024322	100005				BPL 1\$	
9269	024324	005237	025064			INC ENOA	;NO DIGITS LEFT
9270	024330	005037	025070			CLR EDCOPA	;DIGIT = 0
9271	024334	000404				BR 2\$	
9272	024336	004737	024442		1\$:	JSR PC,EGTOPA	;GIT A OPERAND
9273	024342	010237	025070			MOV R2,EDCOPA	;SAVE VALUE
9274	024346	105337	017370		2\$:	DECB E2	;B DIGIT POINTER
9275	024352	100006				BPL 3\$	
9276	024354	005737	025064			TST ENOA	;NO DIGITS LEFT
9277	024360	001077				BNE EXT	
9278	024362	005037	025072			CLR EDCOPB	
9279	024366	000404				BR 4\$	
9280	024370	004737	024474		3\$:	JSR PC,EGTOPB	;GIT B OPERAND
9281	024374	010237	025072			MOV R2,EDCOPB	;SAVE VALUE
9282	024400	005737	025056		4\$:	TST EANEG	;WANT A COMPLEMENTED
9283	024404	100004				BPL 5\$	;NO
9284	024406	005137	025070			COM EDCOPA	;YES
9285	024412	005237	025070			INC EDCOPA	
9286	024416	005737	025060		5\$:	TST EBNEG	;WANT B COMPLEMENTED
9287	024422	100004				BPL 6\$	
9288	024424	005137	025072			COM EDCOPB	;YES
9289	024430	005237	025072			INC EDCOPB	
9290	024434	004737	024526		6\$:	JSR PC,EADIT	;ADD DIGITS STORE RESULT
9291	024440	000724				BR 20\$	
9292							
9293	024442				EGTOPA:		;SUBROUTINE TO GET 'A' OPERAND
9294	024442	013700	017366			MOV E1,R0	;START ADDRESS OF NUMBER
9295	024446	013701	017364			MOV E0,R1	;DIGIT OF NUMBER
9296	024452	013737	025114	025112		MOV EAODD,EODD	
9297	024460	013737	025134	025130		MOV TE0,ELSD	
9298	024466	004737	021372			JSR PC,ESNK	;CALL ROUTINE TO RETRIEVE DIGIT
9299	024472	000207				RTS PC	
9300							
9301	024474				EGTOPB:		;SUBROUTINE TO GET 'B' OPERAND
9302	024474	013737	025116	025112		MOV EBODD,EODD	
9303	024502	013700	017372			MOV E3,R0	;START ADDRESS OF NUMBER
9304	024506	013701	017370			MOV E2,R1	;DIGIT OF NUMBER
9305	024512	013737	025136	025130		MOV TE2,ELSD	
9306	024520	004737	021372			JSR PC,ESNK	;CALL ROUTINE TO RETRIEVE DIGIT
9307	024524	000207				RTS PC	
9308							
9309	024526	004737	023256		EADIT:	JSR PC,EDCAD	;ADD TWO DECIMAL DIGITS
9310	024532	005237	026326			INC EVTPAS	;BUMP PASS COUNTER
9311	024536	005737	025072			TST EDCOPB	;SAVE POSITION OF LAST VALID DIGIT

9312	024542	001403				BEQ 1\$	
9313	024544	013737	026326	026332		MOV EVTPAS,EVTSSV	;SAVE POSITION
9314	024552	113743	025072		1\$:	MOVB EDCOPB,-(R3)	;SAVE RESULT
9315	024556	000207				RTS PC	
9316	024560	005037	032146		EXT:	CLR ENZI	;INITIALIZE NONZERO DIGIT STORED INDICATOR TO ZE
9317	024564	105737	025066			TSTB ECARRY	;ANY CARRY FROM LAST ADD
9318	024570	100007				BPL 1\$	;NO
9319	024572	112743	000001			MOVB #1,-(R3)	;ADD CARRY TO WORD
9320	024576	005237	026326			INC EVTPAS	;BUMP PASS COUNTER
9321	024602	013737	026326	026332		MOV EVTPAS,EVTSSV	;STORE POSITION
9322	024610	012703	026126		1\$:	MOV #EVRTAB+^D40,R3	;ADDRESS OF DATA
9323	024614	105337	017374		2\$:	DECB E4	;LAST TRANSFER COMPLETE
9324	024620	100423				BMI 3\$	;YES
9325	024622	013700	017376			MOV E5,R0	;START ADDRESS OF DEST.
9326	024626	013701	017374			MOV E4,R1	;DIGIT POSITION
9327	024632	114302				MOVB -(R3),P2	;DATA TO DEPOSIT
9328	024634	005702				TST R2	;CHECK DIGIT BEING STORED IN DST
9329	024636	001403				BEQ 7\$	
9330	024640	012737	177777	032146		MOV #177777,ENZI	;DIGIT NOT = 0, SET INDICATOR
9331	024646	013737	025120	025112	7\$:	MOV ECODD,EODD	
9332	024654	013737	025140	025130		MOV TE4,ELSD	
9333	024662	004737	022440			JSR PC,EPUSH	;CALL ROUTINE TO PUSH DIGIT ONTO DST STRING
9334	024666	000752				BR 2\$	
9335	024670	013701	025140		3\$:	MOV TE4,R1	;POSITION OF SIGN
9336	024674	013737	025126	025062		MOV SAVSGN,ERSNEG	;SETUP ERSNEG WITH RESULT SIGN
9337	024702	013737	025140	025130		MOV TE4,ELSD	
9338	024710	005737	026332			TST EVTSSV	;IF ZERO, SIGN = +
9339	024714	001002				BNE 10\$	
9340	024716	005037	025062			CLR ERSNEG	;SET POSITIVE
9341	024722	013700	017376		10\$:	MOV E5,R0	;START OF DEST.
9342	024726	013737	025120	025112		MOV ECODD,EODD	
9343	024734	004737	022440			JSR PC,EPUSH	;SAVE SIGN
9344	024740	005077	172416		EXT1:	CLR @EOPSW	;RESET EMULATE PSW
9345	024744	005737	032146			TST ENZI	;IF = 0 SET Z BIT & SKIP OVER SETTING OF N BIT.
9346	024750	001407				BEQ 1\$	
9347	024752	005737	025062			TST ERSNEG	;IF (-) SET N BIT
9348	024756	100007				BPL 2\$	
9349	024760	052777	000010	172374		BIS #10,@EOPSW	;SET N BIT
9350	024766	000403				BR 2\$	
9351	024770	052777	000004	172364	1\$:	BIS #4,@EOPSW	;COMP. LENGTH FOR OVERFLOW
9352	024776	123737	025140	026332	2\$:	CMPB TE4,EVTSSV	;FIND LENGTH OF RESULT
9353	025004	103003				BHIS 4\$	
9354	025006	052777	000002	172346		BIS #2,@EOPSW	;SET OVERFLOW
9355	025014	013702	017360		4\$:	MOV EORSTK,R2	;REGISTER UNLOAD
9356	025020	005022				CLR (R2)+	;R0 = 0
9357	025022	005022				CLR (R2)+	;R1 = 0
9358	025024	005022				CLR (R2)+	;P2 = 0
9359	025026	005022				CLR (R2)+	;R3 = 0
9360	025030	013722	025140			MOV TE4,(R2)+	;R4 = R4
9361	025034	013722	017376			MOV E5,(R2)+	;R5 = R5
9362	025040	005037	025124			CLR ESUBF	;CLEAR SUBTRACT FLAG
9363	025044	000207				RTS PC	

9365 025046 000000  
9366 025050 000000  
9367 025052 000000  
9368 025054 000000  
9369 025056 000000  
9370 025060 000000  
9371 025062 000000  
9372 025064 000000  
9373 025066 000000  
9374 025070 000000  
9375 025072 000000  
9376 025074 000000  
9377 025076 000000  
9378 025100 000000  
9379 025102 000000  
9380 025104 000000  
9381 025106 000000  
9382 025110 000000  
9383 025112 000000  
9384 025114 000000  
9385 025116 000000  
9386 025120 000000  
9387 025122 000000  
9388 025124 000000  
9389 025126 000000  
9390 025130 000000  
9391 025132 000000  
9392 025134 000000  
9393 025136 000000  
9394 025140 000000  
9395 025142 000000  
9396 025142 173  
9397 025143 101  
9398 025144 102  
9399 025145 103  
9400 025146 104  
9401 025147 105  
9402 025150 106  
9403 025151 107  
9404 025152 110  
9405 025153 111  
9406 025154 175  
9407 025155 112  
9408 025156 113  
9409 025157 114  
9410 025160 115  
9411 025161 116  
9412 025162 117  
9413 025163 120  
9414 025164 121  
9415 025165 122  
9416 025166 000000

EOPA: .WORD 0  
EOPB: .WORD 0  
ES1: .WORD 0  
ES2: .WORD 0  
EANEG: .WORD 0  
EBNEG: .WORD 0  
ERSNEG: .WORD 0  
ENOA: .WORD 0  
ECARRY: .WORD 0  
EDCOPA: .WORD 0  
EDCOPB: .WORD 0  
EFINDA: .WORD 0  
ESGNA: .WORD 0  
ESGNB: .WORD 0  
ESGNC: .WORD 0  
EADSUM: .WORD 0  
EMASK: .WORD 0  
TEMP1: .WORD 0  
EODD: .WORD 0  
EAODD: .WORD 0  
EBODD: .WORD 0  
ECODD: .WORD 0  
ESLSD: .WORD 0  
ESUBF: .WORD 0  
SAVSGN: .WORD 0  
ELSD: .WORD 0  
TEMPE: .WORD 0  
TE0: .WORD 0  
TE2: .WORD 0  
TE4: .WORD 0  
OPEPTB: .BYTE 173  
OPENTB: .BYTE 175  
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

```

          .SBTTL          CVTLP.Z INSTRUCTIONS
          CONVERT LONG FORMAT NUMBERS INTO DECIMAL
          IN EITHER PACKED OR ZONED FORMAT.
-----
9418
9419
9420
9421
9422 025170 005037 026322      ECVTLN:      CLR EPAK          ;ZONED FORMATED OUTPUT
9423 025174 000403              BR ECVT
9424 025176 012737 177777 026322  ECVTLP:      MOV #177777,EPAK    ;PACKED FORMAT OUTPUT
9425 025204 005037 025112      ECVT:        CLR EODD          ;RESET ODD INDICATORS
9426 025210 005037 026332              CLR EVTSSV
9427 025214 005037 025070              CLR EDCOPA
9428 025220 005037 025072              CLR EDCOPB
9429 025224 005037 026330              CLR ESCF
9430 025230 032737 000001 017364      BIT #1,E0          ;IF ODD SET ODD FLAG
9431 025236 001403              BEQ 10$           ;EVEN NO. OF DIGITS
9432 025240 012737 177777 025112      MOV #177777,EODD   ;SET ODD FLAG
9433 025246 012700 026072      10$:        MOV #EVRTAB+14,R0   ;CLEAR DATA TABLE
9434 025252 013702 017370              MOV E2,R2          ;DATA TO BE CONVERTED
9435 025256 013703 017372              MOV E3,R3
9436 025262 005040      1$:        CLR -(R0)
9437 025264 022700 026056      CMP #EVRTAB,R0     ;CLEAR COMPLETE
9438 025270 001374              BNE 1$            ;NO
9439 025272 005037 026324      CLR EVTSGN        ;CLEAR OLD SIGN INFO
9440 025276 005702              TST R2             ;IS DATA NEGATIVE
9441 025300 100026              BPL 2$            ;NO
9442 025302 112737 000377 026324      MOVB #377,EVTSGN  ;YES , SAVE SIGN
9443 025310 005103              COM R3             ;NEGATE DATA
9444 025312 005102              COM R2
9445 025314 022703 177777      CMP #177777,R3    ;CHECK FOR SPECIAL CASE - MOST NEG #.
9446 025320 001007              BNE 6$            ;BRANCH IF THIS IS NOT THE SPECIAL CASE
9447 025322 022702 077777      CMP #077777,R2    ;FOR MOST NEG # SRC HIGH =100000, SRC LOW = 0.
9448 025326 001004              BNE 0$            ;BRANCH IF NOT THE SPECIAL CASE
9449 025330 012737 177777 026330      MOV #177777,ESCF  ;SET SPECIAL CASE FLAG
9450 025336 000405              BR 3$
9451 025340 000257      6$:        CCC
9452 025342 062703 000001      ADD #1,R3
9453 025346 103001              BCC 3$            ;BRANCH IF NO CARRY FROM FIRST WORD
9454 025350 005202              INC R2             ;YES
9455 025352 042702 100000      3$:        BIC #100000,R2    ;CLEAR SIGN BIT
9456 025356 005037 026326      2$:        CLR EVTTPAS   ;RESET PASS COUNTER
9457 025362 023727 026326 000174      5$:        CMP EVTTPAS,#^D124 ;31 PASSES COMPLETE YET
9458 025370 001460              BEQ EVTWRP        ;WRAP UP ROUTEEN
9459 025372 000257      CCC          ;CLEAR CC BITS FOR 32 BIT SHIFT
9460 025374 006002              ROR R2            ;SHIFT LOB INTO R3
9461 025376 006003              ROR R3            ;SHIFT LOB FOR VALUE
9462 025400 103002              BCC 4$            ;IF NOT SET BIT HAS NO VALUE
9463 025402 004737 025416      JSR PC,EVRTAD     ;ADD BIT VALUE TO DECIMAL NUMBER
9464 025406 062737 000004 026326      4$:        ADD #4,EVTTPAS  ;BUMP PASS COUNTER
9465 025414 000762              BR 5$            ;NEXT PASS
9466 025416 005037 025066      EVRTAD:      CLR ECARRY        ;RESET CARRY/BORROW FLAGS
9467 025422 012700 026070      MOV #EVRTAB+12,R0 ;A OPERAND POINTER
9468 025426 013704 026326      MOV EVTTPAS,R4    ;TABLE OFFSET
9469 025432 062704 026334      ADD #EVTABA,R4    ;TABLE START ADDRESS
9470 025436 012401      MOV (R4)+,R1      ;START ADDRESS OF DATA
9471 025440 061401      ADD (R4),R1       ;END ADDRESS OF DATA

```

9472	025442	011404			MOV (R4),R4	;NO. OF DIGITS
9473	025444	010437	026332		MOV R4,EVTSSV	
9474	025450	114037	025070		1\$: MOVB -(R0),EDCOPA	;FIND A OPERAND
9475	025454	114137	025072		MOVB -(R1),EDCOPB	;FIND B OPERAND
9476	025460	004737	023256		JSR PC,EDCAD	;ADD
9477	025464	113710	025072		MOVB EDCOPB,(R0)	;SAVE RESULT
9478	025470	005304			DEC R4	;IS THIS LAST DIGIT
9479	025472	001366			BNE 1\$	;NO
9480	025474	105737	025066		2\$: TSTB ECARRY	;ANY CARRY
9481	025500	100401			BMI 3\$	;YES
9482	025502	000207			RTS PC	
9483	025504	005237	026332		3\$: INC EVTSSV	
9484	025510	114037	025070		MOVB -(R0),EDCOPA	;A OPERAND
9485	025514	005037	025072		CLR EDCOPB	;NO B OPERAND
9486	025520	004737	023256		JSR PC,EDCAD	;ADD THE CARRY
9487	025524	113710	025072		MOVB EDCOPB,(R0)	;SAVE RESULT
9488	025530	000761			BR 2\$	
9489	025532	005037	026316		CLR ENULL	;RESET RESULT = 0 INDICATOR
9490	025536	013737	026324	025062	MOV EVTSGN,ERSNEG	;SETUP SIGN INDICATOR
9491	025544	012704	026070		MOV #EVRTAB+12,R4	;SETUP PCINTER TO CONVERT RESULT
9492	025550	113701	017364		MOVB E0,R1	
9493	025554	160104			SUB R1,R4	
9494	025556	013737	017364	025130	MOV E0,ELSD	;SETUP POINTER TO LEAST SIGN. DIGIT
9495	025564	113737	017364	026320	MOVB E0,EPOPS	
9496	025572	005337	026320		DEC EPOPS	
9497	025576	013701	017364		MOV E0,R1	;SETUP POSITION OF # TO INSERT = 0
9498	025602	042701	000377		BIC #377,R1	
9499	025606	013700	017366		MOV E1,R0	;SETUP ADDRESS OF # TO LOAD
9500	025612	005002			2\$: CLR R2	;SETUP DATA TO LOAD
9501	025614	020427	026056		CMP R4,#EVRTAB	
9502	025620	103403			BLO 1\$	
9503	025622	111402			MOVB (R4),R2	
9504	025624	050237	026316		BIS R2,ENULL	
9505	025630	005204			1\$: INC R4	
9506	025632	004737	022440		JSR PC,EPUSH	;CALL ROUTINE TO PUSH DIGIT INTO STRING
9507	025636	120137	026320		CMPB R1,EPOPS	;READY TO PUSH SIGN?
9508	025642	001004			BNE 3\$	;BRANCH IF NO
9509	025644	005737	026330		TST ESCF	;SPECIAL CASE?
9510	025650	001401			BEQ 3\$	;BRANCH IF NO
9511	025652	000405			BR EVTXT	;WORK WITH SPECIAL CASE BEFORE INSERTING SIGN.
9512	025654	020137	017364		3\$: CMP R1,E0	;ALL DIGITS PLUS SIGN PUSHED?
9513	025660	001402			BEQ EVTXT	;BRANCH IF YES
9514	025662	005201			INC R1	
9515	025664	000752			BR 2\$	;RETURN TO PUSH NEXT DIGIT
9516						
9517	025666	105737	017364		EVTXT: TSTB E0	;IF DST.LEN=0 ,DON'T BOTHER TESTING ; FOR SPECIAL CASE.
9518						
9519	025672	001425			BEG 5\$	
9520	025674	005737	026330		TST ESCF	;SPECIAL CASE? (MOST NEG #).
9521	025700	001422			BEQ 5\$	;BRANCH IF NO.
9522	025702	013737	017364	025130	MOV E0,ELSD	
9523	025710	013701	017364		MOV E0,R1	
9524	025714	005301			DEC R1	;SET R1=DST.LEN-1.
9525	025716	013700	017366		MOV E1,R0	;INCREMENT LEAST SIGN DIGIT TO 2.

9526	025722	012702	000010		MOV #10,R2	
9527	025726	004737	022440		JSR PC,EPUSH	:CALL ROUTINE TO PUSH THE 8 INTO DEST.
9528	025732	005201			INC R1	:INSERT NEGATIVE SIGN
9529	025734	012737	177777	025062	MOV #177777,ERSNEG	
9530	025742	004737	022440		7\$: JSR PC,EPUSH	:CALL ROUTINE TO PUSH NEG SIGN INTO DEST.
9531	025746	005077	171410		5\$: CLR @EOPSW	:INIT EMULATE PSW
9532	025752	005737	026316		TST ENULL	:IF = 0 SET Z BIT; SKIP SETTING OF N BIT
9533	025756	001407			BEQ 1\$	
9534	025760	105737	025062		TSTB ERSNEG	:IF (-) SET N BIT
9535	025764	100007			BPL 2\$	
9536	025766	052777	000010	171366	BIS #10,@EOPSW	:SET N BIT
9537	025774	000403			BR 2\$	
9538	025776	052777	000004	171356	1\$: BIS #4,@EOPSW	:SET Z BIT
9539	026004	123737	017364	026332	2\$: CMPB E0,EVTSSV	
9540	026012	103003			BHIS 4\$	
9541	026014	052777	000002	171340	BIS #2,@EOPSW	:SET OVERFLOW
9542	026022	013702	017360		4\$: MOV EORSTK,R2	:REGISTER UNLOAD
9543	026026	013722	017364		MOV E0,(R2)+	:R0 = R0
9544	026032	013722	017366		MOV E1,(R2)+	:R1 = R1
9545	026036	005022			CLR (R2)+	:R2 = 0
9546	026040	005022			CLR (R2)+	:R3 = 0
9547	026042	013722	017374		MOV E4,(R2)+	:R4 = R4
9548	026046	013712	017376		MOV E5,(R2)	:R5 = R5
9549	026052	000207			RTS PC	
9550	026054	000000			.WORD 0	:MUST PRECEDE EVRTAB.!
9551	026056	000240			.BLKB *D160	
9552	026316	000000			ENULL:	.WORD 0
9553	026320	000000			EPJPS:	.WORD 0
9554	026322	000000			EPAK:	.WORD 0
9555	026324	000000			EVTSGN:	.WORD 0
9556	026326	000000			EVTSPAS:	.WORD 0
9557	026330	000000			ESCF:	.WORD 0
9558	026332	000000			EVTSSV:	.WORD 0
9559	026334	026530	000001		EVTABA:	.WORD E00,1
9560	026340	026531	000001			.WORD E01,1
9561	026344	026532	000001			.WORD E02,1
9562	026350	026533	000001			.WORD E03,1
9563	026354	026534	000002			.WORD E04,2
9564	026360	026536	000002			.WORD E05,2
9565	026364	026540	000002			.WORD E06,2
9566	026370	026542	000003			.WORD E07,3
9567	026374	026545	000003			.WORD E08,3
9568	026400	026550	000003			.WORD E09,3
9569	026404	026553	000004			.WORD E10,4
9570	026410	026557	000004			.WORD E11,4
9571	026414	026563	000004			.WORD E12,4
9572	026420	026567	000004			.WORD E13,4
9573	026424	026573	000005			.WORD E14,5
9574	026430	026600	000005			.WORD E15,5
9575	026434	026605	000005			.WORD E16,5
9576	026440	026612	000006			.WORD E17,6
9577	026444	026620	000006			.WORD E18,6
9578	026450	026626	000006			.WORD E19,6
9579	026454	026634	000007			.WORD E20,7



9580	026460	026643	000007			.WORD	E21,7
9581	026464	025652	000007			.WORD	E22,7
9582	026470	026661	000007			.WORD	E23,7
9583	026474	026670	000010			.WORD	E24,10
9584	026500	026700	000010			.WORD	E25,10
9585	026504	026710	000010			.WORD	E26,10
9586	026510	026720	000011			.WORD	E27,11
9587	026514	026731	000011			.WORD	E28,11
9588	026520	026742	000011			.WORD	E29,11
9589	026524	026753	000012			.WORD	E30,12
9590	026530	001		E00:		.BYTE	^D1
9591	026531	002		E01:		.BYTE	^D2
9592	026532	004		E02:		.BYTE	^D4
9593	026533	010		E03:		.BYTE	^D8
9594	026534	001	006	E04:		.BYTE	^D1,^D6
9595	026536	003	002	E05:		.BYTE	^D3,^D2
9596	026540	006	004	E06:		.BYTE	^D6,^D4
9597	026542	001	002	010 E07:		.BYTE	^D1,^D2,^D8
9598	026545	002	005	006 E08:		.BYTE	^D2,^D5,^D6
9599	026550	005	001	002 E09:		.BYTE	^D5,^D1,^D2
9600	026553	001	000	002 E10:		.BYTE	^D1,^D0,^D2,^D4
	026556	004					
9601	026557	002	000	004 E11:		.BYTE	^D2,^D0,^D4,^D8
	026562	010					
9602	026563	004	000	011 F12:		.BYTE	^D4,^D0,^D9,^D6
	026566	006					
9603	026567	010	001	011 E13:		.BYTE	^D8,^D1,^D9,^D2
	026572	002					
9604	026573	001	006	003 E14:		.BYTE	^D1,^D6,^D3,^D8,^D4
	026576	010	004				
9605	026600	003	002	007 E15:		.BYTE	^D3,^D2,^D7,^D6,^D8
	026603	006	010				
9606	026605	006	005	005 E16:		.BYTE	^D6,^D5,^D5,^D3,^D6
	026610	003	006				
9607	026612	001	003	001 E17:		.BYTE	^D1,^D3,^D1,^D0,^D7,^D2
	026615	000	007	002			
9608	026620	002	006	002 E18:		.BYTE	^D2,^D6,^D2,^D1,^D4,^D4
	026623	001	004	004			
9609	026626	005	002	004 E19:		.BYTE	^D5,^D2,^D4,^D2,^D8,^D8
	026631	002	010	010			
9610	026634	001	000	004 E20:		.BYTE	^D1,^D0,^D4,^D8,^D5,^D7,^D6
	026637	010	005	007			
	026642	006					
9611	026643	002	000	011 E21:		.BYTE	^D2,^D0,^D9,^D7,^D1,^D5,^D2
	026646	007	001	005			
	026651	002					
9612	026652	004	001	011 E22:		.BYTE	^D4,^D1,^D9,^D4,^D3,^D0,^D4
	026655	004	003	000			
	026660	004					
9613	026661	010	003	010 E23:		.BYTE	^D8,^D3,^D8,^D8,^D6,^D0,^D8
	026664	010	006	000			
	026667	010					
9614	026670	001	006	007 E24:		.BYTE	^D1,^D6,^D7,^D7,^D7,^D2,^D1,^D6
	026673	007	007	002			

9615	026676	001	006	005	E25:	.BYTE ^D3,^D3,^D5,^D5,^D4,^D4,^D3,^D2
	026700	003	003	004		
	026703	005	004			
	026706	003	002			
9616	026710	006	007	001	E26:	.BYTE ^D6,^D7,^D1,^D0,^D8,^D8,^D6,^D4
	026713	000	010	010		
	026716	006	004			
9617	026720	001	003	004	E27:	.BYTE ^D1,^D3,^D4,^D2,^D1,^D7,^D7,^D2,^D8
	026723	002	001	007		
	026726	007	002	010		
9618	026731	002	006	010	E28:	.BYTE ^D2,^D6,^D8,^D4,^D3,^D5,^D4,^D5,^D6
	026734	004	003	005		
	026737	004	005	006		
9619	026742	005	003	006	E29:	.BYTE ^D5,^D3,^D6,^D8,^D7,^D0,^D9,^D1,^D2
	026745	010	007	000		
	026750	011	001	002		
9620	026753	001	000	007	E30:	.BYTE ^D1,^D0,^D7,^D3,^D7,^D4,^D1,^D8,^D2,^D4
	026756	003	007	004		
	026761	001	010	002		
	026764	004				
9621	026766					.EVEN

```

          .SBTTL          CONVERT PACKED OR ZONED TO LONG
          CONVERT A NUMBER THAT IS IN EITHER PACKED OR
          ZONED FORMAT INTO LONG FORMAT.          (CVTPL,CVTNL)
-----
9623
9624
9625
9626
9627 026766 005037 026322      ECVTNL:      CLR EPAK          ;ZONED FORMAT INPUT
9628 026772 000403              BR EZL1
9629 026774 012737 177777 026322  ECVTPL:      MOV #177777,EPAK      ;PACKED FORMAT INPUT
9630 027002 005037 027526      EZL1:        CLR EFLO          ;RESET OVERFLOW
9631 027006 005037 025112      CLR EODD      ;RESET ODD LENGTH INDICATOR
9632 027012 005003              CLR R3          ;CLEAR RESULT DATA AREA
9633 027014 005004              CLR R4
9634 027016 005037 026316      CLR ENULL      ;RESET NULL INDICATOR
9635 027022 032737 000001 017364  BIT #1,E0      ;IS SOURCE ODD OR EVEN NO. OF DIGITS
9636 027030 001403              BEQ 1$         ;ITS EVEN
9637 027032 012737 177777 025112  MOV #177777,EODD ;ODD
9638
9639
9640 027040 013701 017364      1$:         FIND SIGN OF NUMBER
9641 027044 013737 017364 025130  MOV E0,R1      ;SETUP POSITION OF SIGN
9642 027052 013700 017366      MOV E0,ELSD    ;SETUP POSITION OF LEAST SIGN DIGIT.
9643 027056 004737 021372      MOV E1,R0      ;SETUP ADDRESS OF STRING
9644 027062 005037 025052      JSR PC,ESNK    ;CALL ROUTINE TO FIND SIGN
9645 027066 005737 025062      CLR E$1
9646 027072 001403              TST ERSNEG     ;IS SIGN NEGATIVE?
9647 027074 012737 177777 025052  BEQ EZL2       ;BRANCH IF NO
9648
9649
9650
9651
9652
9653
9654 027102 013737 017364 027530  EZL2:      MOV E0,ECOUN   ;RESET COUNT(UPPER BYTE OF ECOUN MUST
9655
9656 027110 042737 000377 027530      BIC #377,ECOUN ; CONTAIN TYPE)
9657 027116 013737 017364 025130      MOV E0,ELSD
9658 027124 123737 027530 017364      10$:      CMPB ECOUN,E0  ;CONVERSION COMPLETE YET
9659 027132 001474              BEQ EZLE       ;YES
9660 027134 013701 027530      MOV ECOUN,R1   ;POSITION OF SOURCE
9661 027140 013700 017366      MOV E1,R0      ;START ADDRESS OF SOURCE
9662 027144 004737 021372      JSR PC,ESNK    ;CALL ROUTINE TO FIND DIGIT
9663 027150 005237 027530      2$:        INC ECOUN      ;BUMP COUNTER
9664 027154 050237 026316      BIS R2,ENULL   ;DIGIT SUM FOR NULL TEST
9665 027160 000257              CCC           ;MULTIPLY RESULT BY 10
9666 027162 006104              ROL R4
9667 027164 006103              ROL R3
9668 027166 103003              BCC 3$        ;ANY BIT SHIFTED OUT IS OVERFLOW
9669 027170 012737 177777 027526      3$:        MOV #177777,EFLO
9670 027176 010337 017402      MOV R3,TEMP    ;TEMP DATA HOLD FOR MULTIPLY
9671 027202 010437 025110      MOV R4,TEMP1
9672 027206 000257              CCC
9673 027210 006104              ROL R4
9674 027212 006103              ROL R3
9675 027214 103003              BCC 4$
9676 027216 012737 177777 027526      MOV #177777,EFLO

```

9677	027224	000257		4\$:	CCC	
9678	027226	006104			ROL R4	
9679	027230	006103			ROL R3	
9680	027232	103003			BCC 5\$	
9681	027234	012737	177777 027526		MOV #177777,EFLO	
9682	027242	063704	025110	5\$:	ADD TEMP1,R4	;COMPLETE MULTIPLY
9683	027246	103007			BCC 6\$	
9684	027250	000257			CCC	
9685	027252	062703	000001		ADD #1,R3	
9686	027256	103003			BCC 6\$	
9687	027260	012737	177777 027526		MOV #177777,EFLO	
9688	027266	063703	017402	6\$:	ADD TEMP,R3	
9689	027272	103003			BCC 8\$	
9690	027274	012737	177777 027526		MOV #177777,EFLO	
9691	027302	060204		8\$:	ADD R2,R4	;ADD NEW DIGIT
9692	027304	103307			BCC 10\$	
9693	027306	062703	000001		ADD #1,R3	
9694	027312	103304			BCC 10\$	
9695	027314	012737	177777 027526		MOV #177777,EFLO	
9696	027322	000700			BR 10\$	
9697						
9698						
9699	027324	005703		EZLE:	SET CC BITS , SET RESULT REGISTERS	
9700					TST R3	;BIT 32= 1 IS OVERFLOW EXCEPT IF
9701	027326	100013				; ALL OTHER 31 BITS = 0 & SRC WAS NEGATIVE
9702	027330	005704			BPL 6\$	;BRANCH IF OK
9703	027332	001006			TST R4	;ALL OTHER 31 BITS = 0?
9704	027334	032703	077777		BNE 7\$	;BRANCH IF NO
9705	027340	001003			BIT #77777,R3	
9706	027342	005737	025052		BNE 7\$	;BRANCH IF NO
9707	027346	100403			TST ES1	;WAS SRC NEGATIVE?
9708	027350	012737	177777 027526	7\$:	BMI 6\$	;BRANCH IF YES
9709	027356	005077	170000	6\$:	MOV #177777,EFLO	;ELSE OVERFLOW
9710	027362	005737	027526		CLR @EOPSW	;RESET PSW
9711	027366	001403			TST EFLO	;ANY OVERFLOW
9712	027370	052777	000002 167764		BEQ 1\$	;NO
9713	027376	005704		1\$:	BIS #2,@EOPSW	;SET V BIT
9714	027400	001006			TST R4	;WAS RESULT = 0
9715	027402	005703			BNE 5\$	;NO
9716	027404	001004			TST R3	
9717	027406	052777	000004 167746		BNE 5\$	;NO
9718	027414	000412			BIS #4,@EOPSW	;SET Z BIT
9719	027416	005737	025052	5\$:	BR 2\$	
9720	027422	100007			TST ES1	;WHAT SIGN
9721	027424	005103			BPL 2\$	;POSITIVE
9722	027426	005104			COM R3	;COMPLEMENT VALUE
9723	027430	000257			COM R4	
9724	027432	062704	000001		CCC	
9725	027436	103001			ADD #1,R4	
9726	027440	005203			BCC 2\$	
9727	027442	005703		2\$:	INC R3	
9728	027444	100003			TST R3	;SET N BIT BASED ON SIGN OF RESULT
9729	027446	052777	000010 167706		BPL 3\$	;IF RESULT SIGN -> BRANCH
9730	027454	032777	000004 167700	3\$:	BIS #10,@EOPSW	;SET N BIT
					BIT #4,@EOPSW	;WAS DST = 0

9731	027462	001006	
9732	027464	005737	025052
9733	027470	100003	
9734	027472	052777	000001 167662
9735	027500	013700	017360
9736	027504	005020	
9737	027506	005020	
9738	027510	010320	
9739	027512	010420	
9740	027514	013720	017374
9741	027520	013720	017376
9742	027524	000707	
9743	027526	000000	
9744	027530	000000	

4\$:

EFLO:  
ECOUN:

BNE 4\$  
TST ES1  
BPL 4\$  
BIS #1,@EOPSW  
MOV EORSTK,RO  
CLR (RO)+  
CLR (RO)+  
MOV R3,(RO)+  
MOV R4,(RO)+  
MOV E4,(RO)+  
MOV E5,(RO)+  
RTS PC  
.WORD 0  
.WORD 0

:YES  
:AND WAS SOURCE NEGATIVE  
:NO  
:YES , THEN SET C BIT  
:REGISTER SAVE  
:R0 = 0  
:R1 = 0  
:R2 = DS, HIGH  
:R3 = DST LOW  
:R4 = R4  
:R5 = R5

9746						.SBTTL	CONVERT PACKED TO ZONED
9747	027532	012737	177777	026322	ECVTPN:	MOV #177777,EPAK	;SET PACKED MODE INDICATOR
9748	027540	005037	026332			CLR EVTSSV	;RESET PASS COUNTER
9749	027544	005037	026316			CLR ENULL	;RESET NULL INDICATOR
9750	027550	005077	167606			CLR @EOPSW	;RESET EMULATE PSW
9751	027554	013737	017364	025070		MOV E0,EDCOPA	;START POSITION OF SOURCE
9752	027562	013737	017370	025072		MOV E2,EDCOPB	;START POSITION OF DEST.
9753	027570	005037	025112			CLR EODD	;IS SOURCE ODD OR EVEN NUMBER
9754	027574	032737	000001	017364		BIT #1,E0	;OF DIGITS
9755	027602	001403				BEQ EPZ1	;ITS EVEN
9756	027604	012737	177777	025112		MOV #177777,EODD	;ITS ODD
9757	027612	004737	030554		EPZ1:	JSR PC,EFMSD	;DETERMINE POSITION OF MOST SIGN SRC DIGIT
9758	027616	005002				CLR R2	
9759	027620	105337	025070			DECB EDCOPA	;END OF SOURCE YET
9760	027624	100414				BMI 1\$	;YES
9761	027626	042777	000001	167526		BIC #1,@EOPSW	;RESET END OF SOURCE FLAG
9762	027634	013700	017366			MOV E1,R0	;START ADDRESS OF SOURCE
9763	027640	013701	025070			MOV EDCOPA,R1	;POSITION OF DIGIT
9764	027644	013737	017364	025130		MOV E0,ELSD	
9765	027652	004737	021414			JSR PC,EFINDT	;GRAB DIGIT
9766	027656	105337	025072		1\$:	DECB EDCOPB	;END OF DEST. YET
9767	027662	100421				BMI EPZE	;YES
9768	027664	005237	026332			INC EVTSSV	;DIGIT POSITION
9769	027670	005702				TST R2	;IS DIGIT ZERO
9770	027672	001403				BEQ 2\$	
9771	027674	013737	026332	026316		MOV EVTSSV,ENULL	;POSITION OF LAST NON ZERO DIGIT STORED.
9772	027702	013700	017372		2\$:	MOV E3,R0	;START ADDRESS OF DEST
9773	027706	013701	025072			MOV EDCOPB,R1	;POSITION OF DIGIT
9774	027712	013737	017370	025130		MOV E2,ELSD	
9775	027720	004737	022716			JSR PC,EPUDTZ	;SAVE DIGIT
9776	027724	000732				BR EPZ1	;LOOP TILL COMPLETE
9777	027726	013700	017366		EPZE:	MOV E1,R0	;START ADDRESS OF SOURCE
9778	027732	013701	017364			MOV E0,R1	;POSITION OF SIGN
9779	027736	013737	017364	025130		MOV E0,ELSD	
9780	027744	004737	021414			JSR PC,EFINDT	;GRAB SIGN
9781	027750	005037	025052			CLR ES1	;RESET SIGN FLAG
9782	027754	005737	025062			TST ERSNEG	;IS SOURCE NEG?
9783	027760	001406				BEQ 2\$	;BRANCH IF NO
9784	027762	005737	030636			TST EMSDP	;IF SRC IS ZERO AND NEG TREAT IT AS POSITIVE
9785	027766	001403				BEQ 2\$	
9786	027770	012737	177777	025052		MOV #177777,ES1	;SET NEGATIVE FLAG
9787	027776	005737	026316		2\$:	TST ENULL	;WAS RESULT STORED ZERO
9788	030002	001004				BNE 3\$	;NO
9789	030004	052777	000004	167350		BIS #4,@EOPSW	;SET Z BIT
9790	030012	000406				BR 6\$	;SKIP SETTING OF N BIT
9791	030014	005737	025052		3\$:	TST ES1	;SIGN OF RESULT
9792	030020	001403				BEQ 6\$	;POSITIVE
9793	030022	052777	000010	167332		BIS #10,@EOPSW	;SET N BIT
9794	030030	013737	025052	025062	6\$:	MOV ES1,ERSNEG	;SETUP SIGN OF RESULT
9795	030036	013701	017370			MOV E2,R1	;POSITION OF SIGN
9796	030042	013737	017370	025130		MOV E2,ELSD	
9797	030050	013700	017372			MOV E3,R0	;START ADDRESS OF DEST.
9798	030054	004737	022716			JSR PC,EPUDTZ	;INSERT SIGN IN DST STRING
9799	030060	005737	025062			TST ERSNEG	;WAS SIGN STORED POSITIVE?(UNSIGNED)

9800	030064	001003		
9801	030066	042777	000010	167266
9802	030074	023737	030636	026316
9803	030102	101403		
9804	030104	052777	000002	167250
9805	030112	013702	017360	
9806	030116	005022		
9807	030120	005022		
9808	030122	013722	017370	
9809	030126	013722	017372	
9810	030132	013722	017374	
9811	030136	013722	017376	
9812	030142	000207		

	BNE 4\$	:BRANCH IF NO
	BIC #10,@EOPSW	:CLEAR PSW N BIT
4\$:	CMP EMSDP,ENULL	:CAN DEST. CONTAIN ALL DIGITS
	BLOS 5\$	:YES
	BIS #2,@EOPSW	:SET V BIT
5\$:	MOV EORSTK,R2	:SAVE REGISTERS
	CLR (R2)+	:R0 = 0
	CLR (R2)+	:R1 = 0
	MOV E2,(R2)+	:R2 = R2
	MOV E3,(R2)+	:R3 = R3
	MOV E4,(R2)+	:R4 = R4
	MOV E5,(R2)+	:R5 = R5
	RTS PC	

9814					.SBTTL	CONVERT ZONED TO PACKED
9815	030144	005037	026332		CLR EVTSSV	;RESET PASS COUNTER
9816	030150	005037	026322		CLR EPAK	;MODE = ZONED
9817	030154	005037	026316		CLR ENULL	;RESET NULL INDICATOR
9818	030160	005077	167176		CLR @EOPSW	;RESET EMULATE PSW
9819	030164	013737	017364	025070	MOV E0,EDCOPA	;START POSITION OF SOURCE
9820	030172	013737	017370	025072	MOV E2,EDCOPB	;START POSITION OF DEST.
9821	030200	005037	025112		CLR EODD	;IS DEST. ODD OR EVEN # OF DIGITS.
9822	030204	032737	000001	017370	BIT #1,E2	
9823	030212	001403			BEQ EZP1	;ITS EVEN
9824	030214	012737	177777	025112	MOV #177777,EODD	;ITS ODD
9825	030222	004737	030554		JSR PC,EFMSD	;FIND POSITION OF MOST SIGNIFICANT SOURCE DIGIT
9826	030226	005002			CLR R2	
9827	030230	105337	025070		DECB EDCOPA	;END OF SOURCE YET
9828	030234	100414			BMI 1\$	
9829	030236	042777	000001	167116	BIC #1,@EOPSW	;RESET END OF SOURCE FLAG
9830	030244	013700	017366		MOV E1,R0	;START ADDRESS OF SOURCE
9831	030250	013701	025070		MOV EDCOPA,R1	;POSITION OF DIGIT
9832	030254	013737	017364	025130	MOV E0,ELSD	
9833	030262	004737	021616		JSR PC,EFNDTZ	;GRAB DIGIT
9834	030266	105337	025072		DECB EDCOPB	;END OF DEST. YET
9835	030272	100421			BMI EZPE	;YES
9836	030274	005237	026332		INC EVTSSV	;PASS COUNTER
9837	030300	005702			TST R2	;IS DIGIT ZERO
9838	030302	001403			BEQ 2\$	
9839	030304	013737	026332	026316	MOV EVTSSV,ENULL	;SAVE POSITION OF LAST NON ZERO DIGIT STORED.
9840	030312	013700	017372		MOV F3,R0	;START ADDRESS OF DEST.
9841	030316	013701	025072		MOV EDCOPB,R1	;POSITION OF DIGIT
9842	030322	013737	017370	025130	MOV E2,ELSD	
9843	030330	004737	022462		JSR PC,EPUTDT	;SAVE DIGIT
9844	030334	000732			BR EZP1	;LOOP TILL COMPLETE
9845	030336	013700	017366		MOV E1,R0	;START ADDRESS OF SOURCE
9846	030342	013701	017364		MOV E0,R1	;POSITION OF SIGN
9847	030346	013737	017364	025130	MOV E0,ELSD	
9848	030354	004737	021616		JSR PC,EFNDTZ	
9849	030360	005037	025052		CLR ES1	;RESET SIGN FLAG
9850	030364	005737	025062		TST ERSNEG	;IS SOURCE NEG?
9851	030370	001406			BEQ 2\$	;NO
9852	030372	005737	030636		TST EMSDP	;IF SOURCE IS ZERO AND NEG TREAT IT AS POSITIVE
9853	030376	001403			BEQ 2\$	
9854	030400	012737	177777	025052	MOV #177777,ES1	;SET NEGATIVE FLAG
9855	030406	005737	026316		TST ENULL	;WAS RESULT STORED ZERO
9856	030412	001004			BNE 3\$	;NO
9857	030414	052777	000004	166740	BIS #4,@EOPSW	;SET Z BIT
9858	030422	000406			BR 4\$	
9859	030424	005737	025052		TST ES1	;SET SIGN OF RESULT
9860	030430	001403			REQ 4\$	;POSITIVE
9861	030432	052777	000010	166722	BIS #10,@EOPSW	;SET N BIT
9862	030440	023737	030636	026316	CMP EMSDP,ENULL	;CAN DEST. CONTAIN ALL DIGITS
9863	030446	101403			BLOS 5\$	;YES
9864	030450	052777	000002	166704	BIS #2,@EOPSW	;SET V BIT
9865	030456	013701	017370		MOV E2,R1	;POSITION OF SIGN
9866	030462	013700	017372		MOV E3,R0	;START ADDRESS OF NUMBER
9867	030466	013737	025052	025062	MOV ES1,ERSNEG	



9868	030474	013737	017370	025130		MOV E2,ELSD	
9869	030502	004737	022462			JSR PC,EPUTDT	;SAVE SIGN
9870	030506	005737	025062			TST ERSNEG	;WAS SIGN STORED POSITIVE? (UNSIGNED)
9871	030512	001003				BNE 1\$	;BRANCH IF NO
9872	030514	042777	000010	166640		BIC #10,@EOPSW	;CLEAR PSW N BIT
9873	030522	013702	017360		1\$:	MOV EORSTK,R2	;SAVE REGISTERS
9874	030526	005022				CLR (R2)+	;R0 = 0
9875	030530	005022				CLR (R2)+	;R1 = 0
9876	030532	013722	017370			MOV E2,(R2)+	;R2 = R2
9877	030536	013722	017372			MOV E3,(R2)+	;R3 = R3
9878	030542	013722	017374			MOV E4,(R2)+	;R4 = R4
9879	030546	013722	017376			MOV E5,(R2)+	;R5 = R5
9880	030552	000207				RTS PC	

9882	030554	005037	030640	EFMSD:	CLR ESDC	
9883	030560	005037	030636		CLR EMSDP	
9884	030564	013700	017366		MOV E1,R0	;SET R0=SRC.ADR
9885	030570	013701	017364		MOV E0,R1	;SET R1=SRC.PTR
9886	030574	013737	017364	025130	MOV E0,ELSD	
9887	030602	005301		1\$:	DEC R1	
9888	030604	105701			TSTB R1	;LOOKED AT ALL SRC DIGITS YET?
9889	030606	100412			BMI 2\$	;BRANCH IF YES
9890	030610	004737	021372		JSR PC,ESNK	;GET NEXT SRC DIGIT
9891	030614	005237	030640		INC ESDC	;INCREMENT DIGIT COUNTER
9892	030620	005702			TST R2	;IS DIGIT ZERO (NON-SIGNIFICANT)
9893	030622	001767			BEQ 1\$	;BRANCH IF YES
9894	030624	013737	030640	030636	MOV ESDC,EMSDP	;SAVE POSITION OF DIGIT
9895	030632	000763			BR 1\$	
9896	030634	000207		2\$:	RTS PC	
9897						
9898	030636	000000		EMSDP:	.WORD 0	
9899	030640	000000		ESDC:	.WORD 0	

9901						.SBTTL	ASHP,ASHN INSTRUCTIONS
9902	030642	012737	177777	026322	EASHP:	MOV #177777,EPAK	:INDICATE PACKED MODE
9903	030650	005037	032136			CLR EODDS	
9904	030654	005037	032140			CLR EODDD	
9905	030660	032737	000001	017364		BIT #1,E0	:IS SRC ODD LENGTH?
9906	030666	001403				BEQ 1\$	
9907	030670	012737	177777	032136		MOV #177777,EODDS	:YES - SET ODD INDICATOR
9908	030676	032737	000001	017370	1\$:	BIT #1,E2	:IS DST ODD IN LENGTH?
9909	030704	001406				BEQ EASH	
9910	030706	012737	177777	032140		MOV #177777,EODDD	:YES-SET ODD INDICATOR
9911	030714	000402				BR EASH	
9912	030716	005037	026322		EASHN:	CLR EPAK	:INDICATE ZONE MODE
9913	030722	005037	032144		EASH:	CLR ENCC	:INIT. N CLEAR INDICATOR
9914	030726	005037	032146			CLR ENZI	:INIT. NON ZERO INDICATOR
9915	030732	005037	032150			CLR ETNZI	
9916	030736	005037	025066			CLR ECARRY	:INIT. CARRY INDICATOR
9917	030742	013703	017364			MOV E0,R3	:INIT. SCR.PTR TO SRC.LEN-1
9918	030746	105303				DECB R3	
9919	030750	005737	026322			TST EPAK	:ZONED STRING?
9920	030754	001003				BNE 2\$	:BRANCH IF NO
9921	030756	012704	040237			MOV #40237,R4	:MAKE RESULT DATA TYPE = TRAILING SEPARATE
9922	030762	000406				BR 3\$	
9923	030764	013704	017370		2\$:	MOV E2,R4	:INITIALIZE RESULT PTR TO 127+31+1(MAX SHIFT CT+)
9924	030770	042704	000377			BIC #377,R4	
9925	030774	052704	000237			BIS #237,R4	
9926	031000	010437	033076		3\$:	MOV R4,ETLSD	
9927	031004	105304				DECB R4	
9928	031006	113705	017374			MOVB E4,R5	:INITIALIZE SHIFT # TO SHIFT COUNT
9929	031012	100424				BMI EASHR	:SHIFT COUNT?
9930	031014	005705			1\$:	TST R5	:POSITIVE OR ZERO - SHIFT IN DIRECTION OF : LEAST TO MOST SIGNIFICANT DIGITS
9931							: IS SHIFT # <=0?
9932	031016	003476				BLE ESISRC	:SHIFT #>0; IS RESULT.PTR=0?
9933	031020	122704	000377			CMPB #377,R4	:IF RESULT.PTR<0 BRANCH TO COPY RESULT INTO DST.
9934	031024	001555				BEQ EDETSN	:RESULT.PTR DOES NOT=0. PUSH A ZERO
9935	031026	012700	026056			MOV #EVRTAB,R0	: DIGIT INTO THE DESTINATION.
9936	031032	010401				MOV R4,R1	
9937	031034	013737	033076	025130		MOV ETLSD,ELSD	
9938	031042	005002				CLR R2	:SET R0=RESULT.ADR, R1=RESULT.PTR, R2=0
9939	031044	012737	177777	025112		MOV #177777,EODD	
9940	031052	004737	022440			JSR PC,EPUSH	:CALL ROUTINE TO PUSH THE 0 DIGIT INTO RESULT.
9941	031056	105304				DECB R4	
9942	031060	005305				DEC R5	
9943	031062	000754				BR 1\$	
9944							
9945	031064				EASHR:		:SHIFT COUNT IS NEGATIVE
9946	031064	042703	177400			BIC #177400,R3	
9947	031070	113700	017374			MOVB E4,R0	:GET SHIFT COUNT
9948	031074	060003				ADD R0,R3	:ADD SHIFT COUNT (NEGATIVE) TO SRC.PTR
9949	031076	042703	177400			BIC #177400,R3	
9950	031102	013700	017364			MOV E0,R0	
9951	031106	042700	000377			BIC #377,R0	
9952	031112	060003				ADD R0,R3	
9953	031114	122703	000177			CMPB #177,R3	:CHECK FOR SPECIAL CASE WHEN SRC.LEN=0 AND : SHIFT CNT=-128(200).
9954							

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22 PAGE 85-14 SEQUENCE 125  
ASHP,ASHN INSTRUCTIONS

9955	031120	001515			BEQ EFILZ	;BRANCH IF THIS IS THE SPECIAL CASE.
9956	031122	105703			TSTB R3	
9957	031124	100003			BPL 2\$	;BRANCH IF SRC PTR IS POSITIVE
9958	031126	122703	177777		CMPB #-1,R3	
9959	031132	001110			BNE EFILZ	;IF SRC PTR IS <-1 FILL DST WITH ZERO
9960	031134	013700	017366		MOV E1,R0	;FIND MOST SIGNIFICANT DIGIT TO BE
9961	031140	010301			MOV R3,R1	; SHIFTED OUT
9962	031142	013737	017364	025130	MOV E0,ELSD	
9963	031150	105201			INCB R1	;SET R0=SRC.ADR,R1=SRC.PTR+1
9964	031152	013737	032136	025112	MOV EODDS,EODD	
9965	031160	004737	021372		JSR PC,ESNK	;CALL ROUTINE TO FIND DIGIT
9966	031164	113700	017375		MOVB E4+1,R0	
9967	031170	060002			ADD R0,R2	;ADD RND.DGT TO DIGIT FOUND IN R2
9968	031172	022702	000012		CMP #12,R2	;IS RESULT LESS THAN 10
9969	031176	101004			BHI 1\$	
9970	031200	012737	000001	025066	MOV #1,ECARRY	;NO-SET CARRY
9971	031206	000402			BR ESISRC	
9972	031210	005037	025066		CLR ECARRY	
9973						
9974	031214				ESISRC:	;SHIFT SRC DIGITS INTO DST.
9975	031214	122704	000377		CMPB #377,R4	;IS RESULT.PTR <0?
9976	031220	001457			BEQ EDETSN	;YES - BRANCH TO COPY RESULT INTO DEST
9977	031222	105703			TSTB R3	;NO - IS SRC.PTR <0?
9978	031224	100453			BMI EFILZ	
9979	031226	013700	017366		MOV E1,R0	;NO - FIND SRC DIGIT TO SHIFT INTO RESULT
9980	031232	010301			MOV R3,R1	;SET R0=SRC.ADR, R1=SRC.PTR
9981	031234	013737	017364	025130	MOV E0,ELSD	
9982	031242	013737	032136	025112	MOV EODDS,EODD	
9983	031250	004737	021372		JSR PC,ESNK	;CALL ROUTINE TO FIND SRC DIGIT
9984	031254	063702	025066		ADD ECARRY,R2	;ADD CARRY TO DIGIT FOUND
9985	031260	022702	000011		CMP #11,R2	;DIGIT OVERFLOW?
9986	031264	103005			BHIS 2\$	;IF NO BRANCH
9987	031266	005002			CLR R2	;OVERFLOW - SET CARRY & SET DIGIT=0
9988	031270	012737	000001	025066	MOV #1,ECARRY	
9989	031276	000402			BR 3\$	
9990	031300	005037	025066		CLR ECARRY	
9991	031304	012700	026056		MOV #EVRTAB,R0	;PUSH DIGIT FOUND INTO RESULT.
9992	031310	010401			MOV R4,R1	;SET R0=RESULT.ADR, R1=RESULT.PTR, R2 CONTAINS D
9993	031312	013737	033076	025130	MOV ETLSD,ELSD	
9994	031320	012737	177777	025112	MOV #177777,EODD	
9995	031326	004737	022440		JSR PC,EPUSH	;CALL ROUTINE TO PUSH DIGIT INTO RESULT
9996	031332	042702	177760		BIC #177760,R2	;MASK OFF ALL BUT DIGIT PUSHED
9997	031336	001403			BEQ 1\$	;IS DIGIT PUSHED=0?
9998	031340	012737	177777	032150	MOV #177777,ETNZI	;NO-SET NON ZERO INDICATOR
9999	031346	105303			DEC B R3	;DECREMENT SRC.PTR
10000	031350	105304			DEC B R4	;DECREMENT RESULT.PTR
10001	031352	000720			BR ESISRC	
10002						
10003	031354	000137	032152		EFILZ:	
10004	031360				EDETSN:	
10005					JMP EFILLZ	;DETERMINE SIGN & STORE WITH RESULT
10006						;NOTE: THERE EXIST TWO CASES IN WHICH THE DST SIGN
10007						; WILL DIFFER FROM THE SRC SIGN. THESE CASES ARE
10008						; WHEN:
						; 1)SRC SIGN = -,SHIFT=RIGHT,AND RESULT(ETNZI)=0

Address	Source 1	Source 2	Source 3	Source 4	Instruction	Comment
10009						; 2)SRC SIGN = -,SHIFT=LEFT,AND SRC MAGNITUDE=0
10010	031360	013700	017366		MOV E1,R0	;SETUP SRC ADR
10011	031364	013701	017364		MOV E0,R1	;SETUP PTR TO SIGN
10012	031370	013737	017364	025130	MOV E0,ELSD	;SETUP PTR TO LEAST SIGN DIGIT
10013	031376	013737	032136	025112	MOV EODDS,EODD	
10014	031404	004737	021372		JSR PC,ESNK	;CALL ROUTINE TO FIND SRC SIGN
10015						;SIGN RETURNED IN ERSNEG
10016	031410	005737	032150		TST ETNZI	;NON ZERO INDICATOR SET?
10017	031414	001035			BNE 1\$	;BRANCH IF YES
10018	031416	032737	000200	017374	BIT #200,E4	;SHIFT RIGHT?
10019	031424	001403			BEQ 2\$	;NO
10020	031426	005037	025062		CLR ERSNEG	;MAKE SIGN POSITIVE (CASE 1 NOTED ABOVE)
10021	031432	000426			BR 1\$	
10022	031434	105737	017364	2\$:	TSTB E0	;SRC MAGNITUDE = 0
10023	031440	001421			BEQ 4\$	;BRANCH IF YES (SRC.LEN = 0)
10024	031442	013700	017366		MOV E1,R0	;SETUP SRC.ADR
10025	031446	013701	017364		MOV E0,R1	;SETUP PTR TO SIGN
10026	031452	013737	017364	025130	MOV E0,ELSD	;SETUP PTR TO LEAST SIGN DIGIT
10027	031460	105301		3\$:	DECB R1	
10028	031462	013737	032136	025112	MOV EODDS,EODD	;FIND SRC DIGIT
10029	031470	004737	021372		JSR PC,ESNK	
10030	031474	005702			TST R2	;IS DIGIT = 0?
10031	031476	001004			BNE 1\$	;BRANCH IF NO
10032	031500	105701			TSTB R1	;ALL DIGITS IN SRC TESTED?
10033	031502	001366			BNE 3\$	;BRANCH IF NO
10034	031504	005037	025062	4\$:	CLR ERSNEG	;CASE 2 NOTED ABOVE
10035	031510			1\$:		;STORE SIGN WITH RESULT
10036	031510	012700	026056		MOV #EVRTAB,R0	
10037	031514	013701	033076		MOV ETLSD,R1	
10038	031520	013737	033076	025130	MOV ETLSD,ELSD	
10039	031526	012737	177777	025112	MOV #177777,EODD	
10040	031534	004737	022440		JSR PC,EPUSH	
10041						
10042	031540			ECRID:		;COPY RESULT INTO DESTINATION
10043	031540	005037	032146		CLR ENZI	;CLEAR NON ZERO INDICATOR
10044	031544	113701	017370		MOVB E2,R1	; OF RESULT TO COPY.
10045	031550	005737	026322		TST EPAK	;PACKED INST?
10046	031554	001405			BEQ 1\$	;BRANCH IF NO
10047	031556	012704	026176		MOV #EVRTAB+120,R4	
10048	031562	006201			ASR R1	;CALCULATE # OF BYTES OCCUPIED BY DST
10049	031564	005201			INC R1	; # OF BYTES=# OF DIGITS/2 +1
10050	031566	000402			BR 4\$	
10051	031570	012704	026315	1\$:	MOV #EVRTAB+237,R4	
10052	031574	160104		4\$:	SUB R1,R4	
10053	031576	013701	017370		MOV E2,R1	;SETUP POSITION POINTER TO ZERO
10054	031602	042701	000377		BIC #377,R1	
10055	031606	010103			MOV R1,R3	;SAVE DATA TYPE FIELD
10056	031610	010400		2\$:	MOV R4,R0	;SETUP 'FROM' ADDRESS
10057	031612	013737	017370	025130	MOV E2,ELSD	;SETUP POINTER TO LEAST SIGN DIGIT
10058	031620	013737	032140	025112	MOV EODDD,EODD	
10059	031626	005737	026322		TST EPAK	;ZONED RESULT?
10060	031632	001007			BNE 5\$	;BRANCH IN NO
10061	031634	042701	177400		BIC #177400,R1	;ZONED RESULT DATA TYPE TRAILING SEPARATE
10062	031640	052701	040000		BIS #40000,R1	

10063	031644	112737	000100	025131		MOV B #100,ELSD+1	
10064	031652	004737	021372		5\$:	JSR PC,ESNK	;CALL ROUTINE TO GET RESULT DIGIT
10065	031656	013700	017372			MOV E3,R0	;SETUP 'TO' ADDRESS (I.E. DST)
10066	031662	105037	025131			CLRB ELSD+1	
10067	031666	050337	025130			BIS R3,ELSD	
10068	031672	042701	177400			BIC #177400,R1	
10069	031676	050301				BIS R3,R1	
10070	031700	004737	022440			JSR PC,EPUSH	;CALL ROUTINE TO PUSH RESULT DIGIT INTO DST
10071	031704	005702				TST R2	;WAS DIGIT PUSHED NON ZERO?
10072	031706	001403				BEQ 3\$	;BRANCH IF NO
10073	031710	012737	177777	032146		MOV #177777,ENZI	;SET NON ZERO INDICATOR
10074	031716	020137	017370		3\$:	CMP R1,E2	;COPY DONE?
10075	031722	001402				BEQ ESCC	;BRANCH IF YES
10076	031724	005201				INC R1	;UPDATE PTR AND RETURN TO COPY NXT DIGIT
10077	031726	000730				BR 2\$	
10078							
10079	031730				ESCC:		;SET CONDITION CODES
10080	031730	005077	165426			CLR @EOPSW	;RESET EMULATION PSW
10081	031734	005737	032146			TST ENZI	;SET Z BIT IF NON ZERO INDICATOR=0.
10082	031740	001004				BNE 1\$	
10083	031742	052777	000004	165412		BIS #4,@EOPSW	
10084	031750	000406				BR 2\$	
10085	031752	005737	025062		1\$:	TST ERSNEG	;IS SIGN NEGATIVE?
10086	031756	001403				BEQ 2\$	;BRANCH IF NO
10087	031760	052777	000010	165374		BIS #10,@EOPSW	;SET N BIT
10088	031766				2\$:		;DETERMINE V BIT
10089	031766	113700	017370			MOV B E2,R0	;CALCULATE # OF DIGITS OF RESULT
10090	031772	012704	000237			MOV #237,R4	; THAT WOULD NOT FIT IN DEST.
10091	031776	160004				SUB R0,R4	
10092	032000	001437				BEQ 6\$	
10093	032002	005304				DEC R4	
10094	032004	012700	026056			MOV #EVRTAB,R0	;YES - WERE ANY SIGNIFICANT DIGITS
10095	032010	013701	017370			MOV E2,R1	; NOT STORED?
10096	032014	042701	000377			BIC #377,R1	
10097	032020	005737	026322			TST EPAK	;ZONED RESULT?
10098	032024	001004				BNE 3\$	;BRANCH IF NO
10099	032026	042701	177400			BIC #177400,R1	;ZONED RESULT DATA TYPE =TRAILING SEPARATE
10100	032032	052701	040000			BIS #40000,R1	
10101	032036	012737	177777	025112	3\$:	MOV #177777,EODD	
10102	032044	013737	033076	025130		MOV ETLSD,ELSD	
10103	032052	004737	021372		5\$:	JSR PC,ESNK	; CALL ROUTINE TO FIND RESULT DIGIT
10104	032056	005702				TST R2	; NOT STORED.
10105	032060	001004				BNE 4\$	;GO SET OVERFLOW - V BIT
10106	032062	120104				CMPB R1,R4	
10107	032064	001405				BEQ 6\$	
10108	032066	005201				INC R1	
10109	032070	000770				BR 5\$	
10110	032072	052777	000002	165262	4\$:	BIS #2,@EOPSW	;SET V BIT
10111	032100	013702	017360		6\$:	MOV EORSTK,R2	;REGISTER UNLOAD
10112	032104	005022				CLR (R2)+	;R0=0
10113	032106	005022				CLR (R2)+	;R1=0
10114	032110	013722	017370			MOV E2,(R2)+	;R2=R2
10115	032114	013722	017372			MOV E3,(R2)+	;R3=R3
10116	032120	005022				CLR (R2)+	;R4=0

10117	032122	013722	017376		MOV E5,(R2)+		:R5=R5
10118	032126	000207			RTS PC		
10119							
10120	032130	000000			ETMPRO:	.WORD 0	
10121	032132	000000			ETMPR1:	.WORD 0	
10122	032134	000000			ETMPR2:	.WORD 0	
10123	032136	000000			EODDS:	.WORD 0	
10124	032140	000000			EODDD:	.WORD 0	
10125	032142	000000			ESGN:	.WORD 0	
10126	032144	000000			ENCC:	.WORD 0	
10127	032146	000000			ENZI:	.WORD 0	
10128	032150	000000			ETNZI:	.WORD 0	
10129							
10130	032152				EFILLZ:		:PAD REMAINING RESULT WITH ZERO DIGITS
10131	032152	122704	000377		CMPB #377,R4		:IS RESULT.PTR <0?
10132	032156	001426			BEQ 1\$		:IF YES BRANCH
10133	032160	012700	026056		MOV #EVRTAB,R0		:PUSH A ZERO DIGIT INTO RESULT.
10134	032164	010401			MOV R4,R1		:SET R0=RESULT.ADR, R1=RESULT.PTR, R2=0
10135	032166	013702	025066		MOV ECARRY,R2		
10136	032172	013737	033076	025130	MOV ETLSD,ELSD		
10137	032200	012737	177777	U25112	MOV #177777,EODD		
10138	032206	004737	022440		JSR PC,EPUSH		:CALL ROUTINE TO PUSH ZERO + CARRY
10139							: DIGIT INTO RESULT.
10140	032212	005702			TST R2		:WAS DIGIT PUSHED = 0?
10141	032214	001403			BEQ 2\$		:BRANCH IF YES
10142	032216	012737	177777	032150	MOV #177777,ETNZI		:DIGIT PUSHED NOT = 0;SET NONZERO INDICATOR
10143	032224	105304			2\$: DEC B R4		
10144	032226	005037	025066		CLR ECARRY		
10145	032232	000747			BR EFILLZ		
10146	032234	000137	031360		1\$: JMP EDETSN		

10148						.SBTTL	CMPP,CMPN INSTRUCTIONS
10149	032240	012737	177777	026322	ECMPP:	MOV #177777,EPAK	:INDICATE PACKED MODE
10150	032246	005037	033066			CLR EODDS1	
10151	032252	005037	033070			CLR EODDS2	
10152	032256	032737	000001	017364		BIT #1,E0	:IS SRC NUMBER ODD LENGTH?
10153	032264	001403				BEQ 1\$	
10154	032266	012737	177777	033066		MOV #177777,EODDS1	:YES - SET ODD INDICATOR
10155	032274	032737	000001	017370	1\$:	BIT #1,E2	:IS SRC2 ODD LENGTH?
10156	032302	001406				BEQ ECMP	
10157	032304	012737	177777	033070		MOV #177777,EODDS2	:YES SET ODD INDICATOR
10158	032312	000402				BR ECMP	
10159	032314	005037	026322		ECMPN:	CLR FPAK	:INDICATE ZONED MODE
10160							
10161	032320	013703	017364		ECMP:	MOV E0,R3	:INITIALIZE SRC.PTY TO 0
10162	032324	042703	000377			BIC #377,R3	
10163	032330	013704	017370			MOV E2,R4	:INITIALIZE DST.PTR TO 0
10164	032334	042704	000377			BIC #377,R4	
10165	032340	010337	033072			MOV R3,ELS1M	
10166	032344	010437	033074			MOV R4,ELS2M	
10167	032350	013700	017366		1\$:	MOV E1,R0	:FIND MOST SIGNIFICANT DIGIT IN SRC1
10168	032354	010301				MOV R3,R1	:SET R0=SRC1.ADR, R1=SRC1.PTR
10169	032356	120337	017364			CMPB R3,E0	:REACH END OF SRC1 STRING?
10170	032362	001416				BEQ 2\$	:IF YES BRANCH
10171	032364	013737	017364	025130		MOV E0,ELSD	
10172	032372	013737	033066	025112		MOV EODDS1,EODD	
10173	032400	004737	021372			JSR PC,ESNK	:CALL ROUTINE TO FIND SRC DIGIT
10174	032404	005702				TST R2	:IS SRC1 DIGIT=0?
10175	032406	001004				BNE 2\$	:IF NO BRANCH
10176	032410	005203				INC R3	:UPDATE SRC1.PTR TO NEXT DIGIT
10177	032412	010337	033072			MOV R3,ELS1M	:SAVE SRC1.PTR
10178	032416	000754				BR 1\$	
10179	032420	013705	017364		2\$:	MOV E0,R5	
10180	032424	160305				SUB R3,R5	:CALCULATE # OF SIGN DIGITS IN SRC1
10181	032426	010537	033062			MOV R5,ES1NSD	:SAVE # OF SIGN DIGITS IN ES1NSD
10182	032432	013700	017372		3\$:	MOV E3,R0	:FIND MOST SIGNIFICANT DIGIT IN SRC2
10183	032436	010401				MOV R4,R1	:SET R0=SRC2.ADR, R1=SRC2.PTR
10184	032440	120437	017370			CMPB R4,E2	:REACH END OF SRC2 STRING?
10185	032444	001416				BEQ 4\$	:IF YES BRANCH
10186	032446	013737	017370	025130		MOV E2,ELSD	
10187	032454	013737	033070	025112		MOV EODDS2,EODD	
10188	032462	004737	021372			JSR PC,ESNK	:CALL ROUTINE TO FIND SRC DIGIT
10189	032466	005702				TST R2	:IS SRC2 DIGIT=0
10190	032470	001004				BNE 4\$	:IF NO BRANCH
10191	032472	005204				INC R4	:UPDATE SRC2.PTR TO NEXT DIGIT
10192	032474	010437	033074			MOV R4,ELS2M	:SAVE SRC2.PTR
10193	032500	000754				BR 3\$	
10194	032502	013705	017370		4\$:	MOV E2,R5	
10195	032506	160405				SUB R4,R5	:CALCULATE # OF SIGN DIGITS IN SRC2
10196	032510	010537	033064			MOV R5,ES2NSD	:SAVE # OF SIGN DIGITS IN ES2NSD
10197							
10198	032514				ECNSD:		:COMPARE # OF SIGN DIGITS IN SRC1
10199	032514	005077	164642			CLR @EOPSW	:VERSUS SRC2
10200	032520	023737	033062	033064		CMP ES1NSD,ES2NSD	
10201	032526	101052				BHI 5\$	:BRANCH IF SRC1 HAS MORE SIGN DIGITS



10202	032530	103457			BLO 6\$		;BRANCH IF SRC2 HAS MORE SIGN DIGITS
10203	032532	005737	033062		TST ES1NSD		;SRC1 & SRC2 CONTAIN THE SAME #
10204							; OF SIGNIFICANT DIGITS
10205	032536	001500			BEQ 3\$		;BOTH SRC'S CONTAIN NO SIGNIFICANT DIGITS
10206	032540	013703	033072		MOV ELS1M,R3		;SETUP SRC1.PTR
10207	032544	013704	033074		MOV ELS2M,R4		;SETUP SRC2.PTR
10208	032550	013700	017366		4\$: MOV E1,R0		;GET A SRC1 DIGIT
10209	032554	010301			MOV R3,R1		
10210	032556	013737	017364	025130	MOV E0,ELSD		
10211	032564	013737	033066	025112	MOV EODDS1,EODD		
10212	032572	004737	021372		JSR PC,ESNK		
10213	032576	010237	033060		MOV R2,ES1D		;SAVE SRC1 DIGIT
10214	032602	013700	017372		MOV E3,R0		;GET A SRC2 DIGIT
10215	032606	010401			MOV R4,R1		
10216	032610	013737	017370	025130	MOV E2,ELSD		
10217	032616	013737	033070	025112	MOV EODDS2,EODD		
10218	032624	004737	021372		JSR PC,ESNK		
10219	032630	023702	033060		CMP ES1D,R2		;COMPARE DIGITS
10220	032634	101007			BHI 5\$		;BRANCH IF SRC1 DIGIT IS BIGGER
10221	032636	103414			BLO 6\$		;BRANCH IF SRC2 DIGIT IS BIGGER
10222	032640	005203			INC R3		;DIGITS EQUAL - ALL DIGITS CHECKED?
10223	032642	120337	033062		CMPB R3,ES1NSD		
10224	032646	001416			BEQ 7\$		;BRANCH IF ALL CHECKED - I.E. ALL
10225							; DIGITS ARE EQUAL
10226	032650	005204			INC R4		
10227	032652	000736			BR 4\$		
10228	032654	004737	032774		5\$: JSR PC,EGS1S		;CHECK SIGN OF SRC1
10229	032660	005737	025062		TST ERSNEG		;IS SRC1 NEG?
10230	032664	001021			BNE 2\$		;BRANCH IF YES
10231	032666	000427			BR 1\$		;SRC1 IS POSITIVE
10232							
10233	032670	004737	033026		6\$: JSR PC,EGS2S		;CHECK SIGN OF SRC2
10234	032674	005737	025062		TST ERSNEG		;IS SRC2 NEGATIVE?
10235	032700	001022			BNE 1\$		;BRANCH IF YES
10236	032702	000412			BR 2\$		;SRC2 IS POSITIVE
10237							
10238	032704				7\$: JSR PC,EGS1S		;COMPARE SIGNS
10239	032704	004737	032774		MOV ERSNEG,R5		;GET SRC1 SIGN
10240	032710	013705	025062		JSR PC,EGS2S		;SAVE IT IN R5
10241	032714	004737	033026		CMP ERSNEG,R5		;GET SRC2 SIGN
10242	032720	023705	025062		BEQ 3\$		;SIGNS =?
10243	032724	001405			BR 5\$		;BRANCH IF YES
10244	032726	000752					;SIGNS NOT EQUAL
10245							
10246	032730	052777	000010	164424	2\$: BIS #10,@EOPSW		;SRC2>SRC1 SET N BIT.
10247	032736	000403			BR 1\$		
10248	032740	052777	000004	164414	3\$: BIS #4,@EOPSW		;SRC2=SRC1 SET Z BIT
10249	032746	013702	017360		1\$: MOV EORSTK,R2		;REGISTER UNLOAD
10250	032752	005022			CLR (R2)+		;R0=0
10251	032754	005022			CLR (R2)+		;R1=0
10252	032756	005022			CLR (R2)+		;R2=0
10253	032760	005022			CLR (R2)+		;R3=0
10254	032762	013722	017374		MOV F4,(R2)+		;R4=R4
10255	032766	013722	017376		MOV E5,(R2)+		;R5=R5

10256	032772	000207			RTS PC	
10257						
10258	032774	013700	017366		EGS1S:	MOV E1,R0 ;ROUTINE TO GET SRC1 SIGN
10259	033000	013701	017364			MOV E0,R1 ;SET R0=SRC1.ADR,R1=SRC1.PTR
10260	033004	013737	017364	025130		MOV E0,ELSD
10261	033012	013737	033066	025112		MOV EODDS1,EODD
10262	033020	004737	021372			JSR PC,ESNK ;CALL ROUTINE TO FIND SRC1 SIGN
10263	033024	000207				RTS PC
10264						
10265	033026	013700	017372		EGS2S:	MOV E3,R0 ;ROUTINE TO GET SRC2 SIGN
10266	033032	013701	017370			MOV E2,R1 ;SET R0=SRC2.ADR,R1=SRC2.PTR
10267	033036	013737	017370	025130		MOV E2,ELSD
10268	033044	013737	033070	025112		MOV EODDS2,EODD
10269	033052	004737	021372			JSR PC,ESNK ;CALL ROUTINE TO FIND SRC2 SIGN
10270	033056	000207				RTS PC
10271						
10272	033060	000000			ES1D:	.WORD 0
10273	033062	000000			ES1NSD:	.WORD 0
10274	033064	000000			ES2NSD:	.WORD 0
10275	033066	000000			EODDS1:	.WORD 0
10276	033070	000000			EODDS2:	.WORD 0
10277	033072	000000			ELS1M:	.WORD 0
10278	033074	000000			ELS2M:	.WORD 0
10279	033076	000000			ETLSD:	.WORD 0

```

10281
10282
10283 033100 004737 034142
10284 033104 005037 035130
10285 033110 004737 034224
10286 033114 004737 034300
10287 033120 032737 000004 034610
10288 033126 001147
10289 033130 004737 034350
10290 033134 032737 000004 034610
10291 033142 001141
10292
10293 033144 005037 017370
10294 033150 012737 034552 017372
10295 033156 012737 035060 035126
10296 033164 012737 035102 035124
10297 033172 013737 034574 017364
10298 033200 013737 034576 017366
10299 033206 012737 000040 017374
10300 033214 017737 001706 017376
10301 033222 012737 034554 017360
10302 033230 012737 034610 017362
10303 033236 004737 023402
10304 033242 042737 177775 034610
10305 033250 013777 034610 001646
10306 033256 027727 001644 034612
10307 033264 001041
10308 033266 032737 000001 034600
10309 033274 001404
10310 033276 012737 177777 025112
10311 033304 000402
10312 033306 005037 025112
10313 033312 013700 034602
10314 033316 013701 034600
10315 033322 013737 034600 025130
10316 033330 004737 021372
10317 033334 005737 025062
10318 033340 001413
10319 033342 132737 000001 034632
10320 033350 001404
10321 033352 142737 000001 034632
10322 033360 000403
10323 033362 152737 000001 034632
10324 033370 027727 001532 035032
10325 033376 001424
10326 033400 012737 000040 017364
10327 033406 012737 034612 017366
10328 033414 012737 000040 017370
10329 033422 017737 001500 017372
10330 033430 062737 000002 035126
10331 033436 062737 000002 035124
10332 033444 000660
10333
10334 033446 000565

```

.SBTTL

MULP INSTRUCTION

EMULP:

```

JSR PC,ERSAV
CLR EMVBR
JSR PC,EIRT2
JSR PC,ETSTS1
BIT #4,ETOPSW
BNE EPMID
JSR PC,ETSTS2
BIT #4,ETOPSW
BNE EPMID

```

```

;SAVE MULP CALL PARAMETERS
;INITIALIZE MULP V-BIT RESULT
;INITIALIZE TEMPORARY RESULT BUFFER (ERT2) TO 0+
;IS MULP SRC1=0?

```

;IS MULP SRC2=0?

```

CLR E2
MOV #EZDSC,E3
MOV #EXTBP,EVXTBP
MOV #EXTVB,EVXTVB
MOV ESR0,E0
MOV ESR1,E1
MOV #40,E4
MOV @EVXTBP,E5
MOV #ETRSTK,EORSTK
MOV #ETOPSW,EOPSW
JSR PC,EADDP

```

```

;FORM 1X,2X,3X, ETC TABLE
; USE ADDP - SRC1=MULT.SRC1
; SRC2=PREVIOUS ADDP DST
; DST=E(N)XT TABLE

```

1\$:

```

BIC #177775,ETOPSW
MOV ETOPSW,@EVXTVB
CMP @EVXTBP,#E1XT
BNE 2$
BIT #1,ESR2
BEQ 31$
MOV #177777,EODD
BR 3$

```

```

;CLEAR ALL BUT V BIT FROM ADDP
; RESULT PSW; SAVE V BITS IN TABLE.
;FIRST TABLE ENTRY FORMATION?
;BRANCH IF NO
;WORK ON TABLE ENTRY SIGNS

```

31\$:

3\$:

```

CLR EODD
MOV ESR3,R0
MOV ESR2,R1
MOV ESR2,ELSD
JSR PC,ESNK
TST ERSNEG
BEQ 2$
BITB #1,E1XT+20
BEQ 4$
BICB #1,E1XT+20
BR 2$

```

```

;CALL ROUTINE TO GET MULP.SRC2 SIGN
;IF THIS SIGN IS POSITIVE THEN
; LEAVE TABLE SIGNS = MULP.SRC1 SIGN
;MULP.SRC2 SIGN = NEG; MAKE TABLE
; SIGNS = COMPLEMENT OF MULP.SRC1 SIGN

```

4\$:

2\$:

```

BISB #1,E1XT+20
CMP @EVXTBP,#E9XT
BEQ EISP
MOV #40,E0
MOV #E1XT,E1
MOV #40,E2
MOV @EVXTBP,E3
ADD #2,EVXTBP
ADD #2,EVXTVB
BR 1$

```

```

;ALL TABLE ENTRIES FORMED?
;BRANCH IF YES

```

```

;UPDATE ADDP SRC2 TO CURRENT
; DST POINTER
;UPDATE TABLE POINTERS

```

;RETURN TO FORM NEXT ENTRY.

EPMID:

BR EMID

10335									
10336	033450	005037	035054		EISP:	CLR ESPOS		:INITIALIZE SHIFT POSITION TO ZERO	
10337	033454	032737	000001	034600		BIT #1,ESR2			
10338	033462	001404				BEQ 10\$			
10339	033464	012737	177777	025112		MOV #177777,EODD			
10340	033472	000402				BR 11\$			
10341	033474	005037	025112			10\$: CLR EODD			
10342	033500	013737	025112	035132		11\$: MOV EODD,SEODD		:SAVE EODD	
10343	033506	013737	034600	035056		MOV ESR2,EMS2D		:INITIALIZE MULP.SRC2 POINTER	
10344	033514	013737	035132	025112		1\$: MOV SEODD,EODD		:RESTORE EODD	
10345	033522	105337	035056			DECB EMS2D			
10346	033526	100535				BMI EMID		:BRANCH IF NO MORE MULP.SRC2 DIGITS TO WORK ON	
10347	033530	013700	034602			MOV ESR3,R0			
10348	033534	013737	034600	025130		MOV ESR2,ELSD			
10349	033542	013701	035056			MOV EMS2D,R1			
10350	033546	004737	021372			JSR PC,ESNK		:CALL ROUTINE TO GET NEXT SRC2 DIGIT	
10351	033552	005702				TST R2			
10352	033554	001517				BEQ 2\$		:BRANCH IF DIGIT = 0	
10353	033556	005302				DEC R2			
10354	033560	010237	035124			MOV R2,EVXTVB			
10355	033564	006337	035124			ASL EVXTVB			
10356	033570	062737	035102	035124		ADD #EXTVB,EVXTVB		:SETUP POINTER INTO V-BIT TABLE	
10357	033576	010237	035126			MOV R2,EVXTBP		:INDEX INTO 1X,2X,ETC TABLE USING	
10358	033602	006337	035126			ASL EVXTBP		: NEXT SRC2 DIGIT	
10359	033606	062737	035060	035126		ADD #EXTBP,EVXTBP			
10360	033614	012737	000040	017364		MOV #40,E0		:MULTIPLY TABLE VALUE BY PROPER	
10361	033622	017737	001300	017366		MOV @EVXTBP,E1		: POWER OF TEN INDICATOR BY SHIFT	
10362	033630	013737	035054	017374		MOV ESPOS,E4		: POSITION	
10363	033636	012737	000037	017370		MOV #37,E2		: USE ASHP - SRC=TABLE VALUE	
10364	033644	012737	034420	017372		MOV #ERT1,E3		: SHFT.CT=SHFT.POSITION	
10365	033652	012737	034554	017360		MOV #ETRSTK,EORSTK		: DST=ERT1	
10366	033660	012737	034610	017362		MOV #ETOPSW,EOPSW			
10367	033666	004737	030642			JSR PC,EASHP			
10368	033672	042737	177775	034610		BIC #177775,ETOPSW		:WORK ON V BIT	
10369	033700	053737	034610	035130		BIS ETOPSW,EMVBR		: 'OR' ASHP V BIT WITH RESULT V BIT.	
10370	033706	057737	001212	035130		BIS @EVXTVB,EMVBR		: 'OR' TABLE V BIT WITH RESULT V BIT	
10371	033714	012737	000037	017364		MOV #37,E0		:ADD SHIFTED VALUE TO RESULT	
10372	033722	012737	034442	017366		MOV #ERT2,E1		: USE ADDP - SRC1=ERT2	
10373	033730	012737	000037	017370		MOV #37,E2		: SRC2=ERT1	
10374	033736	012737	034420	017372		MOV #ERT1,E3		: DST=ERT2	
10375	033744	012737	000037	017374		MOV #37,E4			
10376	033752	012737	034442	017376		MOV #ERT2,E5			
10377	033760	012737	034554	017360		MOV #ETRSTK,EORSTK			
10378	033766	012737	034610	017362		MOV #ETOPSW,EOPSW			
10379	033774	004737	023402			JSR PC,EADDP			
10380	034000	042737	177775	034610		BIC #177775,ETOPSW		:WORK ON V BIT	
10381	034006	053737	034610	035130		BIS ETOPSW,EMVBR		: 'OR' ADDP V BIT WITH RESULT	
10382									
10383	034014	005237	035054			2\$: INC ESPOS		:INCREMENT SHIFT POSITION FOR NEXT	
10384	034020	000635				BR 1\$		: MULP SRC2 DIGIT.	
10385									
10386	034022	012737	000037	017364	EMID:	MOV #37,E0		:MOVE RESULT INTO MULP DST	
10387	034030	012737	034442	017366		MOV #ERT2,E1		: USE ASHP - SRC=ERT2	
10388	034036	005037	017374			CLR E4		: DST=MULP.DST	

10389 034042 005037 017376  
10390 034046 013737 034604 017370  
10391 034054 013737 034606 017372  
10392 034062 013737 034570 017360  
10393 034070 013737 034572 017362  
10394 034076 004737 030642  
10395 034102 053777 035130 163252  
10396  
10397  
10398 034110 013702 017360  
10399  
10400  
10401 034114 016262 000004 000010  
10402 034122 016262 000006 000012  
10403 034130 005062 000004  
10404 034134 005062 000006  
10405 034140 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

10407						.SBTTL	MULP/DIVP SUBROUTINES
10408	034142	013737	017360	034570	ERSAV:	MOV EORSTK,ESOSTK	;SAVE MULP/DIVP CALL PARAMETERS
10409	034150	013737	017362	034572		MOV EOPSW,ESOPSW	;THIS PERMITS CALLING OTHER
10410	034156	013737	017364	034574		MOV E0,ESR0	; EMULATION ROUTINES WHILE
10411	034164	013737	017366	034576		MOV E1,ESR1	; IN THE MULP/DIVP EMULATION.
10412	034172	013737	017370	034600		MOV E2,ESR2	
10413	034200	013737	017372	034602		MOV E3,ESR3	
10414	034206	013737	017374	034604		MOV E4,ESR4	
10415	034214	013737	017376	034606		MOV E5,ESR5	
10416	034222	000207				RTS PC	
10417							
10418	034224	005037	017364		EIRT2:	CLR E0	;INITIALIZE ERT2 BUFFER TO 0+
10419	034230	012737	034552	017366		MOV #EZDSC,E1	; USE ASHP - SRC.LEN=0
10420	034236	005037	017374			CLR E4	; SHFT.CT=0
10421	034242	012737	000037	017370		MOV #37,E2	; DST.LEN=31
10422	034250	012737	034442	017372		MOV #ERT2,E3	; DST.ADR=ERT2
10423	034256	012737	034554	017360		MOV #ETRSTK,EORSTK	
10424	034264	012737	034610	017362		MOV #ETOPSW,EOPSW	
10425	034272	004737	030642			JSR PC,EASHP	
10426	034276	000207				RTS PC	
10427							
10428	034300	005037	017370		ETSTS1:	CLR E2	;COMPARE MULP/DIVP SRC1 WITH 0.
10429	034304	012737	034552	017372		MOV #EZDSC,E3	; USE CMPP - SRC2.LEN=0
10430	034312	013737	034574	017364		MOV ESR0,E0	; SRC1.LEN=MULP.SRC1.LEN
10431	034320	013737	034576	017366		MOV ESR1,E1	; SRC1.ADR=MULP.SRC1.ADR
10432	034326	012737	034554	017360		MOV #ETRSTK,EORSTK	
10433	034334	012737	034610	017362		MOV #ETOPSW,EOPSW	
10434	034342	004737	032240			JSR PC,ECMPP	
10435	034346	000207				RTS PC	
10436							
10437	034350	005037	017370		ETSTS2:	CLR E2	;COMPARE MULP/DIVP SRC2 WITH 0.
10438	034354	012737	034552	017372		MOV #EZDSC,E3	; USE CMPP - SRC2.LEN=0
10439	034362	013737	034600	017364		MOV ESR2,E0	; SRC1.LEN=MULP.SRC2.LEN
10440	034370	013737	034602	017366		MOV ESR3,E1	; SRC1.ADR=MULP.SRC2.ADR
10441	034376	012737	034554	017360		MOV #ETRSTK,EORSTK	
10442	034404	012737	034610	017362		MOV #ETOPSW,EOPSW	
10443	034412	004737	032240			JSR PC,ECMPP	
10444	034416	000207				RTS PC	
10445							

10447  
10448  
10449 034420 000011  
10450 034442 000011  
10451 034464 000011  
10452 034506 000011  
10453 034530 000011  
10454 034552 000000  
10455 034554 000006  
10456 034570 000000  
10457 034572 000000  
10458 034574 000000  
10459 034576 000000  
10460 034600 000000  
10461 034602 000000  
10462 034604 000000  
10463 034606 000000  
10464 034610 000000  
10465 034612 000011  
10466 034634 000011  
10467 034656 000011  
10468 034700 000011  
10469 034722 000011  
10470 034744 000011  
10471 034766 000011  
10472 035010 000011  
10473 035032 000011  
10474 035054 000000  
10475 035056 000000  
10476 035060 034612  
10477 035062 034634  
10478 035064 034656  
10479 035066 034700  
10480 035070 034722  
10481 035072 034744  
10482 035074 034766  
10483 035076 035010  
10484 035100 035032  
10485 035102 000011  
10486 035124 000000  
10487 035126 000000  
10488 035130 000000  
10489 035132 000000  
10490 035134 000000  
10491 035136 000000  
10492 035140 000000

.SBTTL  
ERT1: .BLKW 11  
ERT2: .BLKW 11  
ERT3: .BLKW 11  
ERT4: .BLKW 11  
ERT5: .BLKW 11  
EZDSC: .WORD 0  
ETRSTK: .BLKW 6  
ESOSTK: .WORD 0  
ESOPSW: .WORD 0  
ESR0: .WORD 0  
ESR1: .WORD 0  
ESR2: .WORD 0  
ESR3: .WORD 0  
ESR4: .WORD 0  
ESR5: .WORD 0  
ETOPSW: .WORD 0  
E1XT: .BLKW 11  
E2XT: .BLKW 11  
E3XT: .BLKW 11  
E4XT: .BLKW 11  
E5XT: .BLKW 11  
E6XT: .BLKW 11  
E7XT: .BLKW 11  
E8XT: .BLKW 11  
E9XT: .BLKW 11  
ESPOS: .WORD 0  
EMS2D: .WORD 0  
EXTBP: .WORD E1XT  
.WORD E2XT  
.WORD E3XT  
.WORD E4XT  
.WORD E5XT  
.WORD E6XT  
.WORD E7XT  
.WORD E8XT  
.WORD E9XT  
EXTVB: .BLKW 11  
EVXTVB: .WORD 0  
EVXTBP: .WORD 0  
EMVBR: .WORD 0  
SEODD: .WORD 0  
ESS2SN: .WORD 0  
ESS1SN: .WORD 0  
ESUBCT: .WORD 0

MULP/DIVP VARIABLES AND BUFFERS

: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

Address	Instruction	Comment	Instruction	Comment
10494			.SBTTL	DIVP INSTRUCTION
10495				
10496	035142 004737 034142	EDIVP:	JSR PC,ERSAV	;SAVE DIVP CALL PARAMETERS
10497	035146 005037 036210		CLR EZDF	;CLEAR ZERO DIVIDE FLAG
10498	035152 005037 035054		CLR ESPOS	;CLEAR SHIFT POSITION
10499	035156 004737 034224		JSR PC,EIRT2	;INITIALIZE TEMPORARY RESULT BUFFER (ERT2) TO 0+
10500	035162 004737 034300		JSR PC,ETSTS1	;IS DIVP SRC1 = 0?
10501	035166 032737 000004 034610		BIT #4,ETOPSW	:
10502	035174 001152		BNE E1XZD	;BRANCH IF YES
10503	035176 042737 177767 034610		BIC #177767,ETOPSW	
10504	035204 013737 034610 035136		MOV ETOPSW,ESS1SN	;SAVE SRC1 SIGN
10505	035212 004737 034350		JSR PC,ETSTS2	;IS DIVP SRC2=0?
10506	035216 032737 000004 034610		BIT #4,ETOPSW	
10507	035224 001134		BNE E1DID	;BRANCH IF YES
10508	035226 042737 177767 034610		BIC #177767,ETOPSW	
10509	035234 013737 034610 035134		MOV ETOPSW,ESS2SN	;SAVE SRC2 SIGN
10510				
10511	035242 013737 034574 017364		MOV ESRO,E0	;MOVE DIVP SRC1 INTO ERT3
10512	035250 013737 034576 017366		MOV ESR1,E1	; USE ASHP - SRC = DIVP.SRC1
10513	035256 005037 017374		CLR E4	; DST = ERT3
10514	035262 005037 017376		CLR E5	; SHFT.CT=0
10515	035266 012737 000040 017370		MOV #40,E2	
10516	035274 012737 034464 017372		MOV #ERT3,E3	
10517	035302 004737 030642		JSR PC,EASHP	
10518	035306 142737 000001 034504		BICB #1,@#ERT3+20	;MAKE ERT3 SIGN - +
10519				
10520	035314 013737 034600 017364		MOV ESR2,E0	;MOVE DIVP SRC2 INTO ERT4
10521	035322 013737 034602 017366		MOV ESR3,E1	; USE ASHP - SRC = DIVP SRC2
10522	035330 005037 017374		CLR E4	; DST = ERT4
10523	035334 005037 017376		CLR E5	; SHFT.CT = 0
10524	035340 012737 000037 017370		MOV #37,E2	
10525	035346 012737 034506 017372		MOV #ERT4,E3	
10526	035354 004737 030642		JSR PC,EASHP	
10527	035360 142737 000001 034525		BICB #1,@#ERT4+17	;MAKE ERT4 SIGN +
10528				
10529	035366 012737 000040 017364	EPOSS1:	MOV #40,E0	;SHIFT DIVP SRC1 LEFT UNTIL SRC2 - SRC1(SHIFTED)
10530	035374 012737 034464 017366		MOV #ERT3,E1	; IS NEGATIVE. NOTE: LEN=40 IS LEGAL FOR EMUL.
10531	035402 012737 000037 017370		MOV #37,E2	;SUBTRACT ERT3 FROM ERT4
10532	035410 012737 034506 017372		MOV #ERT4,E3	; USE SUBP - SRC1 = ERT3
10533	035416 012737 000037 017374		MOV #37,E4	; SRC2 = ERT4
10534	035424 012737 034420 017376		MOV #ERT1,E5	; DST = ERT1
10535	035432 004737 023420		JSR PC,ESUBP	
10536	035436 032777 000010 161716		BIT #10,@EOPSW	;IS RESULT NEGATIVE?
10537	035444 001030		BNE EPOSD	;BRANCH IF YES
10538	035446 012737 000040 017364		MOV #40,E0	;SHIFT DIVP SRC1 LEFT 1 PLACE
10539	035454 012737 034464 017366		MOV #ERT3,E1	; USE ASHP - SRC = ERT3
10540	035462 012737 000001 017374		MOV #1,E4	; DST = ERT3
10541	035470 012737 000040 017370		MOV #40,E2	; SHFT.CT = 1
10542	035476 012737 034464 017372		MOV #ERT3,E3	;NOTE - LEGAL FOR EMULATOR.
10543	035504 004737 030642		JSR PC,EASHP	
10544	035510 005237 035054		INC ESPOS	;INCREMENT SHIFT POSITION
10545	035514 000724		BR EPOSS1	
10546	035516 000137 036054	E1DID:	JMP FDID	
10547	035522 000137 036140	E1XZD:	JMP EXZD	



```

10548
10549 035526 005737 035054      EPOSD:      TST ESPOS      ;SHIFT POSITION = 0?
10550 035532 001550                REQ EDID        ;BRANCH IF YES
10551
10552 035534 012737 000040 017364  EDIVL:      MOV #40,E0      ;REPOSITION ERT3 BACK 1 PLACE(RIGHT)
10553 035542 012737 034464 017366      MOV #ERT3,E1    ; USE ASHP - SRC = ERT3
10554 035550 012737 000377 017374      MOV #377,E4     ;           DST = ERT3
10555 035556 012737 000040 017370      MOV #40,E2     ;           SHFT.CT=-1
10556 035564 012737 034464 017372      MOV #ERT3,E3    ;NOTE - THIS IS LEGAL FOR EMULATOR
10557 035572 004737 030642      JSR PC,EASHP
10558
10559 035576 005037 035140                CLR ESUBCT      ;CLEAR SUBTRACT COUNTER
10560
10561 035602 012737 000040 017364      1$:  MOV #40,E0      ;SUBTRACT DIVP SRC1(SHIFTED) FROM DIVP SRC2
10562 035610 012737 034464 017366      MOV #ERT3,E1    ; USE SUBP - SRC1 = ERT3
10563 035616 012737 000037 017370      MOV #37,E2     ;           SRC2 = ERT4
10564 035624 012737 034506 017372      MOV #ERT4,E3    ;           DST = ERT4
10565 035632 012737 000037 017374      MOV #37,E4
10566 035640 012737 034506 017376      MOV #ERT4,E5
10567 035646 004737 023420      JSR PC,ESUBP
10568 035652 032777 000010 161502      BIT #10,@EOPSW ;IS RESULT OF SUBP POSITIVE?
10569 035660 001003                BNE ESBDT      ;BRANCH IF NO
10570 035662 005237 035140      INC ESUBCT     ;INCREMENT SUBTRACT COUNTER
10571 035666 000745                BR 1$
10572
10573 035670 012737 000040 017364  ESBDT:      MOV #40,E0      ;BACKUP TO LAST POSITIVE RESULT FROM SUBP
10574 035676 012737 034464 017366      MOV #ERT3,E1    ; USE ADDP - SRC1 = ERT3
10575 035704 012737 000037 017370      MOV #37,E2     ;           SRC2 = ERT4
10576 035712 012737 034506 017372      MOV #ERT4,E3    ;           DST = ERT4
10577 035720 012737 000037 017374      MOV #37,E4
10578 035726 012737 034506 017376      MOV #ERT4,E5
10579 035734 004737 023402      JSR PC,EADDP
10580
10581 035740 012737 000037 017364      MOV #37,E0      ;STORE SUBTRACT COUNTER IN RESULT(ERT2)
10582 035746 012737 034442 017366      MOV #ERT2,E1    ;SHIFT RESULT, THEN ENTER DIGIT
10583 035754 012737 000001 017374      MOV #1,E4       ; USE ASHP - SRC = ERT2
10584 035762 012737 000037 017370      MOV #37,E2     ;           DST=ERT2
10585 035770 012737 034442 017372      MOV #ERT2,E3    ;           SHFT.CT=1
10586 035776 004737 030642      JSR PC,EASHP
10587 036002 006337 035140      ASL ESUBCT     ;INSERT SUBTRACT COUNTER IN RESULT
10588 036006 006337 035140      ASL ESUBCT
10589 036012 006337 035140      ASL ESUBCT
10590 036016 006337 035140      ASL ESUBCT
10591 036022 153737 035140 034461      BISB ESUBCT,@#ERT2+17
10592 036030 005337 035054      DEC ESPOS      ;DECREMENT SHIFT POSITION
10593 036034 001237                BNE EDIVL      ;IS SHIFT POSITION=0? BRANCH IF NO
10594 036036 023737 035134 035136      CMP ESS2SN,ESS1SN ;DIVP SRC1 SIGN = DIVP SRC2 SIGN?
10595 036044 001403                BEQ EDID        ;BRANCH IF YES
10596 036046 152737 000001 034461      BISB #1,@#ERT2+17 ;NO - MAKE SIGN IN ERT2 NEGATIVE
10597
10598 036054 012737 000037 017364  EDID:      MOV #37,E0      ;MOVE RESULT INTO DST
10599 036062 012737 034442 017366      MOV #ERT2,E1    ; USE ASHP - SRC = ERT2
10600 036070 005037 017374      CLR F4         ;           DST = DIVP.DST
10601 036074 005037 017376      CLR E5         ;           SHFT.CT = 0

```

10602	036100	013737	034604	017370		MOV ESR4,E2	
10603	036106	013737	034606	017372		MOV ESR5,E3	
10604	036114	013737	034570	017360		MOV ESOSTK,EORSTK	
10605	036122	013737	034572	017362		MOV ESOPSW,EOPSW	
10606	036130	004737	030642			JSR PC,EASHP	
10607	036134	000137	034110			JMP EXMD	:EXIT DIVP
10608							
10609	036140	013737	034572	017362	EXZD:	MOV ESOPSW,EOPSW	:EXIT FROM DIVIDE BY ZERO
10610	036146	052777	000003	161206		BIS #3,@EOPSW	:SET V & C COND. CODES
10611							:SET ZERO DIVIDE FLAG TO SIGNAL
10612	036154	012737	177777	036210		MOV #177777,EZDF	:TABLE DRIVER NOT TO COMPARE
10613							:ANYTHING EXCEPT V & C COND. CODE
10614							:SAVE POINTER TO START & # OF BYTES
10615	036162	013737	034576	036212		MOV ESR1,EZDBEG	:OF DST STRING
10616							:CONTENTS OF DST STRING UNPREDICTABLE
10617	036170	013737	034574	036214		MOV ESRO,EZDEND	:AFTER ZERO DIVP
10618	036176	006237	036214			ASR EZDEND	
10619	036202	005237	036214			INC EZDEND	
10620	036206	000207				RTS PC	:RESULTS
10621							
10622	036210	000000			EZDF:	.WORD 0	
10623	036212	000000			EZDBEG:	.WORD 0	
10624	036214	000000			EZDEND:	.WORD 0	
10625							

10627					.SBTTL	LOAD DESCRIPTORS
10628	036216	013701	017364	EL2D0:	MOV E0,R1	:GET REGISTER POINTER
10629	036222	012102		EL2:	MOV (R1)+,R2	:GET ADDRESS OF DESCRIPTOR
10630	036224	011237	017364		MOV (R2),E0	:LOAD 1ST WORD OF DESC INTO E0
10631	036230	016237	000002	017366	MOV 2(R2),E1	:LOAD 2ND WORD OF DESC INTO E1
10632	036236	011102			MOV (R1),R2	:GET ADDRESS OF NEXT DESC
10633	036240	011237	017370		MOV (R2),E2	:LOAD 1ST WORD OF DESC INTO E2
10634	036244	016237	000002	017372	MOV 2(R2),E3	:LOAD 2ND WORD OF DESC INTO E3
10635	036252	013700	017360	EXL2:	MOV EORSTK,R0	:RETURN CLEAN UP
10636	036256	013720	017364		MOV E0,(R0)+	:R0=R0
10637	036262	013720	017366		MOV E1,(R0)+	:R1=R1
10638	036266	013720	017370		MOV E2,(R0)+	:R2=R2
10639	036272	013720	017372		MOV E3,(R0)+	:R3=R3
10640	036276	013720	017374		MOV E4,(R0)+	:R4=R4
10641	036302	013720	017376		MOV E5,(R0)+	:R5=R5
10642	036306	013720	017400		MOV E6,(R0)+	:R6=R6
10643	036312	012777	000017	161042	MOV #17,@EOPSW	:SET ALL COND. CODE BITS.
10644	036320	000207			RTS PC	
10645						
10646	036322			EL2D1:		
10647	036322	013701	017366		MOV E1,R1	
10648	036326	000735			BR EL2	
10649						
10650	036330			EL2D2:		
10651	036330	013701	017370		MOV E2,R1	
10652	036334	000732			BR EL2	
10653						
10654	036336			EL2D3:		
10655	036336	013701	017372		MOV E3,R1	
10656	036342	000727			BR EL2	
10657						
10658	036344			EL2D4:		
10659	036344	013701	017374		MOV E4,R1	
10660	036350	062737	000004	017374	ADD #4,E4	
10661	036356	000721			BR EL2	
10662						
10663	036360			EL2D5:		
10664	036360	013701	017376		MOV E5,R1	
10665	036364	062737	000004	017376	ADD #4,E5	
10666	036372	000713			BR EL2	
10667						
10668	036374			EL2D6:		
10669	036374	013701	017400		MOV E6,R1	
10670	036400	012102			MOV (R1)+,R2	
10671	036402	011237	017364		MOV (R2),E0	
10672	036406	016237	000002	017366	MOV 2(R2),E1	
10673	036414	012102			MOV (R1)+,R2	
10674	036416	011237	017370		MOV (R2),E2	
10675	036422	016237	000002	017372	MOV 2(R2),E3	
10676	036430	010137	017400		MOV R1,E6	:NOTE:L2D6 UPDATES R6 (POPS STACK)
10677	036434	000137	036252		JMP EXL2	
10678						
10679	036440			EL2D7:		
10680	036440	013701	050014		MOV TINST+2,R1	

10681	036444	011137	017364		MOV (R1),E0
10682	036450	016137	000002	017366	MOV 2(R1),E1
10683	036456	013701	050016		MOV TINST+4,n
10684	036462	011137	017370		MOV (R1),E2
10685	036466	016137	000002	017372	MOV 2(R1),E3
10686	036474	000137	036252		JMP EXL2
10687					
10688					

10690	036500	013701	017364		EL3D0:	MOV E0,R1	:GET REGISTER POINTER
10691	036504	012102			EL3:	MOV (R1)+,R2	:GET ADDRESS OF DESCRIPTOR
10692	036506	011237	017364			MOV (R2),E0	:LOAD 1ST WORD OF DESC INTO E0
10693	036512	016237	000002	017366		MOV 2(R2),E1	:LOAD 2ND WORD OF DESC INTO E1
10694	036520	012102				MOV (R1)+,R2	:GET ADDRESS OF NEXT DESC
10695	036522	011237	017370			MOV (R2),E2	:LOAD 1ST WORD OF DESC INTO E2
10696	036526	016237	000002	017372		MOV 2(R2),E3	:LOAD 2ND WORD OF DESC INTO E3
10697	036534	011102				MOV (R1),R2	:GET ADDRESS OF NEXT DESC
10698	036536	011237	017374			MOV (R2),E4	:LOAD 1ST WORD OF DESC INTO E4
10699	036542	016237	000002	017376		MOV 2(R2),E5	:LOAD 2ND WORD OF DESC INTO E5
10700	036550	000137	036252			JMP EXL2	
10701							
10702	036554				EL3D1:		
10703	036554	013701	017366			MOV E1,R1	
10704	036560	000751				BR EL3	
10705							
10706	036562				EL3D2:		
10707	036562	013701	017370			MOV E2,R1	
10708	036566	000746				BR EL3	
10709							
10710	036570				EL3D3:		
10711	036570	013701	017372			MOV E3,R1	
10712	036574	000743				BR EL3	
10713							
10714	036576				FL3D4:		
10715	036576	013701	017374			MOV E4,R1	
10716	036602	000740				BR EL3	
10717							
10718	036604				EL3D5:		
10719	036604	013701	017376			MOV E5,R1	
10720	036610	000735				BR EL3	
10721							
10722	036612				EL3D6:		
10723	036612	013701	017400			MOV E6,R1	
10724	036616	012102				MOV (R1)+,R2	
10725	036620	011237	017364			MOV (R2),E0	
10726	036624	016237	000002	017366		MOV 2(R2),E1	
10727	036632	012102				MOV (R1)+,R2	
10728	036634	011237	017370			MOV (R2),E2	
10729	036640	016237	000002	017372		MOV 2(R2),E3	
10730	036646	012102				MOV (R1)+,R2	
10731	036650	011237	017374			MOV (R2),E4	
10732	036654	016237	000002	017376		MOV 2(R2),E5	
10733	036662	010137	017400			MOV R1,E6	:NOTE:L2D6 UPDATES R6 (POPS STACK)
10734	036666	000137	036252			JMP EXL2	
10735							
10736	036672				EL3D7:		
10737	036672	013701	050014			MOV TINST+2,R1	
10738	036676	011137	017364			MOV (R1),E0	
10739	036702	016137	000002	017366		MOV 2(R1),E1	
10740	036710	013701	050016			MOV TINST+4,R1	
10741	036714	011137	017370			MOV (R1),E2	
10742	036720	016137	000002	017372		MOV 2(R1),E3	
10743	036726	013701	050020			MOV TINST+6,R1	

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22 N 11  
LOAD DESCRIPTORS PAGE 88-1 SEQUENCE 143

10744 036732 011137 017374  
10745 036736 016137 000002  
10746 036744 000137 036252  
10747

017376

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

```

10749          .SBTTL  CIS INSTRUCTION TEST LOOP
10750
10751 036750 012737 177777 001744 SEEDST: MOV #177777,ESEED      ;SET ENTER RNG SEED FLAG
10752 036756 000476          BR DVTST
10754 036760 012737 177777 001750 START:  MOV #177777,N200M      ;SET FLAG TO INDICATE THAT PROG WAS
10755          ;STARTED AT LOC 200
10757 036766 012737 177777 002206 QVST:   MOV #177777,QVMODE      ;SET QVMODE FLAG
10763 036774 005037 001746          NST:   CLR DENS          ;CLEAR DON'T ENTER NORMAL RNG SEED FLAG.
10764 037000 005037 001120          CLR $TESTN      ;CLEAR TEST COUNT
10765 037004 012706 001100          MOV     #STACK,SP ;SETUP THE STACK POINTER
(1) 037010 012737 111214 000034          MOV     #$TRAP,@#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
(1) 037016 012737 000340 000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
(1) 037024 012737 111246 000024          MOV     #$PWRDN,@#PWRVEC ;POWER FAILURE VECTOR
(1) 037032 012737 000340 000026          MOV     #340,@#PWRVEC+2 ;LEVEL 7
(2) 037040 013746 000004          MOV     @#4,-(SP)     ;SAVE ERROR VECTOR
(2) 037044 013746 000006          MOV     @#6,-(SP)
(2) 037050 012737 037064 000004          MOV     #64$,4       ;SET UP TIME OUT VECTOR
(2) 037056 005777 142654          TST     @SWR          ;TRY TO REFERENCE HARDWARE SWR
(2) 037062 000407          BR      65$          ;BRANCH IF NO TIMEOUT TRAP OCCURS
(2) 037064 012737 000176 001736 64$:   MOV     #SWREG,SWR    ;POINT TO SOFTWARE SWR
(2) 037072 012737 000174 001740          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
(2) 037100 022626          CMP     (SP)+,(SP)+  ;RESTORE STACK
(2) 037102 012637 000006 65$:   MOV     (SP)+,@#6    ;RESTORE ERROR VECTOR
(2) 037106 012637 000004          MOV     (SP)+,@#4
(1) 037112 005037 001122          CLR     $PASS        ;CLEAR PASS COUNT
(1) 037116 132737 000200 001135          BITB   #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT
(1) 037124 001403          BEQ    3$            ;YES,USE NON-APT SWITCH
(1) 037126 012737 001136 001736          MOV     #SWREG,SWR   ;NO,USE APT SWITCH REGISTER
(1) 037134          3$:
10766          ; NO QUESTIONS ASKED - EXERCISES FIXED TABLE TEST
10767          ; CONDITIONS FIRST THEN ENTERS RANDOM MODE TESTING
10768 037134 012737 177777 002074          MOV #177777,FSRUN
10769 037142 005037 001660          CLR INCSQ1          ;PRIOR TO EACH TEST, BUFFERS WILL BE INITIALIZED TO ZERO
10770 037146 005037 001662          CLR INCSQ2
10771 037152 000462          BR COMST
10772 037154 005037 002206          DVTST: CLR QVMODE
10773 037160 005037 001746          CLR DENS          ;CLEAR DON'T ENTER NORMAL RNG SEED FLAG
10774 037164 012706 001100          MOV     #STACK,SP  ;SETUP THE STACK POINTER
(1) 037170 012737 111214 000034          MOV     #$TRAP,@#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
(1) 037176 012737 000340 000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
(1) 037204 012737 111246 000024          MOV     #$PWRDN,@#PWRVEC ;POWER FAILURE VECTOR
(1) 037212 012737 000340 000026          MOV     #340,@#PWRVEC+2 ;LEVEL 7
(2) 037220 013746 000004          MOV     @#4,-(SP)   ;SAVE ERROR VECTOR
(2) 037224 013746 000006          MOV     @#6,-(SP)
(2) 037230 012737 037244 000004          MOV     #64$,4     ;SET UP TIME OUT VECTOR
(2) 037236 005777 142474          TST     @SWR          ;TRY TO REFERENCE HARDWARE SWR
(2) 037242 000407          BR      65$          ;BRANCH IF NO TIMEOUT TRAP OCCURS
(2) 037244 012737 000176 001736 64$:   MOV     #SWREG,SWR    ;POINT TO SOFTWARE SWR
(2) 037252 012737 000174 001740          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
(2) 037260 022626          CMP     (SP)+,(SP)+  ;RESTORE STACK
(2) 037262 012637 000006 65$:   MOV     (SP)+,@#6    ;RESTORE ERROR VECTOR
(2) 037266 012637 000004          MOV     (SP)+,@#4
(1) 037272 005037 001122          CLR     $PASS        ;CLEAR PASS COUNT
(1) 037276 132737 000200 001135          BITB   #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT

```

```

(1) 037304 001403          BEQ      3$          ;;YES,USE NON-APT SWITCH
(1) 037306 012737 001136 001736 3$:  MOV     #SWREG,SWR    ;;NO,USE APT SWITCH REGISTER
(1) 037314          ; RESULTS IN DIALOG WITH USER TO DETERMINE
10775          ; EXACT RUN MODE DESIRED.
10776          CLR FSRUN
10781 037314 005037 002074          COMST:
10782 037320          CLR @TPSW          ;SET PROCESSOR PRIORITY TO ZERO
10783 037320 005077 142340          MOV #22$,@#RESVEC    ;CHECK FOR SWITCH ON CIS MODULE TO BE IN CORRECT POSITIO
10784 037324 012737 037356 000010  CLR @#RESVEC+2
10785 037332 005037 000012          76001          ;THIS INST SHOULD TRAP TO LOC 10 IF
10786 037336 076001          ; SWITCH POSITION IS OK; OTHERWISE
10787          ; IT WILL ACT LIKE A 'NOP'.
10788          ;INDICATE THAT SWITCH POSITION IS INCORECT
10789 037340          PRINTB #SWG
(6) 037340 012746 013454          MOV     #SWG,-(SP)
(3) 037344 010600          MOV     SP,R0
(4) 037346 004737 065410          JSR PC,FPRINT
10790 037352 000000          HALT
10791 037354 000761          BR COMST
10798 037356 005726          22$:  TST (SP)+          ;FIX UP STACK
10799 037360 005726          TST (SP)+
10800 037362 005737 001744          TST ESEED          ;GET NEW SEED CONSTANTS?
10801 037366 001437          BEQ 2$          ;BRANCH IF NO
10802 037370          111$:  PRINTB #ACCSEED    ;PRINT MESSAGE: ENTER RNG SEED CONSTANTS
(6) 037370 012746 016111          MOV     #ACCSEED,-(SP)
(3) 037374 010600          MOV     SP,R0
(4) 037376 004737 065410          JSR PC,FPRINT
10803 037402 004737 065102          JSR PC,ACCOCT    ;GET OCTAL SEED
10804 037406 000770          BR 111$          ;<CR> RETURN
10805 037410 000240          NOP          ;XXXXXX<CR> RETURN
10806 037412 000240          NOP          ;XXXXXX<-> RETURN
10807 037414 012637 063612          MOV (SP)+,RNCON  ;INSERT FIRST SEED CONSTANT
10808 037420 004737 065102          JSR PC,ACCOCT    ;GET SECOND SEED
10809 037424 000761          BR 111$
10810 037426 000240          NOP
10811 037430 000240          NOP
10812 037432 012637 063614          MOV (SP)+,RP1    ;INSERT SECOND SEED CONSTANT
10813 037436 004737 065102          JSR PC,ACCOCT    ;GET THIRD SEED
10814 037442 000752          BR 111$
10815 037444 000240          NOP
10816 037446 000240          NOP
10817 037450 012637 063616          MOV (SP)+,RP2    ;INSERT THIRD SEED CONSTANT
10818 037454 012737 177777 001746  MOV #177777,DENS ;SET DON'T ENTER NORMAL SEED FLAG
10819 037462 005037 001744          CLR ESEED
10820 037466 012737 057202 000010  2$:  MOV #ILLSER,@#RESVEC ;SETUP ILLEGAL INST TRAP CATCHER
10822 037474 005037 000012          CLR @#RESVEC+2
10823 037500 012737 057364 000250  MOV #MMVIOL,@#MMVEC ;SETUP MEMORY MANAGEMENT TRAP CATCHER
10824 037506 005037 000252          CLR @#MMVEC+2
10825 037512 012737 057062 000004  MOV #HLTSER,@#ERRVEC ;SETUP TIMEOUT INST. TRAP VECTOR
10826 037520 005037 000006          CLR @#ERRVEC+2
10827 037524 004737 054252          JSR PC,SIZEPT    ;SETUP PROCESSOR DEPENDENT CONSTANTS
10828 037530 023727 000042 053424  1$:  CMP @#42,#ENDAD  ;!F IN ACT CHAIN MODE SKIP PRINTING OF PROG TITLE
10829 037536 001405          BEQ 14$
10830 037540 005737 001122          TST $PASS          ;IDENTIFY PROGRAM ON 1ST PASS ONLY

```



10831	037544	001002			BNE 14\$		
10832	037546	104400			TYPE		
10833	037550	016621			PNAME		:TYPE PROGRAM NAME
10834	037552	005037	002470		CLR ONEINS	14\$:	:CLEAR SINGLE INST TEST FLAG
10835	037556	012737	000207	061174	MOV #207,DI		:INHIBIT INTERRUPT DURING INTR SERVICE DIVPI
10836	037564	012737	000414	047720	MOV #414,TOLTC		:INHIBIT LTC TURN ON
10837	037572	012737	000403	047752	MOV #403,TOPC2		:INHIBIT LATENCY & INTERRUPTABILITY TURN ON
10838	037600	012737	000403	047762	MOV #403,TOPC1		
10839	037606	005037	002054		CLR ERRCT		:CLEAR ERROR COUNT
10840	037612	005037	002162		CLR DEN		:CLEAR D-SPACE FNABLE FLAG
10841	037616	005037	003030		CLR LCNT		:CLEAR LTC COUNT
10842	037622	005037	002544		CLR LATEN		:CLEAR LATENCY TESTING FLAG
10843	037626	005037	002216		CLR STOPTF		:CLEAR FLAG WHICH WHEN CLR WILL TRIGGER
10844							: PROGRAM TO REQUEST STOP TEST NUMBER
10845	037632	005037	002132		CLR FATAL		:CLEAR FATAL ERROR INDICATOR
10846	037636	005037	001760		CLR RANDOM		:CLEAR RANDOM EXERCISE MODE FLAG
10847	037642	005037	002042		CLR NOERDS		:CLEAR 'NO-ERROR DISPLAY' SWITCH
10856	037646	004737	055446		JSR PC,SETPAR	66\$:	:SETUP PAR'S (MEM MGMT)
10857	037652	013700	001654		MOV PSEED,RO		:FORM RNG PRINT SEED MASK
10858	037656	005100			COM RO		
10859	037660	005200			INC RO		
10860	037662	010037	002076		MOV RO,MSEED		
10862	037666	005737	001746		TST DENS		:ENTER NORMAL SEED?
10863	037672	001011			BNE 61\$		:BRANCH IF NO
10865	037674	013737	063620	063612	MOV KRNCON,RNCON		:INITIALIZE RANDOM # GENERATOR
10866	037702	013737	063622	063614	MOV KRP1,RP1		
10867	037710	013737	063624	063616	MOV KRP2,RP2		
10868	037716	012737	072222	072220	MOV #IL2D,INPTBL	61\$:	:INITIALIZE INPUT TABLE POINTER
10869	037724	005737	002074		TST FSRUN		:FIELD SERVICE TYPE RUN
10870	037730	001431			BEQ 13\$		:BRANCH IF NO TO ENTER DIALOG WITH USER
10871							:DETERMINE IF LINE CLOCK IS AVAILABLE FOR
10872							: FIELD SERVICE TYPE RUN
10873	037732	004737	062366		JSR PC,LTCP	31\$:	:IS LTC ON SYSTEM?
10874	037736	000137	037776		JMP 32\$		:NO - CANT TEST INTERRUPTABILITY
10875	037742	005737	001122		TST 3PASS		:IDENTIFY INTR SOURCE ON 1ST PASS
10876	037746	001005			BNE 103\$		
10877	037750				PRINTB #KW11L		:INDICATE THAT LINE CLOCK WILL BE USED
(6)	037750	012746	013620		MOV #KW11L,-(SP)		
(3)	037754	010600			MOV SP,RO		
(4)	037756	004737	065410		JSR PC,FPRINT		
10878							:FOR INTERRUPT SOURCE.
10879	037762	004737	062250		JSR PC,LTC SUP	103\$:	:SYNC UP TO LTC
10880	037766	004737	062312		JSR PC,LTC CNT		:DETERMINE COUNT PER CLOCK TICK
10881	037772	000137	040546		JMP FDIALG		:SKIP OVER DIALOG WITH USER
10882	037776				PRINTB #NOINT	32\$:	:PRINT CANT TEST INTR MESSAGE
(6)	037776	012746	013401		MOV #NOINT,-(SP)		
(3)	040002	010600			MOV SP,RO		
(4)	040004	004737	065410		JSR PC,FPRINT		
10883	040010	000137	040546		JMP FDIALG		
10884	040014				PRINTB #ASKINT	13\$:	:ASK IF INTERRUPTABILITY MODE IS DESIRED?
(6)	040014	012746	012721		MOV #ASKINT,-(SP)		
(3)	040020	010600			MOV SP,RO		
(4)	040022	004737	065410		JSR PC,FPRINT		
10885	040026	004737	064546		JSR PC,YORN		:ACCEPT ASCIZ FROM TTY

```

10886 040032 000137 040336      JMP ARMQ      ;N RESPONSE
10887 040036 000137 040052      JMP 5$       ;Y RESPONSE
10888 040042 000137 040336      JMP ARMQ      ;R OR H RESPONSE (ILLEGAL HERE)
10889 040046 000137 040336      JMP ARMQ      ;C RESPONSE (ILLEGAL HERE)
10890
10891 040052      5$: PRINTB #ASKSRC      ;ASK FOR INTERRUPT SOURCE
      (6) 040052 012746 013104      MOV #ASKSRC,-(SP)
      (3) 040056 010600      MOV SP,R0
      (4) 040060 004737 065410      JSR PC,FPRINT
10892 040064 004737 064546      JSR PC,YORN   ;ACCEPT ASCIZ
10893 040070 000137 040110      JMP 52$      ;(N) KW11-P @100KHZ
10894 040074 000137 040134      JMP 53$      ;(Y) KW11-P EXT OSC
10895 040100 000137 040174      JMP 54$      ;(R) LINE TIME CLOCK
10896 040104 000137 040220      JMP 55$      ;(C) KW11-P @10KHZ
10897
10898 040110      52$: JSR PC,PC1CK      ;MAKE KW11-P @100KHZ THE INTERRUPT SOURCE
10899 040110 004737 061602      JMP MNOPC1    ;CHECK FOR (& SETUP) P-CLK 1
10900 040114 000137 040242      JSR PC,PC2CK  ;NOT PRESENT ON SYSTEM RETURN
10901 040120 004737 061716      JSR PC,PC2CK  ;P-CLK EXISTS RETURN - CHECK FOR (& SETUP) 2ND PCLK
10902      ; FOR LATENCY TESTING
10903 040124 000137 040260      JMP MNOPC2    ;NOT PRESENT ON SYSTEM RETURN
10904 040130 000137 040272      JMP ADIQ      ;2ND PCLK EXISTS
10905
10906 040134      53$: JSR PC,PC1CK      ;MAKE KW11-P WITH EXTERNAL OSCILLATOR THE
10907      ; INTERRUPT SOURCE
10908 040134 004737 061602      JMP MNOPC1    ;CHECK FOR (& SETUP) P-CLK ON SYSTEM
10909 040140 000137 040242      BIS #6,@PC1CSR ;NOT PRESENT ON SYSTEM RETURN
10910 040144 052777 000006 142346      JSR PC,PC2CK  ;SET PCLK 1 FOR EXTERNAL OSCILLATOR
10911 040152 004737 061716      JMP MNOPC2    ;CHECK FOR 2ND P-CLK FOR LATENCY TESTING
10912 040156 000137 040260      BIS #6,@PC2CSR ;NOT PRESENT RETURN
10913 040162 052777 000006 142340      JMP ADIQ      ;SET PCLK2 FOR EXTERNAL OSC
10914 040170 000137 040272
10915
10916 040174      54$: JSR PC,LTCPC      ;MAKE LINE TIME CLOCK THE INTR SOURCE
10917 040174 004737 062366      JMP MNOLTC    ;CHECK FOR LTC ON SYSTEM
10918 040200 000137 040242      JSR PC,LTCSUP ;NOT PRESENT RETURN
10919 040204 004737 062250      JSR PC,LTCCNT ;SYNC UP TO LTC
10920 040210 004737 062312      JMP ARMQ      ;DETERMINE COUNT PER CLOCK TICK
10921 040214 000137 040336
10922
10923 040220      55$: JSR PC,PC1CK      ;MAKE KW11-P @10KHZ THE INTERRUPT SOURCE
10924 040220 004737 061602      JMP MNOPC1    ;CHECK FOR P-CLK ON SYSTEM
10925 040224 000137 040242      BIS #2,@PC1CSR ;NOT PRESENT RETURN
10926 040230 052777 000002 142262      JMP ARMQ      ;SET PCLK FOR 10KHZ (NO LATENCY TESTING)
10927 040236 000137 040336
10928
10929 040242      MNOLTC:
10930 040242      MNOPC1: PRINTB #NOINT      ;PRINT CANT TEST INTERRUPTABILITY MESSAGE
      (6) 040242 012746 013401      MOV #NOINT,-(SP)
      (3) 040246 010600      MOV SP,R0
      (4) 040250 004737 065410      JSR PC,FPRINT
10931 040254 000137 040336      JMP ARMQ
10932 040260      MNOPC2: PRINTB #NOLAT     ;PRINT CANT TEST LATENCY MESSAGE
      (6) 040260 012746 013330      MOV #NOLAT,-(SP)

```

```

(3) 040264 010600      MOV      SP,RO
(4) 040266 004737 065410 JSR PC,FPRINT
10933 040272      PRINTB #ASKDI      ;ASK IF USER ALLOWS AN INTERRUPT DURING CIS
(6) 040272 012746 013217 ADIO:
(3) 040276 010600      MOV      #ASKDI,-(SP)
(4) 040300 004737 065410 JSR PC,FPRINT
10934      ; INST EXECUTED ON NORMAL INTR SERVICE ROUTINE
10935 040304 004737 064546 JSR PC,YORN      ;ACCEPT ASCIZ FROM TTY
10936 040310 000137 040336 JMP ARMQ        ;N RESPONSE
10937 040314 000137 040330 JMP 1$         ;Y RESPONSE
10938 040320 000137 040336 JMP ARMQ        ;ILLEGAL RESPONSE
10939 040324 000137 040336 JMP ARMQ        ;ILLEGAL RESPONSE
10940 040330 013737 001670 061174 1$: MOV KNOP,DI      ;OVERWRITE 'RTS PC' TO ALLOW P-CLK
10941      ; INTERRUPT DURING CIS INST EXECUTED
10942      ; WITHIN NORMAL P-CLK INTERRUPT SERVICE ROUTINE.
10943
10944
10945
10946 040336      ARMQ: PRINTB #ASKRM      ;ASK IF RANDOM EXERCISE MODE IS DESIRED?
(6) 040336 012746 012461 MOV      #ASKRM,-(SP)
(3) 040342 010600      MOV      SP,RO
(4) 040344 004737 065410 JSR PC,FPRINT
10947 040350 004737 064546 JSR PC,YORN      ;ACCEPT ASCIZ FROM TTY
10948 040354 000137 040412 JMP 2$         ;N RESPONSE
10949 040360 000137 040374 JMP 3$         ;Y RESPONSE
10950 040364 000137 040506 JMP 1$         ;R OR H RESPONSE (ILLEGAL HERE)
10951 040370 000137 040506 JMP 1$         ;C RESPONSE (ILLEGAL HERE)
10952 040374 012737 177777 001760 3$: MOV #177777,RANDOM ;SET RANDOM FLAG
10953 040402 012737 072144 072220 MOV #IDUM,INPTBL ;SET INPUT TABLE POINTER TO DUMMY INPUT TABLE
10954 040410 000436      BR 1$
10955 040412      2$: PRINTB #ASKMOD      ;ASK FOR PROCESSOR TEST MODE
(6) 040412 012746 012527 MOV      #ASKMOD,-(SP)
(5) 040416 010600      MOV      SP,RO
(4) 040420 004737 065410 JSR PC,FPRINT
10956 040424 004737 057462 JSR PC,KSORU
10957 040430 000770      BR 2$
10958 040432      20$: PRINTB #ASKMM      ;ASK FOR MEM MGMT TEST MODE
(6) 040432 012746 012613 MOV      #ASKMM,-(SP)
(3) 040436 010600      MOV      SP,RO
(4) 040440 004737 065410 JSR PC,FPRINT
10959 040444 004737 064546 JSR PC,YORN
10960 040450 000137 040502 JMP 23$        ;N RESPONSE - MEM MGMT OFF
10961 040454 000137 040472 JMP 22$        ;D RESPONSE - D SPACE ENABLED
10962 040460 000137 040506 JMP 1$         ;H RESPONSE - D SPACE DISABLED
10963 040464 000137 040432 JMP 20$        ;ILLEGAL RESPONSE - ASK AGAIN
10964 040470 000406      BR 1$
10965 040472 012737 177777 002162 22$: MOV #177777,DEN      ;SET D ENABLED FLAG
10966 040500 000402      BR 1$
10967 040502 005037 002156 23$: CLR MMFLG      ;SET NO MEM MGMT FLAG
10968 040506      1$: PRINTB #ASK      ;ASK FOR SPECIFIC INST TO TEST
(6) 040506 012746 012412 MOV      #ASK,-(SP)
(3) 040512 010600      MOV      SP,RO
(4) 040514 004737 065410 JSR PC,FPRINT
10969 040520 004737 067664 JSR PC,ACASZ      ;ACCEPT ASCIZ FROM TTY

```

```

10970 040524 005737 002462          TST ACINST          ;DEFAULTED TO ALL INSTRUCTIONS?
10971 040530 001406          BEQ FDIALG          ;BRANCH IF YES
10972 040532 004737 067562          JSR PC,SFCI        ;LOOK FOR MATCH BETWEEN INST ENTERED
10973                                     ;AND LIST OF CIS INST ASCII.
10974 040536 000763          BR 1$              ;NO MATCH RETURN
10975 040540 012737 177777 002470    MOV #177777,ONEINS ;MATCH - SET SINGLE INST TESTING FLAG
10976 040546 013737 072220 002134    FDIALG: MOV INPTBL,INPT ;INITIALIZATION
10977 040554 005037 001420          CLR TOTTC          ;ZERO COUNT OF TOTAL TESTS EXECUTED
10978 040560 005037 001416          CLR TOTTC          ;ZERO COUNT OF INVALID TESTS - TESTS ABORTED
10979 040564 005037 001422          CLR INVTC          ;ZERO COUNT OF REDUNDANT TESTS - TESTS ABORTED.
10980 040570 005037 001424          CLR REDTC
10981 040574 005002          CLR R2
10982 040576 013737 001670 047772    MOV KNOP,PREINS   ;INSERT NOP BEFORE INST UNDER TEST.
10983 040604 012700 120606          MOV #XLTL1,R0     ;INITIALIZE MOVTC TRANSLATION TABLES
10984 040610 012701 121206          MOV #ELTBL,R1    ; AS FOLLOWS; 1 IN LOC 0, 2 IN LOC 1, ETC.
10985 040614 005202          11$: INC R2
10986 040616 110220          MOVB R2,(R0)+
10987 040620 020001          CMP R0,R1
10988 040622 103774          BLO 11$
10989 040624 005737 002074          TST FSRUN         ;NORMAL FIELD SERVICE TYPE RUN?
10990 040630 001426          BEQ NITE          ;BRANCH IF NO
10992 040632 023727 000042 053424    CMP @#42,#ENDAD  ;IF IN ACT CHAIN MODE SKIP OVER PRINTING OF HEADER
10993 040640 001412          BEQ 15$
10994 040642 005737 002206          TST QVMODE        ;IF IN QVMODE PRINT QV MODE
10995 040646 001403          BEQ 1$
10996 040650 104400          TYPE
10997 040652 016032          QVHDR
10998 040654 000404          BR 15$
10999 040656 104400          1$: TYPE
11000 040660 015720          FSHDR            ;PRINT FIELD SERVICE HEADER INFO
11028 040662 104400          10$: TYPE        ;PRINT PASS TIME MESSAGE
11029 040664 016174          FSHDR1
11031 040666 012737 071576 002142    15$: MOV #YL2D,EMPTR
11033 040674 042737 000007 050012    BIC #7,TINST     ;CLEAR REGISTER FIELD (FOR L2D DISPLAY ONLY)
11034 040702 004737 063436          JSR PC,IDINST    ;IDENTIFY INST UNDER TEST
11036
11037 040706 005037 042032          NITE: CLR MTYPE   ;CONTROL IS PASSED TO THIS POINT WHENEVER
11038                                     ; ALL TEST CONDITIONS FOR A GIVEN INPUT
11039                                     ; TABLE HAVE BEEN EXHAUSTED. PRIOR
11040                                     ; TO ENTRY TO THIS POINT, THE INPUT
11041                                     ; TABLE POINTER (INPTP) HAS BEEN UPDATED
11042                                     ; TO POINT TO THE NEXT INPUT TABLE
11043                                     ; OF TEST CONDITIONS. IN RANDOM MODE
11044                                     ; CONTROL IS PASSED HERE FOLLOWING EACH TEST.
11045 040712 005037 002472          CLR PZCODE
11046 040716 005037 002454          CLR PKPTW
11047 040722 005037 002440          CLR DECINS
11048 040726 005037 002456          CLR NDESC
11049 040732 005737 001760          TST RANDOM       ;RUNNING IN RANDOM EXERCISE MODE?
11050 040736 001436          BEQ 2$           ;BRANCH IF NO
11051 040740 004737 062452          JSR PC,SRNGST    ;SAVE RANDOM # GENERATOR STATE AT START OF EACH TEST
11053 040744 013700 001420          MOV TOTTC,R0     ;PRINT RANDOM # GEN SEED?
11054 040750 043700 002076          BIC MSEED,R0
11055 040754 001016          BNE 3$           ;BRANCH IF NO

```

```

11056 040756 005737 002044      TST PROGD          ;PROGRESS DISPLAY?
11057 040762 001013      BNE 3$            ;BRANCH IF NO
11058 040764                PRINTB #FORM36,RNCON,RP1,RP2 ;PRINT 3 SEED CONSTANTS
      (9) 040764 013746 063616      MOV RP2,-(SP)
      (8) 040770 013746 063614      MOV RP1,-(SP)
      (7) 040774 013746 063612      MOV RNCON,-(SP)
      (6) 041000 012746 014314      MOV #FORM36,-(SP)
      (3) 041004 010600                MOV SP,R0
      (4) 041006 004737 065410      JSR PC,FPRINT
11059                                ;NOTE: TO USE SEED, LOAD 3 CONSTANTS INTO
11060                                ; KRNCON,KRP1,KRP2 THEN RESTART AT LOC 204.
11067 041012 005737 002470      3$: TST ONEINS    ;SINGLE INST TESTING?
11068 041016 001002                BNE 1$            ;BRANCH IF YES
11069 041020 004737 062736      JSR PC,GENR1      ;GENERATE NEXT RANDOM CIS INST TO TEST
11070 041024 004737 062776      1$: JSR PC,LDINPT ;LOAD UP DUMMY INPUT TABLE USING RANDOM # GEN.
11071 041030 004737 063210      JSR PC,LDCON     ;LOAD MISCELLANEOUS CONSTANTS USING RANDOM # GEN.
11072 041034 013700 002134      2$: MOV INPTP,R0  ;R0 POINTS TO ENTRY IN INPUT TABLE
11073 041040 016001 000002      MOV 2(R0),R1
11074 041044 042701 177776      BIC #177776,R1  ;CLEAR OUT ALL BUT TYPE BIT IN ENTRY TYPE WORD
11075 041050 006301                ASL R1
11076 041052 062701 001426      ADD #ITYPE,R1
11077 041056 000171 000000      JMP @ (R1)      ;DISPATCH ON INPUT TABLE ENTRY TYPE
11078
11079 041062                TYPE0:          ;INPUT PARAMETERS FULLY SPECIFIED IN INPUT TABLE ENTRY
11080 041062 012737 177777 042032  MOV #177777,MTYPE ;SET TYPE FLAG TO INDICATE TYPE 0 ENTRY
11081 041070 010001                MOV R0,R1        ;SETUP PARAMETER TABLE POINTERS
11082 041072 062701 000004      ADD #4,R1        ;R1 POINTS TO iP1
11083 041076 012702 001570      MOV #PTP01,R2   ;R2 POINTS TO TOP OF PARAMETER TABLE POINTER LIST
11084 041102 010122                1$: MOV R1,(R2)+    ;LOAD PTP
11085 041104 062701 000002      ADD #2,R1
11086 041110 022702 001636      CMP #PTP24,R2  ;ALL PTP'S LOADED
11087 041114 103372                BHIS 1$          ;BRANCH IF NO
11088
11089 041116 005037 042034      2$: CLR ID1
11090 041122 005037 042036      CLR ID2
11091 041126 011001                MOV (R0),R1     ;UPDATE PTP CONTENTS FOR INDIRECTLY
11092 041130 006301                ASL R1          ; SPECIFIED DATA DESCRIPTORS.
11093 041132 062701 004364      ADD #INSTID,R1
11094 041136 111137 042034      MOVB (R1),ID1
11095 041142 005201                INC R1
11096 041144 111137 042036      MOVB (R1),ID2  ;ID1 AND ID2 CONTAIN OFFSETS INTO PTP TABLE
11097 041150 005737 042034      TST ID1
11098 041154 001445                BEQ LDCOD      ;BRANCH IF OFFSET = 0. NO UPDATE REQUIRED
11099 041156 006337 042034      ASL ID1
11100 041162 062737 001566 042034  ADD #PTP,ID1   ;USE OFFSET TO GET PTP ENTRY
11101 041170 017701 000640      MOV @ID1,R1    ;USE ENTRY TO GET ADDRESS OF DATA DESCRIPTOR
11102 041174 011177 000634      MOV (R1),@ID1
11103 041200 005737 042036      TST ID2
11104 041204 001431                BEQ LDCOD      ;BRANCH IF SECOND OFFSET = 0
11105 041206 006337 042036      ASL ID2
11106 041212 062737 001566 042036  ADD #PTP,ID2   ;HANDLE SECOND OFFSET SAME AS FIRST
11107 041220 017701 000612      MOV @ID2,R1
11108 041224 011177 000606      MOV (R1),@ID2
11109 041230 000417                BR LDCOD
  
```

11110									
11111	041232					TYPE1:			: INPUT PARAMETERS SPECIFIED IN TABLES
11112	041232	010001							: SETUP PARAMETER TABLE POINTERS
11113	041234	062701	000004						: R1 POINTS TO IP1
11114	041240	012702	001570						: R2 POINTS TO TOP OF PARAMETER TABLE POINTER (PTP) LIST.
11115	041244	012112			1\$:	MOV (R1)+, (R2)			: LOAD PTP FROM IP
11116	041246	005712				TST (R2)			: PTP=0 ?
11117	041250	001402				BEQ 2\$			: YES - DON'T ADVANCE IT
11118	041252	062712	000002			ADD #2, (R2)			: ADVANCE PTP TO FIRST ENTRY
11119	041256	062702	000002		2\$:	ADD #2, R2			: UPDATE POINTER
11120	041262	020227	001640			CMP R2, #PTP24+2			: ALL PTP'S LOADED?
11121	041266	002766				BLT 1\$			: NO
11122									
11123	041270	011001			LDCOD.	MOV (R0), R1			: LOAD OCTAL CODING FOR CIS INST
11124	041272	010137	002276			MOV R1, OCTIC			: UNDER TEST INTO EINST & TINST
11125	041276	010137	002132			MOV R1, FATAL			: LOAD CODING FOR INST UNDER TEST INTO FATAL
11126									: ERROR INDICATOR WORD
11127	041302	006301				ASL R1			
11128	041304	062701	003726			ADD #OINST, R1			
11129	041310	011137	046136			MOV (R1), EINST			
11130	041314	011137	050012			MOV (R1), TINST			
11131	041320	004737	062616			JSR PC, SRNGSW			: SAVE STATE OF RANDOM # GEN. AS STATE W
11132									
11133	041324	004737	062642		NTC:	JSR PC, RRNGSW			: CONTROL IS PASSED TO THIS POINT TO EXECUTE
11134									: NEXT TEST CONDITION FOR GIVEN INPUT
11135									: TABLE. PARAMETER TABLE POINTERS
11136									: HAVE BEEN UPDATED TO POINT TO NEXT
11137									: TEST CONDITION PRIOR TO ENTRY
11138									: TO THIS POINT.
11139									: RESTORE RANDOM # GEN. TO STATE W.
11140	041330	013701	002276			MOV OCTIC, R1			: LOAD # OF INST DESC INTO NDESC
11141	041334	006301				ASL R1			: LOAD DATA TYPE CONTROL WORDS
11142	041336	006301				ASL R1			: PKPTW, ZPM, SXTYPE
11143	041340	063701	004262			ADD DECTYP, R1			
11144	041344	012137	002300			MOV (R1)+, TW1			
11145	041350	113737	002300	002454		MOVB TW1, PKPTW			: PKPTW IDENTIFIES STARTING DATA TYPE FOR EACH INST TESTI
11146	041356	113737	002301	002456		MOVB TW1+1, NDESC			
11147	041364	011137	002302			MOV (R1), TW2			
11148	041370	113701	002302			MOVB TW2, R1			
11149	041374	042701	177770			BIC #177770, R1			
11150	041400	010137	002442			MOV R1, S1TYPE			: S1TYPE IDENTIFIES 1ST STRING DESC DATA TYPE
11151	041404	113701	002302			MOVB TW2, R1			
11152	041410	042701	177707			BIC #177707, R1			
11153	041414	006201				ASR R1			
11154	041416	006201				ASR R1			
11155	041420	006201				ASR R1			
11156	041422	010137	002444			MOV R1, S2TYPE			
11157	041426	113701	002302			MOVB TW2, R1			
11158	041432	006301				ASL R1			
11159	041434	006301				ASL R1			
11160	041436	000301				SWAB R1			
11161	041440	010137	002446			MOV R1, S3TYPE			
11162	041444	113701	002303			MOVB TW2+1, R1			
11163	041450	006201				ASR R1			

11164	041452	010137	002460		MOV R1,ZPM	
11165	041456	005737	001760		TST RANDOM	; IN RANDOM EXERCISE MODE?
11166	041462	001447			BEQ NTCTS	; BRANCH IF NO
11167	041464	023727	002442	000006	CMP S1TYPE,#6	; RANDOMIZE STRING DATA TYPES
11168	041472	103405			BLO 1\$	
11169	041474	004737	063464		JSR PC,RPTYPE	; GET A RANDOM PACKED DATA TYPE
11170	041500	010037	002442		MOV R0,S1TYPE	; STORE IT IN S1TYPE
11171	041504	000404			BR 2\$	
11172	041506	004737	063512	1\$:	JSR PC,RZTYPE	; GET A RANDOM ZONED DATA TYPE
11173	041512	010037	002442		MOV R0,S1TYPE	; STORE IT IN S1TYPE
11174	041516	023727	002444	0C0006	2\$:	CMP S2TYPE,#6
11175	041524	103405			BLO 3\$	
11176	041526	004737	063464		JSR PC,RPTYPE	
11177	041532	010037	002444		MOV R0,S2TYPE	
11178	041536	000404			BR 4\$	
11179	041540	004737	063512	3\$:	JSR PC,RZTYPE	
11180	041544	010037	002444		MOV R0,S2TYPE	
11181	041550	023727	002446	000006	4\$:	CMP S3TYPE,#6
11182	041556	103405			BLO 5\$	
11183	041560	004737	063464		JSR PC,RPTYPE	
11184	041564	010037	002446		MOV R0,S3TYPE	
11185	041570	000404			BR NTCTS	
11186	041572	004737	063512	5\$:	JSR PC,RZTYPE	
11187	041576	010037	002446		MOV R0,S3TYPE	
11188						
11189	041602				NTCTS:	; CONTROL IS PASSED TO THIS POINT TO EXECUTE
11190						; THE GIVEN TEST CONDITION USING
11191						; THE NEXT DATA TYPE.
11192	041602	005237	001120		INC \$TESTN	; INCREMENT TEST # IN APT MAILBOX
11193	041606	005737	001760		TST RANDOM	; RUNNING IN RANDOM MODE?
11194	041612	001410			BEQ 2\$	; BRANCH IF NO
11195	041614	023737	001120	001652	CMP \$TESTN,TPERP	; HAS TEST # REACHED MAX PER APT PASS?
11196	041622	101404			BLOS 2\$	; BRANCH IF NO
11197	041624	005237	001122		INC \$PASS	; INCREMENT APT PASS COUNTER
11198	041630	005037	001120		CLR \$TESTN	
11199	041634	005237	001420	2\$:	INC TOTTC	; UPDATE TESTS EXECUTED COUNTER
11200	041640	001002			BNE 1\$	; BRANCH IF TEST COUNT NOT = ZERO
11201	041642	005237	001416		INC TOTTC	; INCREMENT TEST COUNT OVERFLOW EVERY TIME
11202						; TEST COUNT (TOTTC) EXCEEDS 177777 OCTAL.
11203	041646	013777	001420	140064	1\$:	MOV TOTTC,@DISPLAY
11204	041654	004737	062546		JSR PC,SRNGSY	; SET TEST # INTO DISPLAY LIGHTS
11205						; SAVE STATE OF RANDOM # GEN AS STATE Y.
11206	041660				RTC:	; CONTROL IS PASSED TO THIS POINT TO REPEAT
11207						; THE PREVIOUS TEST USING THE SAME
11208						; TEST CONDITION & DATA TYPES.
11209						; SETUP MICRO BREAK REG (11/74)
11210	041660	000240			NOP	; (11/74 MICROBREAK - REPLACE WITH 013737)
11211	041662	000240			NOP	; ( " " " " " " 177570)
11212	041664	000240			NOP	; ( " " " " " " 177770)
11213	041666	004737	062572		JSR PC,RRNGSY	; RESTORE RANDOM # GEN TO STATE Y.
11214	041672	005037	002140		CLR SPHAND	; CLEAR SPECIAL HANDLING REQUESTS
11215	041676	005037	002274		CLR FILLS2	
11216	041702	012737	111111	120400	MOV #111111,PRECSK	; INITIALIZE STACK OVERFLOW CONSTANTS
11217	041710	012737	111111	120602	MOV #111111,PSTCSK	

11218	041716	005037	002152		CLR ERRSTK		;CLEAR STACK ERROR FLAG.
11219	041722	012700	120402		MOV #PRECSK+2,R0		;INITIALIZE STACK CONTENTS BEFORE EACH TEST
11220	041726	020027	120602	1\$:	CMP R0,#PSTCSK		
11221	041732	001403			BEQ 2\$		
11222	041734	012720	055555		MOV #055555,(R0)+		
11223	041740	000772			BR 1\$		
11224	041742	013700	002134	2\$:	MOV INPTP,R0		;R0 POINTS TO ENTRY IN INPUT TABLE
11225	041746	011001			MOV (R0),R1		;SETUP POINTER TO PROPER CIS
11226	041750	006301			ASL R1		; INSTRUCTION FLOW TABLE
11227	041752	062701	001436		ADD #INO,R1		
11228	041756	011137	001742		MOV (R1),FLOPTR		
11229	041762	011001			MOV(R0),R1		;SETUP POINTER TO PROPER ERROR
11230	041764	006301			ASL R1		; MESSAGE HEADER
11231	041766	062701	070052		ADD #INEM,R1		
11232	041772	011137	002142		MOV (R1),EMPTR		
11233							
11234	041776			XINST:			
11235	041776	017701	137740		MOV @FLOPTR,R1		;GET NEXT ENTRY FROM INST. FLOW TABLE
11236	042002	006101			ROL R1		
11237	042004	006101			ROL R1		
11238	042006	006101			ROL R1		
11239	042010	006101			ROL R1		
11240	042012	006101			ROL R1		
11241	042014	042701	177760		BIC #177760,R1		;LOOK ONLY AT FLOW TABLE ENTRY COMMAND
11242	042020	006301			ASL R1		; FORM INDEX INTO FLOW DISPATCH TABLE
11243	042022	062701	001526		ADD #FLODIS,R1		
11244	042026	000171	000000		JMP @(R1)		;DISPATCH ON FLOW COMMAND
11245	042032	000000		MTYPE:	.WORD 0		
11246	042034	000000		ID1:	.WORD 0		
11247	042036	000000		ID2:	.WORD 0		
11248							
11249	042040	005737	002074	CTLC:	TST FSRUN		;DVT TYPE RUN?
11250	042044	001006			BNE NOCTC		;BRANCH IF NO
11251	042046	000137	037154		JMP DVTST		;RESTART DVT TYPE RUN
11252							
11253							
11254	042052			FC01:			;FLOW COMMAND = 01 - COPY TEST OPERAND
11255							; FROM PARAMETER TABLE INTO TRN.
11256	042052	004737	063232		JSR PC,EXTBK		;HANDLE OPERATOR REQUESTS
11257	042056	000137	042040		JMP CTLC		;CNTL C RETURN FROM SUBROUTINE.
11258	042062	032737	000100	050012 NGCTC:	BIT #100,TINST		;THE IN-LINE TEST CASE ALWAYS FOLLOWS THE SAME
11259	042070	001067			BNE FCRTN		;REG TEST CASE - THEREFORE DO NOT REFILL THE TRNS
11260							; BECAUSE SXTYPES ARE ALREADY UPDATED FOR NEXT REG
11261							; TEST CONDITION.
11262	042072	004737	053570		JSR PC,PF1		;FORM PARAMETER TABLE POINTER FROM
11263							; PF1 FIELD OF FLOW TABLE ENTRY
11264	042076	004737	054004		JSR PC,RF4		;FORM TEST OPERAND POINTER FROM RF4 FIELD
11265							; OF FLOW TABLE ENTRY.
11266	042102	017711	140030		MOV @PTPTR,(R1)		;COPY TEST OPERAND FROM PARAMETER TABLE
11267	042106	010102			MOV R1,R2		
11268	042110	004737	053766		JSR PC,RF3X		;LOAD R1 WITH TWICE CONTENTS OF RF3 FIELD OF
11269	042114	005701			TST R1		; FLOW TABLE ENTRY
11270	042116	001454			BEQ FCRTN		;BRANCH IF FIELD CONTAINS ZERO
11271	042120	012737	177777	002440	MOV #177777,DECINS		;SET FLAG TO INDICATE THAT INST IS A DECIMAL INST.



11272	042126	020127	000002			CMP R1,#2			;IS TEST OPERAND PART OF 1ST DECIMAL DESC
11273									; OPERAND FOR INST?
11274	042132	001013				BNE 1\$			;BRANCH IF NO
11275	042134	013704	002442			MOV S1TYPE,R4			
11276	042140	006304				ASL R4			
11277	042142	006304				ASL R4			
11278	042144	006304				ASL R4			
11279	042146	006304				ASL R4			
11280	042150	000304				SWAB R4			
11281	042152	042704	107777			BIC #107777,R4			
11282	042156	050412				BIS R4,(R2)			; 'OR' SRC1 TYPE FIELD INTO TEST OPERAND
11283	042160	000433				BR FCRTN			
11284	042162	020127	000004	1\$:		CMP R1,#4			;IS TEST OPERAND PART OF 2ND DECIMAL DESC
11285									; OPERAND FOR INST?
11286	042166	001013				BNE 2\$			;BRANCH IF NO
11287	042170	013704	002444			MOV S2TYPE,R4			
11288	042174	006304				ASL R4			
11289	042176	006304				ASL R4			
11290	042200	006304				ASL R4			
11291	042202	006304				ASL R4			
11292	042204	000304				SWAB R4			
11293	042206	042704	107777			BIC #107777,R4			
11294	042212	050412				BIS R4,(R2)			; 'OR' SRC2 TYPE FIELD INTO TEST OPERAND
11295	042214	000415				BR FCRTN			
11296	042216	020127	000006	2\$:		CMP R1,#6			;IS TEST OPERAND PART OF 3RD DECIMAL DESC
11297									; OPERAND FOR INST?
11298	042222	001012				BNE FCRTN			;BRANCH IF NO
11299	042224	013704	002446			MOV S3TYPE,R4			
11300	042230	006304				ASL R4			
11301	042232	006304				ASL R4			
11302	042234	006304				ASL R4			
11303	042236	006304				ASL R4			
11304	042240	000304				SWAB R4			
11305	042242	042704	107777			BIC #107777,R4			; 'OR' DST TYPE FIELD INTO TEST OPERAND
11306	042246	050412				BIS R4,(R2)			
11307									
11308	042250	062737	000002	001742	FCRTN:	ADD #2,FLOPTR			;UPDATE FLOW TABLE POINTER TO NEXT COMMAND
11309	042256	000137	041776			JMP XINST			
11310									
11311	042262				FC02:				;FLOW COMMAND = 02 - GENERATE TEST OPERAND
11312									; FROM PARAMETER TABLE ENTRY.
11313	042262	032737	000100	050012		BIT #100,TINST			;SKIP THIS FLOW COMMAND FOR INLINE CASE
11314	042270	001367				BNE FCRTN			
11315	042272	005737	042032			TST MTYPE			
11316	042276	001410				BEQ 1\$			;BRANCH IF NOT TYPE 0 ENTRY
11317	042300	004737	053570			JSR PC,PF1			;FORM PARAMETER TABLE POINTER FROM
11318									; PF1 FIELD OF FLOW TABLE ENTRY.
11319	042304	004737	054004			JSR PC,RF4			;FORM TEST OPERAND POINTER FROM
11320									; RF4 FIELD OF FLOW TABLE ENTRY
11321	042310	017711	137622			MOV @PTPTR,(R1)			;COPY TEST OPERAND FROM PARAMETER TABLE
11322	042314	000137	042250			JMP FCRTN			
11323	042320	004737	053570	1\$:		JSR PC,PF1			;FORM PARAMETER TABLE POINTER FROM PF1
11324									; FIELD OF FLOW TABLE ENTRY.
11325	042324	004777	137606			JSR PC,@PTPTR			;EXFCUTE PARAMETER TABLE ENTRY TO

11326									: GENERATE TEST OPERAND.
11327	042330	000137	042250			JMP FCRTN			
11328									
11329	042334	005237	001424			REDNTC: INC REDTC			:TEST CONDITION REDUNDANT - ABORT TEST.
11330	042340	005737	001420			TST TOTTC			:DID TEST COUNT OVERFLOW ON LAST INCREMENT?
11331	042344	001002				BNE 1\$			:BRANCH IF NO
11332	042346	005337	001416			DEC TOTTC			
11333	042352	005337	001420			1\$: DEC TOTTC			
11334	042356	000137	052654			JMP N\$TTC			
11335									
11336	042362					FC03:			:FLOW COMMAND = 03 - VERIFY THAT STRING'S
11337									: LOWER ADDRESS LIMIT FALLS WITHIN TEST BUFFER.
11338	042362	032737	000100	050012		BIT #100,TINST			:SKIP THIS FLOW COMMAND FOR INLINE CASE
11339	042370	001327				BNE FCRTN			
11340	042372	032737	000010	002140		BIT #10,SPHAND			:SPECIAL HANDLING REQUEST?
11341	042400	001323				BNE FCRTN			:SKIP THIS FLOW COMMAND IF YES.
11342	042402	004737	053570			JSR PC,PF1			:FORM PARAMETER TABLE POINTER TO STRING
11343									: SURROUND LENGTH FROM PF1 FIELD OF FLOW
11344									: TABLE ENTRY.
11345	042406	004737	054004			JSR PC,RF4			:FORM TEST OPERAND POINTER TO STRING.ADR FROM
11346									: RF4 FIELD OF FLOW TABLE ENTRY
11347	042412	011101				MOV (R1),R1			
11348	042414	167701	137516			SUB @PTPTR,R1			:SUBTRACT STRING.SURR.LEN FROM STRING.ADR
11349									: TO GET STRING.SURR.ADR.
11350	042420	020127	000020			CMP R1,#20			
11351	042424	002475				BLT 1TC			:STRING.SURR.ADR < 20 -SKIP THIS TEST CONDITION
11352									: (20 ALLOWS SPACE FOR IN-LINE DESCRIPTORS AT BEGINNING
11353	042426	000137	042250			JMP FCRTN			
11354									
11355	042432					FC04:			:FLOW COMMAND = 04 - VERIFY THAT STRINGS
11356									: UPPER ADDRESS LIMIT FALLS WITHIN TEST
11357									: BUFFER.
11358	042432	032737	000100	050012		BIT #100,TINST			:SKIP THIS FLOW COMMAND FOR INLINE CASE
11359	042440	001035				BNE 1\$			
11360	042442	032737	000001	002140		BIT #1,SPHAND			:SPECIAL HANDLING REQUEST
11361	042450	001031				BNE 1\$			:SKIP VERIFICATION IF YES
11362	042452	004737	053570			JSR PC,PF1			:FORM PARAMETER TABLE POINTER TO STRING.SURR.LEN
11363									: FROM PF1 FIELD OF FLOW TABLE ENTRY
11364	042456	004737	053754			JSR PC,RF3			:FORM TEST OPERAND POINTER TO STRING.LEN
11365									: FROM RF3 FIELD OF FLOW TABLE ENTRY
11366	042462	011101				MOV (R1),R1			
11367	042464	005737	002440			TST DECINS			:IS INST A DECIMAL INST?
11368	042470	001402				BEQ 2\$			:BRANCH IF NO
11369	042472	043701	002450			BIC TYPFLD,R1			:CLEAR TYPE FIELD SO AS NOT TO
11370									: DISTORT UPPER ADDRESS CALCULATION.
11371	042476	017702	137434			2\$: MOV @PTPTR,R2			:R2 NOW CONTAINS SUM OF STRING.SURR.LEN
11372	042502	060102				ADD R1,R2			: AND STRING.LEN
11373									: FORM TEST OPERAND POINTER TO STRING.ADR
11374	042504	004737	054004			JSR PC,RF4			: FROM RF4 FIELD OF FLOW TABLE ENTRY.
11375									: R2 NOW CONTAINS STRING.ADR + STRING.LEN
11376	042510	061102				ADD (R1),R2			: + STRING.SURR.ADR
11377									:RANDOM EXERCISE MODE?
11378	042512	005737	0C1760			TST RANDOM			:BRANCH IF NO.
11379	042516	001403				BEQ 3\$			

11380	042520	020237	001644			CMP R2,RTBLEN	;COMPARE ADDRESS WITH END OF RANDOM TEST BUFFER
11381	042524	000402				BR 4\$	
11382	042526	020237	001642	3\$:		CMP R2,TBLEN	;DOES THIS ADDRESS EXCEED TEST BUFFER
11383							; LENGTH ?
11384	042532	003032		4\$:		BGT NXTC	;YES - SKIP THIS TEST CONDITION
11385	042534	000137	042250	1\$:		JMP FCRTN	
11386							
11387	042540			FC05:			;FLOW COMMAND = 05 - ADJUST TEST OPERANDS
11388							; TO INCLUDE BASE ADDRESS OF TEST BUFFER.
11389	042540	032737	000100	050012		BIT #100,TINST	;SKIP THIS FLOW COMMAND FOR INLINE CASE
11390	042546	001022				BNE EC05	
11391	042550	004737	053666			JSR PC,RF1	;FORM 1ST TEST OPERAND POINTER FROM RF1
11392							; FIELD OF FLOW TABLE ENTRY
11393	042554	063711	001640			ADD TBADR,(R1)	;ADD TEST BUFFER BASE ADDRESS TO OPERAND
11394	042560	004737	053720			JSR PC,RF2	;FORM 2ND TEST OPERAND POINTER FROM RF2
11395							; FIELD OF FLOW TABLE ENTRY
11396	042564	020127	003630			CMP R1,#TRN	;IF R1 STILL POINTS TO #TRN THEN THERE WAS
11397							; ONLY ONE TEST OPERAND TO BE UPDATED
11398	042570	001411				BEQ EC05	;UPDATING COMPLETE
11399	042572	063711	001640			ADD TBADR,(R1)	;ADD TEST BUFFER BASE ADDRESS TO OPERAND
11400	042576	004737	053754			JSR PC,RF3	;FORM 3RD TEST OPERAND POINTER FROM RF3
11401							; FIELD OF FLOW TABLE ENTRY.
11402	042602	020127	003630			CMP R1,#TRN	;WAS THERE A THIRD ENTRY?
11403	042606	001402				BEQ EC05	;NO - UPDATING COMPLETE
11404	042610	063711	001640			ADD TBADR,(R1)	;ADD TEST BUFFER BASE ADDRESS TO OPERAND
11405	042614	000137	042250	EC05:		JMP FCRTN	
11406							
11407	042620	005237	001422	NXTC:		INC INVTC	;TEST CONDITION INVALID - ABORT TEST
11408	042624	005737	001420			TST TOTTC	;DID TEST COUNT OVERFLOW ON LAST INCREMENT?
11409	042630	001002				BNE 1\$	;BRANCH IF NO
11410	042632	005337	001416			DEC TOTTC	
11411	042636	005337	001420	1\$:		DEC TOTTC	
11412	042642	000137	052654			JMP NXTC	
11413							
11414	042646			FC06:			;FLOW COMMAND = 06 - INITIALIZE TEST BUFFER
11415							; TO AND INCREMENTING SEQUENCE.
11416	042646	032737	000100	050012		BIT #100,TINST	
11417	042654	001402				BEQ 4\$	
11418	042656	004737	062712			JSR PC,RRNGSV	;RESTORE RANDOM # GEN TO STATE V
11419	042662	004737	062666	4\$:		JSR PC,SRNGSV	;SAVE RANDOM # GENERATOR STATE AS STATE V
11420	042666	013737	001640	001754		MOV TBADR,TBEND	
11421	042674	005737	001760			TST RANDOM	;RANDOM EXERCISE MODE?
11422	042700	001404				BEQ 2\$	;BRANCH IF NO
11423	042702	063737	001644	001754		ADD TBLEN,TBEND	;THE BUFFER SIZE FOR RANDOM EXERCISE MODE
11424							; IS FIXED
11425	042710	000403				BR 3\$	
11426	042712	063737	001642	001754	2\$:	ADD TBLEN,TBEND	;SETUP A POINTER TO END OF BUFFER
11427	042720	013701	001640	3\$:		MOV TBADR,R1	
11428	042724	013702	001640			MOV TBADR,R2	;POINT R1 & R2 TO START OF BUFFER
11429	042730	013721	001660			MOV INCSQ1,(R1)+	
11430	042734	013721	001662			MOV INCSQ2,(R1)+	;LOAD THE FIRST TWO BUFFER LOCATIONS
11431	042740	012211		1\$:		MOV (R2)+,(R1)	
11432	042742	061221				ADD (R2),(R1)+	;CONTENTS OF NEXT LOC = SUM OF CONTENTS
11433							; OF PREVIOUS 2 LOCATIONS.

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
CIS INSTRUCTION TEST LOOP

B 13  
PAGE 88-15 SEQUENCE 157

11434 042744 023701 001754  
11435 042750 003373  
11436 042752 000137 042250

CMP TBEND,R1  
BGT 1\$  
JMP FCRTN

11438	042756					FC07:		;FLOW COMMAND = 07 - INSERT STRING IN
11439								; TEST BUFFER.
11440	042756	005037	002476			CLR SAVSRF		
11441	042762	005737	001760			TST RANDOM		;RANDOM EXERCISE MODE?
11442	042766	001406				BEQ 1\$		;BRANCH IF NO
11443	042770	023727	002276	000011		CMP OCTIC,#11		;IS INST A CHAR STRING INST?
11444	042776	101002				BHI 1\$		;BRANCH IF NO
11445	043000	000137	042250			JMP FCRTN	2\$:	;IN RANDOM MODE, NO CHAR STRINGS NEED TO BE
11446								; INSERTED FOR CHAR TYPE INSTRUCTIONS.
11447								; RANDOM CHAR BYTES ARE DERIVED BY
11448								; RANDOMIZING THE 'SEED' CONSTANTS USED
11449								; TO INITIALIZE THE ENTIRE BUFFER.
11450	043004	032737	000020	002140	1\$:	BIT #20,SPHAND		;SPECIAL HANDLING REQUEST?
11451	043012	001372				BNE 2\$		;SKIP INSERTING STRINGS IF YES
11452	043014	004737	053570			JSR PC,PF1		;FORM PARAMETER TABLE POINTER TO STRING
11453								; DESCRIPTOR FROM PF1 FIELD OF FLOW TABLE ENTRY
11454	043020	004737	053754			JSR PC,RF3		;FORM TEST OPERAND POINTER TO STRING.LEN
11455								; FROM RF3 FIELD OF FLOW TABLE ENTRY
11456	043024	010137	002104			MOV R1,TRL		;SAVE POINTER
11457	043030	004737	054004			JSR PC,RF4		;FORM TEST OPERAND POINTER TO STRING.ADR
11458								; FROM RF4 FIELD OF FLOW
11459								; TABLE ENTRY.
11460	043034	010137	002102			MOV R1,TRA		;SAVE POINTER
11461	043040	004737	062476			JSR PC,SRNGSX		;SAVE RANDOM NUMBER GEN. STATE X
11462	043044	004537	043332			JSR R5,ISTG		;INSERT STRING IN TEST BUFFER
11463	043050	002136				PTPTR		;POINTER TO STRING DESCRIPTOR
11464	043052	002104				TRL		;POINTER TO STRING.LEN
11465	043054	002102				TRA		;POINTER TO STRING.ADR
11466								
11467	043056	005737	002476			TST SAVSRF		;SAVE STRING FOR ERROR PRINTOUT?
11468	043062	001436				BEQ A3X		
11469	043064	004737	053570			JSR PC,PF1		;YES - RESTORE POINTER TO STRING DESCRIPTOR
11470	043070	004737	053754			JSR PC,RF3		;RESTORE POINTER TO STRING LENGTH
11471	043074	010137	002104			MOV R1,TRL		
11472	043100	005737	002274			TST FILLS2		;WHERE SHOULD STRING BE STORED?
11473	043104	001406				BEQ 4\$		
11474	043106	012737	002506	002474		MOV #INSRC2+2,SAVPTR		;STORE STRING IN BUFSR?
11475	043114	011137	002504			MOV (R1),INSRC2		;SAVE STRING LEN IN BUFFER DESCRIPTOR
11476	043120	000410				BR 5\$		
11477	043122	012737	002502	002474	4\$:	MOV #INSRC1+2,SAVPTR		;SAVE STRING IN BUFSR1
11478	043130	012737	177777	002274		MOV #177777,FILLS2		;SIGNAL THAT BUFSR1 IS OCCUPIED.
11479	043136	011137	002500			MOV (R1),INSRC1		;SAVE STRING LEN IN BUFFER DESCRIPTOR
11480	043142	004737	062522			JSR PC,RRNGSX	5\$:	;RESTORE RANDOM NUMBER GEN TO STATE X.
11481	043146	004537	043332			JSR R5,ISTG		;INSERT STRING IN SAVE BUFFER
11482	043152	002136				PTPTR		;POINTER TO STRING DESCRIPTOR
11483	043154	002104				TRL		;POINTER TO STRING LEN
11484	043156	002474				SAVPTR		;POINTER TO STRING ADDRESS (EITHER BUFSR1 OR BUFSR2)
11485								
11486	043160	062737	000002	001742	A3X:	ADD #2,FLOPTR		;LOOK AT NEXT FLOW COMMAND?
11487	043166	017701	136550			MOV @FLOPTR,R1		
11488	043172	042701	017777			BIC #017777,R1		
11489	043176	005701				TST R1		;IS IT = 0 - A CONTINUATION OF THE 07
11490								; COMMAND?
11491	043200	001050				BNE 2\$		;NO - DON'T INSERT SURROUND STRINGS

11492	043202	004737	053570			JSR PC,PF1		;YES - FORM PARAMETER TABLE POINTER
11493								; TO STRING.SURR.LEN
11494	043206	013737	002136	002046		MOV PIPTR,SURLEN		
11495	043214	004737	053642			JSR PC,PF2		;FORM PARAMETER TABLE POINTER TO SURR DATA
11496								; DESCRIPTOR
11497	043220	017737	136660	002104		MOV @TRL,TRL		;FORM SURR.ADR (UPPER PORTION)
11498	043226	005737	002440			TST DECINS		;INST = DECIMAL?
11499	043232	001403				BEQ 3\$		;BRANCH IF NO
11500	043234	042737	070000	002104		BIC #070000,TRL		;CLEAR TYPE FIELD FROM STRING LENGTH
11501	043242	067737	136634	002104	3\$:	ADD @TRA,TRL		;SURR.ADR = STRING.ADR + STRING.LEN
11502	043250	012737	002104	002050		MOV #TRL,SURADR		
11503	043256	004537	043332			JSR R5,ISTG		;INSERT UPPER HALF OF SURR STRING
11504	043262	002136				PTPTR		;POINTER TO SURR.DATA DESCRIPTOR
11505	043264	002046				SURLEN		;POINTER TO SURR.LEN
11506	043266	002050				SURADR		;POINTER TO SURR.ADR
11507	043270	017737	136606	002104		MOV @TRA,TRL		;FORM SURR STRING ADDRESS (LOWER PORTION)
11508	043276	167737	136544	002104		SUB @SURLEN,TRL		;SURR.ADR = STRING.ADR - SURR.LEN
11509	043304	004537	043332			JSR R5,ISTG		;INSERT LOWER HALF OF SURR STRING
11510	043310	002136				PTPTR		
11511	043312	002046				SURLEN		
11512	043314	002050				SURADR		
11513	043316	000137	042250		1\$:	JMP FCRTN		
11514	043322	162737	000002	001742	2\$:	SUB #2,FLOPTR		;RESTORE FLOW COMMAND POINTER
11515	043330	000772				BR 1\$		
11516								
11517	043332					ISTG:		;SUBROUTINE TO INSERT STRING IN TEST BUFFER.
11518	043332	013501				MOV @(R5)+,R1		
11519	043334	012137	002106			MOV (R1)+,STGDS1		;GET STRING DATA DESCRIPTOR
11520	043340	011137	002110			MOV (R1),STGDS2		
11521	043344	013501				MOV @(R5)+,R1		;GET STRING.LEN
11522	043346	011137	002112			MOV (R1),STGLN		
11523	043352	032737	000002	002140		BIT #2,SPHAND		;SPECIAL HANDLING REQUEST?
11524	043360	001403				BEQ 1\$		;BRANCH IF NO
11525	043362	042737	100000	002112		BIC #100000,STGLN		;YES - STRIP BIT 15 FROM LENGTH
11526	043370	013501			1\$:	MOV @(R5)+,R1		;SETUP STRING STARTING ADDRESS
11527	043372	011137	002114			MOV (R1),STGAD		
11528	043376	013701	002106			MOV STGDS1,R1		
11529	043402	042737	160000	002106		BIC #160000,STGDS1		;STRIP OFF 'TYPE' FROM 1ST WORD OF DATA DESCRIPTOR
11530	043410	042701	017777			BIC #17777,R1		;LOOK ONLY AT DESCRIPTOR TYPE
11531	043414	005701				TST R1		
11532	043416	001426				BEQ DSTYP0		;DATA DESCRIPTOR IS TYPE 0.
11533	043420	022701	020000			CMP #020000,R1		
11534	043424	001445				BEQ DSTYP1		;DATA DESCRIPTOR IS TYPE 1
11535	043426	022701	040000			CMP #040000,R1		
11536	043432	001461				BEQ DSTYP2		;DATA DESCRIPTOR IS TYPE 2
11537	043434	022701	060000			CMP #060000,R1		
11538	043440	001516				BEQ DSTYP3		;DATA DESCRIPTOR IS TYPE 3
11539	043442	022701	100000			CMP #100000,R1		
11540	043446	001404				BEQ 2\$		;DATA DESCRIPTOR IS TYPE 4
11541	043450	022701	120000			CMP #120000,R1		
11542	043454	001405				BEQ 3\$		;DATA DESCRIPTOR IS TYPE 5
11543	043456	000000				HALT		;**DATA DESCRIPTOR NOT TYPE 0,1,2,3,4, OR 5.
11544	043460	005037	001770		2\$:	CLR RANDTA		
11545	043464	000137	044574			JMP DSTYP4		

11546	043470	000137	045642		3\$:	JMP DSTYP5	
11547							
11548	043474				DSTYP0:		: ALL BYTES OF STRING ARE IDENTICAL
11549	043474	005737	002440			TST DECINS	: INST = DECIMAL?
11550	043500	001403				BEQ 3\$	: BRANCH IF NO
11551	043502	042737	070000	002112		BIC #070000,STGLN	: CLEAR TYPE FIELD STRING LENGTH WORD
11552	043510	013701	002114		3\$:	MOV STGAD,R1	: R1 CONTAINS STRING STARTING ADDRESS
11553	043514	005737	002112		1\$:	TST STGLN	: ENTIRE STRING BEEN INSERTED?
11554	043520	001405				BEQ 2\$	: YES
11555	043522	113721	002106			MOVB STGDS1,(R1)+	: NO - MOVE STRING DATA BYTE INTO NEXT
11556							: TEST BUFFER LOCATION
11557	043526	005337	002112			DEC STGLN	
11558	043532	000770				BR 1\$	
11559	043534	000137	045740		2\$:	JMP EISTG	
11560							
11561	043540				DSTYP1:		: STRING BYTE N = STRING BYTE N-1 + INC.
11562	043540	013701	002114			MOV STGAD,R1	: R1 CONTAINS STRING STARTING ADDRESS
11563	043544	005737	002112		1\$:	TST STGLN	: ENTIRE STRING BEEN INSERTED?
11564	043550	001410				BEQ 2\$	: YES
11565	043552	113721	002110			MOVB STGDS2,(R1)+	
11566	043556	063737	002106	002110		ADD STGDS1,STGDS2	: NO FORM AND INSERT NEXT STRING BYTE
11567	043564	005337	002112			DEC STGLN	
11568	043570	000765				BR 1\$	
11569	043572	000137	045740		2\$:	JMP EISTG	
11570							
11571	043576				DSTYP2:		: INSERT BYTES FROM GIVEN STRING
11572	043576	013701	002114			MOV STGAD,R1	: SET R1 TO STARTING ADDRESS OF STRING TO BE
11573							: FORMED IN TEST BUFFER.
11574	043602	005737	002106			TST STGDS1	: IS GIVEN STRING LENGTH = 0?
11575	043606	001431				BEQ 3\$	: YES - DON'T DO ANY INSERTING
11576	043610	013737	002106	002116		MOV STGDS1,SAVSL	: NO - SAVE STRING LENGTH IN CASE STRING
11577	043616	013737	002110	002120		MOV STGDS2,SAVSA	: TO BE FORMED IS LONGER THAN GIVEN STRING.
11578	043624	005737	002112		1\$:	TST STGLN	: ENTIRE STRING BEEN INSERTED?
11579	043630	001420				BEQ 3\$	: YES
11580	043632	117721	136252			MOVB @STGDS2,(R1)+	: NO - INSERT STRING BYTE FROM GIVEN STRING
11581	043636	005237	002110			INC STGDS2	: UPDATE GIVEN STRING ADDRESS TO NEXT BYTE
11582	043642	005337	002106			DEC STGDS1	
11583	043646	001006				BNE 2\$	: ALL BYTES IN GIVEN STRING USED?
11584	043650	013737	002116	002106		MOV SAVSL,STGDS1	: YES - STRING BEING FORMED IS LONGER THAN
11585	043656	013737	002120	002110		MOV SAVSA,STGDS2	: GIVEN STRING. RESET STRING ADDRESS
11586							: BACK TO BEGINNING OF GIVEN STRING.
11587	043664	005337	002112		2\$:	DEC STGLN	: DECREMENT COUNT OF # OF CHARACTERS YET
11588							: TO BE INSERTED.
11589	043670	000755				BR 1\$	
11590	043672	000137	045740		3\$:	JMP EISTG	
11591							
11592	043676				DSTYP3:		: DECIMAL STRING - ALL DIGITS IDENTICAL
11593	043676	022737	010000	002106		CMP #10000,STGDS1	: IS FORMED STRING TO BE PACKED OR ZONED DECIMAL?
11594	043704	101070				BHI PTYP3P	
11595	043706				TYP3Z:		: ZONED
11596	043706	013701	002112			MOV STGLN,R1	: GET STRING DESC. TYPE FIELD
11597	043712	000301				SWAB R1	
11598	043714	006201				ASR R1	
11599	043716	006201				ASR R1	

11600	043720	006201		ASR R1	
11601	043722	006201		ASR R1	
11602	043724	042701	177770	BIC #177770,R1	
11603	043730	010137	002310	MOV R1,STGTYP	;SAVE TYPE
11604	043734	042737	177740	BIC #177740,STGLN	;STRIP TYPE OFF STRING LENGTH WORD
11605	043742	013737	002106	MOV STGDS1,SIGN	;STRIP OFF SIGN FROM DATA DESCRIPTOR WORD 1
11606	043750	042737	177760	BIC #177760,SIGN	
11607	043756	006237	002106	ASR STGDS1	;GET AND RIGHT ADJUST ZONED DATA BYTE
11608	043762	006237	002106	ASR STGDS1	
11609	043766	006237	002106	ASR STGDS1	
11610	043772	006237	002106	ASR STGDS1	
11611	043776	042737	177400	BIC #177400,STGDS1	;GOT ZONED DATA BYTE (HIGH NIBBLE & DIGIT)
11612					; RIGHT ADJUSTED IN STGDS1.
11613	044004	013702	002106	MOV STGDS1,R2	
11614	044010	042702	177760	BIC #177760,R2	
11615	044014	010237	002306	MOV R2,STGDIG	;SAVE JUST THE DIGIT IN STGDIG
11616	044020	005737	002112	TST STGLN	;STRING TO BE FORMED HAVE 0 LENGTH?
11617	044024	001535		BEQ TFS	;YES - NOTE: A ZERO LENGTH ZONED
11618					; STRING OCCUPIES NO MEMORY (ESCEPT SEPARATE TYPES).
11619	044026	013701	002114	MOV STGAD,R1	;SET R1 TO STARTING ADDRESS OF STRING
11620					; TO BE FORMED IN TEST BUFFER.
11621	044032	113721	002106	1\$: MOVSB STGDS1,(R1)+	;NO - INSERT NEXT ZONED DATA BYTE.
11622	044036	005337	002112	DEC STGLN	
11623	044042	005737	002112	TST STGLN	;ENTIRE STRING BEEN INSERTED?
11624	044046	001371		BNE 1\$	;NO
11625	044050	013702	002310	MOV STGTYP,R2	;YES - INSERT SIGN BYTE
11626	044054	006302		ASL R2	
11627	044056	062702	002312	ADD #TYPTAB,R2	
11628	044062	000172	000000	JMP @ (R2)	;VECTOR TO APPROPRIATE STRING TYPE ROUTINE
11629					; TO ENTER SIGN BYTE.
11630					
11631	044066	000137	044354	PTYP3P: JMP TYP3P	
11632					
11633	044072	142741	000360	TYP3Z: BICB #360,-(R1)	;SIGNED ZONED
11634					;CLEAR OUT THE HIGH NIBBLE OF LEAST
11635					; SIGNIFICANT STRING BYTE.
11636	044076	006337	002124	ASL SIGN	
11637	044102	006337	002124	ASL SIGN	
11638	044106	006337	002124	ASL SIGN	
11639	044112	006337	002124	ASL SIGN	
11640	044116	153711	002124	BISB SIGN,(R1)	; 'OR' IN SIGN
11641	044122	000512		BR EXTYP	
11642	044124	000511		TYP4Z: BR EXTYP	;UNSIGNED ZONED - NO ACTION REQUIRED
11643					
11644	044126			TYPTO:	;TRAILING OVERPUNCHED
11645	044126	022737	000003	CMP #3,SIGN	;IS SIGN = +?
11646	044134	001416		BEQ 1\$	;BRANCH IF YES
11647	044136	004737	063532	JSR PC,RN	;RANDOMLY SELECT FROM 2 NEGATIVE SIGN TABLES
11648	044142	032700	000001	BIT #1,R0	
11649	044146	001403		BEQ 3\$	
11650	044150	012702	002346	MOV #NEGTB1,R2	
11651	044154	000402		BR 2\$	
11652	044156	012702	002334	3\$: MOV #NEGTAB,R2	;SIGN IS NEGATIVE
11653	044162	063702	002306	2\$: ADD STGDIG,R2	



11654	044166	111241			MOVB (R2),-(R1)		;COPY ENCODED SIGN FROM TABLE INTO STRING
11655	044170	000467			BR EXTYP		
11656	044172	004737	063532		1\$: JSR PC,RN		;RANDOMLY SELECT FROM 3 POSITIVE SIGN TABLES.
11657	044176	032700	000001		BIT #1,R0		
11658	044202	001403			BEQ 4\$		
11659	044204	012702	002372		MOV #POSTB1,R2		
11660	044210	000764			BR 2\$		
11661	044212	032700	000002		4\$: BIT #2,R0		
11662	044216	001403			BEQ 5\$		
11663	044220	012702	002404		MOV #POSTB2,R2		
11664	044224	000756			BR 2\$		
11665	044226	012702	002360		5\$: MOV #POSTAB,R2		
11666	044232	000753			BR 2\$		
11667							
11668	044234				TYPLO:		;LEADING OVERPUNCHED
11669	044234	013701	002114		MOV STGAD,R1		;SETUP POINTER TO MOST SIGN. BYTE OF STRING
11670	044240	005201			INC R1		
11671	044242	000731			BR TYPTO		
11672							
11673	044244				TYPTS:		;TRAILING SEPARATE
11674	044244	022737	000003	002124	CMP #3,SIGN		;IS SIGN +
11675	044252	001403			BEQ 1\$		;BRANCH IF YES
11676	044254	112711	000055		MOVB #055,(R1)		;SIGN = -; COPY SIGN BYTE INTO STRING
11677	044260	000433			BR EXTYP		
11678	044262	004737	063532		1\$: JSR PC,RN		;RANDOMLY SELECT BETWEEN 2 POSITIVE SEPARATE SIGNS.
11679	044266	032700	000001		BIT #1,R0		
11680	044272	001403			BEQ 2\$		
11681	044274	112711	000040		MOVB #040,(R1)		
11682	044300	000423			BR EXTYP		
11683	044302	112711	000053		2\$: MOVB #053,(R1)		;SIGN = +; COPY SIGN BYTE INTO BYTE STRING
11684	044306	000420			BR EXTYP		
11685							
11686	044310				TYPLS:		;LEADING SEPARATE
11687	044310	013701	002114		MOV STGAD,R1		;SETUP POINTER TO BYTE BEFORE MOST SIGN.
11688							; DIGIT OF STRING.
11689	044314	005301			DEC R1		
11690	044316	000752			BR TYPTS		
11691							
11692	044320				TFS:		;0 LENGTH STRING
11693	044320	022737	000005	002310	CMP #5,STGTYP		;IS STRING TYPE = LEADING SEPARATE?
11694	04432	001001			BNE 1\$		;BRANCH IF NO
11695	044336	000767			BR TYPLS		;INSERT SIGN AT 'A-1'
11696	044332	022737	000004	002310	1\$: CMP #4,STGTYP		;IS STRING TYPE = TRAILING SEPARATE
11697	044340	001003			BNE EXTYP		;BRANCH IF NO
11698	044342	013701	002114		MOV STGAD,R1		;INSERT SIGN AT 'A'
11699	044346	000736			BR TYPTS		
11700	044350	000137	045740		EXTYP: JMP EISTG		
11701							
11702	044354				TYP3P:		;PACKED
11703	044354	013701	002112		MOV STGLN,R1		;GET STRING DESC. TYPE FIELD
11704	044360	000301			SWAB R1		
11705	044362	006201			ASR R1		
11706	044364	006201			ASR R1		
11707	044366	006201			ASR R1		

11708	044370	006201				ASR R1	
11709	044372	042701	177770			BIC #177770,R1	
11710	044376	010137	002310			MOV R1,STGTYP	;SAVE TYPE
11711	044402	042737	177740	002112		BIC #177740,STGLN	;STRIP TYPE OFF STRING LENGTH WORD
11712	044410	113737	002106	002124		MOVB STGDS1,SIGN	;LOAD SIGN WITH DIGIT-SIGN BYTE
11713	044416	113701	002106			MOVB STGDS1,R1	;FORM DIGIT-DIGIT BYTE
11714	044422	006201				ASR R1	;RIGHT ADJUST DIGIT
11715	044424	006201				ASR R1	
11716	044426	006201				ASR R1	
11717	044430	006201				ASR R1	
11718	044432	042701	177760			BIC #177760,R1	;CLEAR OUT ALL BUT DIGIT
11719	044436	042737	177417	002106		BIC #177417,STGDS1	;CLEAR OUT ALL BUT DIGIT IN DATA DESCRIPTOR WORD 1
11720	044444	050137	002106			BIS R1,STGDS1	;STGDS1 NOW CONTAINS DIGIT-DIGIT BYTE
11721	044450	110137	002110			MOVB R1,STGDS2	;SAVE 0000-DIGIT BYTE IN CASE STRING LENGTH
11722							; IS EVEN - I.E. MOST SIGNIF DIGIT BYTE CONTAINS
11723							; ONLY A SINGLE DIGIT.
11724	044454	013701	002114			MOV STGAD,R1	;SET R1 TO STARTING ADDRESS OF STRING
11725							; TO BE FORMED IN TEST BUFFER.
11726	044460	005737	002112			TST STGLN	;STRING TO BE FORMED HAVE ZERO LENGTH?
11727	044464	001004				BNE 3\$	;NO
11728	044466	042737	177760	002124		BIC #177760,SIGN	;YES - INSERT SIGN. NOTE: A ZERO LENGTH
11729	044474	000424				BR INSIGN	; PACKED STRING OCCUPIES 1 BYTE.
11730	044476	013737	002112	002116	3\$:	MOV STGLN,SAVSL	;STRING TO BE FORMED HAVE EVEN LENGTH?
11731	044504	042737	177776	002116		BIC #177776,SAVSL	
11732	044512	001003				BNE 1\$	
11733	044514	113721	002110			MOVB STGDS2,(R1)+	;YES - INSERT 0000-DIGIT BYTE
11734	044520	000404				BR 2\$	
11735	044522	113721	002106		1\$:	MOVB STGDS1,(R1)+	;INSERT NEXT PACKED DATA BYTE
11736	044526	005337	002112			DEC STGLN	
11737	044532	005337	002112		2\$:	DEC STGLN	
11738	044536	005737	002112			TST STGLN	;ENTIRE STRING BEEN INSERTED?
11739	044542	003367				BGT 1\$	;NO - CONTINUE INSERTING
11740	044544	005301				DEC R1	;YES - BACKUP
11741	044546	022737	000007	002310	INSIGN:	CMP #7,STGTYP	;IS STRING TYPE UNSIGNED?
11742	044554	001003				BNE 1\$	;BRANCH IF NO
11743	044556	052737	000017	002124		BIS #17,SIGN	;UNSIGNED PACKED STRING SIGN MUST = (1111)
11744	044564	113711	002124		1\$:	MOVB SIGN,(R1)	;INSERT SIGN
11745	044570	000137	045740			JMP E1STG	
11746							
11747	044574				DSTYP4:		;DECIMAL STRING - USER DEFINED DIGIT STRING
11748	044574	032737	000002	002460		BIT #2,ZPM	;IS FORMED STRING TO BE PACKED OR ZONED
11749	044602	001002				BNE TYP4Z	;BRANCH IF ZONED
11750	044604	000137	045314			JMP TYP4P	
11751							
11752	044610				TYP4Z:		;ZONED
11753	044610	013701	002112			MOV STGLN,R1	;GET STRING DESC. TYPE FIELD
11754	044614	000301				SWAB R1	
11755	044616	006201				ASR R1	
11756	044620	006201				ASR R1	
11757	044622	006201				ASR R1	
11758	044624	006201				ASR R1	
11759	044626	042701	177770			BIC #177770,R1	
11760	044632	010137	002310			MOV R1,STGTYP	;SAVE TYPE
11761	044636	042737	177740	002112		BIC #177740,STGLN	;STRIP TYPE OFF STRING LENGTH WORD

11762	044644	042737	010000	002106	BIC #10000,STGDS1	;STRIP OFF ALL BUT LENGTH
11763	044652	013701	002114		MOV STGAD,R1	;INSERTION IS DONE FROM LS TO MS DIGIT
11764	044656	063701	002112		ADD STGLN,R1	;FORM IN R1 ADDRESS WHERE FIRST BYTE IS TO BE INSERTED
11765	044662	010137	002436		MOV R1,ONEBEY	;SAVE PTR TO 1 BYTE BEYOND LS END OF STRING
11766	044666	013737	002110	002120	MOV STGDS2,SAVSA	;SAVE GIVEN STRING DESCRIPTOR WORDS
11767	044674	013737	002106	002116	MOV STGDS1,SAVSL	
11768	044702	063737	002106	002110	ADD STGDS1,STGDS2	;FORM IN STGDS2 ADDRESS WHERE FIRST BYTE IS TO BE
11769						; TAKEN FROM
11770	044710	005337	002110		DEC STGDS2	
11771	044714	005737	001770		TST RANDTA	;USE RANDOM SIGN?
11772	044720	001403			BEQ 4\$	
11773	044722	004737	045154		JSR PC,GETSGN	;YES
11774	044726	000416			BR 5\$	
11775	044730	117737	135154	002124	4\$: MOVB @STGDS2,SIGN	
11776	044736	006237	002124		ASR SIGN	
11777	044742	006237	002124		ASR SIGN	
11778	044746	006237	002124		ASR SIGN	
11779	044752	006237	002124		ASR SIGN	
11780	044756	042737	177760	002124	BIC #177760,SIGN	
11781	044764	005737	002112	5\$:	TST STGLN	;STRING TO BE FORMED HAVE 0 LENGTH?
11782	044770	001002			BNE 1\$	;BRANCH IF NO
11783	044772	000137	044320		JMP TFS	
11784	044776	005737	002112	1\$:	TST STGLN	;ENTIRE STRING BEEN INSERTED?
11785	045002	001432			BEQ WONSGN	;BRANCH IF YES
11786	045004	005737	001770		TST RANDTA	;USE RANDOM DATA?
11787	045010	001404			BEQ 3\$	;BRANCH IF NO
11788	045012	004737	045654		JSR PC,GRZDB	;GENERATE IN R0 A RANDOM ZONED DATA BYTE
11789	045016	110041			MOVB R0,-(R1)	;INSERT BYTE
11790	045020	000420			BR 2\$	
11791	045022	117741	135062	3\$:	MOVB @STGDS2,-(R1)	;INSERT NEXT BYTE
11792	045026	005337	002110		DEC STGDS2	;UPDATE POINTERS
11793	045032	005337	002106		DEC STGDS1	
11794	045036	001011			BNE 2\$	;GIVEN STRING EXHAUSTED? BRANCH IF NO
11795	045040	013737	002116	002106	MOV SAVSL,STGDS1	;RESET POINTERS BACK TO BEGINNING OF GIVEN STRING
11796	045046	013737	002120	002110	MOV SAVSA,STGDS2	
11797	045054	063737	002106	002110	ADD STGDS1,STGDS2	
11798	045062	005337	002112	2\$:	DEC STGLN	;DECREMENT COUNT OF # OF DIGITS TO BE INSERTED
11799	045066	000743			BR 1\$	;RETURN
11800	045070	013702	002310	WONSGN:	MOV STGTYP,R2	;WORK ON INSERTING SIGN BYTE
11801	045074	006302			ASL R2	
11802	045076	062702	002416		ADD #PTYPTA,R2	
11803	045102	000172	000000		JMP @(R2)	
11804						
11805	045106				PTYPTO:	;FIX UP POINTERS TO ENABLE USE OF
11806						; TYP3Z ROUTINES.
11807	045106	013701	002436		MOV ONEBEY,R1	;GET LEAST SIGN DIGIT
11808	045112	114101			MOVB -(R1),R1	
11809	045114	042701	177760		BIC #177760,R1	
11810	045120	010137	002306		MOV R1,STGDIG	;SAVE IN STGDIG
11811	045124	004737	045154		JSR PC,GETSGN	
11812	045130	013701	002436		MOV ONEBEY,R1	;SETUP R1 TO ONE BYTE BEYOND STRING
11813	045134	023727	002310	000002	CMP STGTYP,#2	
11814	045142	001402			BEQ 1\$	
11815	045144	000137	044234		JMP TYPLO	

11816	045150	000137	044126	1\$:	JMP TYPTO	
11817						
11818	045154	005737	001770	GETSGN:	TST RANDTA	;USE RANDOM SIGN?
11819	045160	001415			BEQ 1\$	;BRANCH IF NO
11820	045162	004737	063532		JSR PC,RN	;GET A RANDOM #
11821	045166	032700	000001		BIT #1,R0	;USE LEAST SIGN BIT TO PICK A SIGN
11822	045172	001404			BEQ 2\$	
11823	045174	012737	000007	002124	MOV #7,SIGN	;LS BIT = 1; MAKE SIGN NEG.
11824	045202	000417			BR 3\$	
11825	045204	012737	000003	002124	2\$:	MOV #3,SIGN
11826	045212	000413			BR 3\$	;LS BIT = 0;MAKE SIGN POS.
11827	045214	013701	002436	1\$:	MOV ONEBEY,R1	;SUBROUTINE TO GET SIGN FROM INSERTED STRING
11828	045220	114101			MOVB -(R1),R1	
11829	045222	042701	177417		BIC #177417,R1	
11830	045226	006201			ASR R1	
11831	045230	006201			ASR R1	;RIGHT ADJUST
11832	045232	006201			ASR R1	
11833	045234	006201			ASR R1	
11834	045236	010137	002124		MOV R1,SIGN	;SAVE SIGN IN 'SIGN'
11835	045242	000207		3\$:	RTS PC	
11836						
11837	045244			PTYPSZ:		;INSERT SIGN TYPE SIGN
11838	045244	013701	002436		MOV ONEBEY,R1	
11839	045250	000137	044072		JMP TYPSZ	
11840						
11841	045254			PTYPLO:		;INSERT LEADING OVERPUNCH SIGN
11842	045254	117701	134634		MOVB @STGAD,R1	;GET MOST SIGN. DIGIT
11843	045260	000137	045114		JMP PTO	
11844						
11845	045264			PTYPTS:		;INSERT TRAILING SEPARATE SIGN
11846	045264	004737	045154		JSR PC,GETSGN	;GET SIGN TO INSERT
11847	045270	013701	002436		MOV ONEBEY,R1	
11848	045274	000137	044244		JMP TYPTS	
11849						
11850	045300			PTYPLS:		;INSERT LEADING SEPARATE SIGN
11851	045300	004737	045154		JSR PC,GETSGN	
11852	045304	013701	002436		MOV ONEBEY,R1	
11853	045310	000137	044310		JMP TYPLS	
11854						
11855	045314	013701	002112	TYP4P:	MOV STGLN,R1	;GET STRING DESC. TYPE FIELD
11856	045320	000301			SWAB R1	
11857	045322	006201			ASR R1	
11858	045324	006201			ASR R1	
11859	045326	006201			ASR R1	
11860	045330	006201			ASR R1	
11861	045332	042701	177770		BIC #177770,R1	
11862	045336	010137	002310		MOV R1,STGTYP	;SAVE TYPE
11863	045342	042737	177740	002112	BIC #177740,STGLN	;STRIP TYPE OFF STRING LENGTH WORD
11864	045350	013737	002112	002122	MOV STGLN,SAVSGL	;PACKED - SAVE STRING LENGTH
11865	045356	006237	002112		ASR STGLN	;INSERTION IS DONE FROM LS TO MS DIGIT
11866	045362	013701	002112		MOV STGLN,R1	;FORM IN R1 ADDRESS WHERE 1ST BYTE IS TO BE INSERTED
11867	045366	063701	002114		ADD STGAD,R1	
11868	045372	006237	002106		ASR STGDS1	
11869	045376	013737	002106	002116	MOV STGDS1,SAVSL	

11870	045404	013737	002110	002120		MOV STGDS2,SAVSA	;SAVE GIVEN STRING DESCRIPTOR WORDS
11871	045412	063737	002106	002110		ADD STGDS1,STGDS2	;FORM IN STGDS2 ADDRESS WHERE 1ST BYTE
11872	045420	005201				INC R1	;IS TO BE TAKEN FROM
11873	045422	005737	002112		1\$:	TST STGLN	;ENTIRE STRING BEEN INSERTED?
11874	045426	100432				BMI 3\$	;BRANCH IF YES
11875	045430	005737	001770			TST RANDTA	;USE RANDOM DATA?
11876	045434	001404				BEQ 5\$	;BRANCH IF NO
11877	045436	004737	045720			JSR PC,GRPDB	;GENERATE IN RO A RANDOM PACKED DATA BYTE
11878	045442	110041				MOV B R0,-(R1)	;INSERT BYTE
11879	045444	000420				BR 2\$	
11880	045446	117741	134436		5\$:	MOV B @STGDS2,-(R1)	;INSERT NEXT BYTE
11881	045452	005337	002110			DEC STGDS2	
11882	045456	005337	002106			DEC STGDS1	;UPDATE POINTERS
11883	045462	002011				BGE 2\$	;GIVEN STRING EXHAUSTED? BRANCH IF NO
11884	045464	013737	002116	002106		MOV SAVSL,STGDS1	;RESET POINTERS BACK TO BEGINNING OF GIVEN STRING
11885	045472	013737	002120	002110		MOV SAVSA,STGDS2	
11886	045500	063737	002106	002110		ADD STGDS1,STGDS2	
11887	045506	005337	002112		2\$:	DEC STGLN	;DECREMENT COUNT OF # OF BYTES TO BE INSERTED
11888	045512	000743				BR 1\$	
11889	045514	032737	000001	002122	3\$:	BIT #1,SAVSGL	;IS STRING LENGTH ODD?
11890	045522	001002				BNE 4\$	;BRANCH IF YES - DONE
11891	045524	142721	000360			BICB #360,(R1)+	;ZERO NIBBLE IN MOST SIGN BYTE
11892	045530	013701	002122		4\$:	MOV SAVSGL,R1	;CALCULATE SIGN ADDRESS
11893	045534	006201				ASR R1	
11894	045536	063701	002114			ADD STGAD,R1	
11895	045542	022737	000007	002310		CMP #7,STGTYP	;IS STRING TYPE UNSIGNED?
11896	045550	001431				BEQ 6\$	;BRANCH IF YES
11897	045552	005737	001770			TST RANDTA	;USE RANDOM SIGN?
11898	045556	001470				BEQ EISTG	;BRANCH IF NO
11899	045560	142711	000017			BICB #17,(R1)	;CLEAR OUT FOR SIGN
11900	045564	004737	063532			JSR PC,RN	
11901	045570	032700	000001			BIT #1,R0	
11902	045574	001403				BEQ 7\$	
11903	045576	152711	000013			BISB #13,(R1)	;MAKE SIGN NEGATIVE
11904	045602	000456				BR EISTG	
11905	045604	032700	000002		7\$:	BIT #2,R0	
11906	045610	001403				BEQ 10\$	
11907	045612	152711	000016			BISB #16,(R1)	;MAKE SIGN +
11908	045616	000450				BR EISTG	
11909	045620	032700	000004		10\$:	BIT #4,R0	
11910	045624	001403				BEQ 6\$	
11911	045626	152711	000012			BISB #12,(R1)	;MAKE SIGN +
11912	045632	000442				BR EISTG	
11913	045634	152711	000017		6\$:	BISB #17,(R1)	;OVERWRITE SIGN TO (1111)
11914	045640	000437				BR EISTG	
11915							
11916	045642				DSTYP5:		;DECIMAL STRING - RANDOM DATA & SIGN
11917	045642	012737	177777	001770		MOV #177777,RANDTA	;SET RANDOM DATA FLAG
11918	045650	000137	044574			JMP DSTYP4	;USE DSTYP4 ROUTINES.
11919							
11920	045654				GRZDB:		;SUBROUTINE TO GENERATE A RANDOM ZONED
11921							;DATA BYTE IN RO.
11922	045654	004737	063532			JSR PC,RN	;GET A RANDOM #.
11923	045660	042700	177740			BIC #BS128,R0	;STRIP OFF ALL BUT LS NIBBLE

11924	045664	020027	000011		CMP R0,#11		;VALID NIBBLE = 0 TO 11.
11925	045670	101402			BLOS 1\$		;BRANCH IF NIBBLE IS VALID
11926	045672	042700	000010		BIC #10,R0		;CONVERT INVALID NIBBLE TO A VALID ONE
11927	045676	010037	001774	1\$:	MOV R0,RNIB		;SAVE NIBBLE
11928	045702	004737	063532		JSR PC,RN		;GET ANOTHER RANDOM #
11929	045706	043700	001766		BIC ZMSK,R0		;MASK OFF ALL BUT ZONE FIELD
11930	045712	053700	001774		BIS RNIB,R0		;OR - IN THE DECIMAL NIBBLE
11931	045716	000207			RTS PC		;RETURN WITH ZONED DATA BYTE IN R0
11932							
11933	045720			GPPDB:			;SUBROUTINE TO GENERATE A RANDOM PACKED
11934							;DATA BYTE IN R0.
11935	045720	004737	045654		JSR PC,GZDB		;GET A RANDOM ZONED DATA BYTE
11936	045724	020027	000231		CMP R0,#231		;HIGH NIBBLE MUST BE <=9
11937	045730	101402			BLOS 1\$		;BRANCH IF VALID
11938	045732	042700	000200		BIC #200,R0		;CONVERT TO VALID
11939	045736	000207		1\$:	RTS PC		
11940	045740	000205		E1STG:	RTS R5		
11941							

11943	045742				FC10:		;FLOW COMMAND = 10 - COPY TEST BUFFER INTO
11944							; EMULATION BUFFER
11945	045742	032737	000100	050012		BIT #100,TINST	;THE IN-LINE TEST CASE ALWAYS FOLLOWS
11946							; THE SAME REGISTER TEST CASE - THEREFORE
11947							; THEIR IS NO NEED TO REEMULATE.
11948	045750	001010				BNE 2\$	
11949	045752	013701	001640			MOV TBADR,R1	;R1 POINTS TO START OF TEST BUFFER
11950	045756	013702	001646			MOV EBADR,R2	;R2 POINTS TO START OF EMULATION BUFFER
11951	045762	012122			1\$:	MOV (R1)+,(R2)+	;COPY NEXT WORD
11952	045764	023701	001754			CMP TBEND,R1	;COPY COMPLETE ?
11953	045770	003374				BGT 1\$	;NO
11954	045772	000137	042250		2\$:	JMP FCRTN	;YES
11955							
11956	045776				FC11:		;FLOW COMMAND = 11 - SETUP EMULATION
11957							; OPERANDS IN ERO - ER5 AND EMULATE
11958							; THE CIS INSTRUCTION UNDER TEST.
11959	045776	032737	000100	050012		BIT #100,TINST	;THE IN-LINE TEST CASE ALWAYS FOLLOWS THE SAME
11960							; REGISTER TEST CASE - THEREFORE THEIR IS NO
11961							; NEED TO REEMULATE.
11962	046004	001060				BNE FINEM	
11963	046006	013737	003630	003670		MOV TR0,ERO	;COPY ERN DIRECTLY FROM TRN
11964	046014	013737	003632	003672		MOV TR1,ER1	
11965	046022	013737	003634	003674		MOV TR2,ER2	
11966	046030	013737	003636	003676		MOV TR3,ER3	
11967	046036	013737	003640	003700		MOV TR4,ER4	
11968	046044	013737	003642	003702		MOV TR5,ER5	
11969	046052	013737	003644	003704		MOV TR6,ER6	
11970							
11971							;ADJUST ERN SPECIFIED IN FLOW TABLE ENTRY
11972							; TO ACCOUNT FOR EMULATION VERSUS TEST
11973							; BUFFER STARTING ADDRESS
11974	046060	004737	053666			JSR PC,RF1	;FORM POINTER TO FIRST TEST OPERAND TO
11975							; BE ADJUSTED.
11976	046064	020127	003630			CMP R1,#TRN	;ANY OPERANDS TO BE ADJUSTED?
11977	046070	001420				BEQ EC11	;BRANCH IF NO.
11978	046072	004737	046152			JSR PC,ADJEOP	;ADJUST OPERAND
11979	046076	004737	053720			JSR PC,RF2	;FORM POINTER TO SECOND TEST OPERAND TO
11980							; BE ADJUSTED.
11981	046102	020127	003630			CMP R1,#TRN	;IF R1 STILL POINTS TO #TRN THEN THERE
11982							; WAS ONLY ONE OPERAND TO BE ADJUSTED.
11983	046106	001411				BEQ EC11	;ADJUSTING COMPLETE
11984	046110	004737	046152			JSR PC,ADJEOP	;ADJUST NEXT EMULATION OPERAND
11985	046114	004737	053754			JSR PC,RF3	;FORM POINTER TO THIRD TEST OPERAND TO
11986							; BE ADJUSTED.
11987	046120	020127	003630			CMP R1,#TRN	;WAS THERE A THIRD OPERAND TO BE ADJUSTED?
11988	046124	001402				BEQ EC11	;NO - ADJUSTING COMPLETE
11989	046126	004737	046152			JSR PC,ADJEOP	;YES - ADJUST IT.
11990							
11991	046132				EC11:		;EMULATION OPERANDS ALL SET.
11992							;EMULATE CIS INST.
11993	046132	004737	017144			JSR PC,EMULATE	;CALL EMULATOR
11994	046136	000000			EINSTR:	0	;INSTRUCTION
11995	046140	003670				ERN	;POINTER TO REGISTER OPERANDS
11996	046142	003706				ERNR	;POINTER TO REGISTER RESULTS

11997	046144	003724			ECCR				: POINTER TO COND. CODE RESULTS
11998	046146	000137	042250		FINEM: JMP FCRTN				
11999									
12000	046152				ADJEOP:				: SUBROUTINE TO UPDATE EMULATION OPERANDS
12001									: TO REFLECT EMULATION BUFFER STARTING ADDRESS.
12002	046152	162701	003630		SUB #TRN,R1				
12003	046156	062701	003670		ADD #ERN,R1				: POINT R1 TO EMULATION OPERAND RATHER THAN
12004									: TEST OPERAND.
12005	046162	163711	001640		SUB TBADR,(R1)				: SUBTRACT OUT TEST BUFFER STARTING ADDRESS
12006	046166	063711	001646		ADD EBADR,(R1)				: ADD IN EMULATION BUFFER STARTING ADDRESS
12007	046172	000207			RTS PC				
12008									
12009	046174				FC12:				: FLOW COMMAND = 12 - SETUP REGISTERS (OR IN-LINE POINTERS
12010									: AND COND CODES AND EXECUTE CIS INSTRUCTION.
12011									: SAVE RESULTS.
12012	046174	013737	003630	002234	MOV TRO,TRO				: SAVE TRNS IN TEMPORARY STORAGE
12013	046202	013737	003632	002236	MOV TR1,TTR1				
12014	046210	013737	003634	002240	MOV TR2,TTR2				
12015	046216	013737	003636	002242	MOV TR3,TTR3				
12016	046224	013737	003640	002244	MOV TR4,TTR4				
12017	046232	013737	003642	002246	MOV TR5,TTR5				
12018	046240	032737	000100	050012	BIT #100,TINST				: INST TYPE (REG OR IN-LINE)?
12019	046246	001014			BNE INLINE				: BRANCH IF INLINE TYPE
12020	046250	023727	050012	076027	CMP TINST,#76027				
12021	046256	001406			BEQ NONOP				
12022	046260	023727	050012	076067	CMP TINST,#76067				
12023	046266	001402			BEQ NONOP				
12024	046270	000137	047124		JMP REGOP				
12025	046274	000137	047170		NONOP: JMP LTSTOP				
12026	046300	004737	053700		INLINE: JSR PC,RF1X				: GET # OF IN-LINE DESCRIPTOR POINTERS
12027	046304	006201			ASR R1				
12028	046306	012702	000005		MOV #5,R2				: CALCULATE BRANCH TO INSERT IMMEDIATELY
12029									: FOLLOWING IN LINE DESCRIPTORS.
12030	046312	160102			SUB R1,R2				
12031	046314	062702	000400		ADD #400,R2				
12032	046320	010237	001756		MOV R2,SBR				: SAVE BRANCH INST
12033	046324	005002			CLR R2				
12034	046326	013703	001640		MOV TBADR,R3				
12035	046332	005701			2\$: TST R1				
12036	046334	001410			BEQ 1\$				: BRANCH IF ALL IN-LINE PTRS HAVE BEEN INSERTED.
12037	046336	010362	050014		MOV R3,TINST+2(R2)				: INSERT DESCRIPTOR POINTERS IN-LINE
12038	046342	005301			DEC R1				: UPDATE FOR NEXT IN-LINE PTR
12039	046344	062702	000002		ADD #2,R2				
12040	046350	062703	000004		ADD #4,R3				
12041	046354	000766			BR 2\$				: RETURN TO WORK ON NEXT IN-LINE PTR.
12042	046356	012737	000001	002232	1\$: MOV #1,NXFLD				: INITIALIZE FIELD PTR TO SECOND FIELD
12043	046364	004737	054132		11\$: JSR PC,RFN				: GET NEXT FIELD CONTENTS.
12044	046370	020127	003630		CMP R1,#TRN				: ALL OPERANDS INSERTED IN-LINE?
12045	046374	001407			BEQ 3\$				: BRANCH IF YES
12046	046376	005337	001756		DEC SBR				
12047	046402	011162	050014		MOV (R1),TINST+2(R2)				: MOVE REGISTER OPERAND INTO IN-LINE LOC
12048	046406	062702	000002		ADD #2,R2				: UPDATE TO NEXT IN-LINE LOC
12049	046412	000764			BR 11\$				: RETURN TO WORK ON NEXT IN-LINE OPERAND
12050	046414	013762	001756	050014	3\$: MOV SBR,TINST+2(R2)				: INSERT BRANCH



12051	046422	012737	050014	002214		MOV #TINST+2,ICOMPC	;SAVE BRANCH LOCATION FOR PC
12052	046430	060237	002214			ADD R2,ICOMPC	; CHECK ON IN-LINE INSTRUCTION COMPLETION
12053	046434	062702	000002		12\$:	ADD #2,R2	;INSERT HALTS FOLLOWING BRANCH
12054	046440	022702	000014			CMP #14,R2	
12055	046444	001404				BEQ 4\$	
12056	046446	013762	001672	050014		MOV KHALT,TINST+2(R2)	
12057	046454	000767				BR 12\$	
12058							
12059	046456	062737	000002	001742	4\$:	ADD #2,FLOPTR	;LOOK AT NEXT FLOW COMMAND (=00)
12060	046464	013702	001640			MOV TBADR,R2	;INITIALIZE POINTERS TO TEST & EMULATION BUFFERS
12061	046470	013703	001646			MOV EBADR,R3	
12062	046474	005037	002232			CLR NXFLD	;INITIALIZE FIELD PTR TO 1ST FIELD
12063	046500	004737	054132		41\$:	JSR PC,RFN	;GET NEXT FIELD CONTENTS
12064	046504	020127	003630			CMP R1,#TRN	;ALL BUFFER DESCRIPTORS FILLED?
12065	046510	001441				BEQ 5\$	;BRANCH IF YES
12066	046512	020127	003636			CMP R1,#TR3	;FOR CVTLN & CVTLP THE MEMORY ORDER
12067	046516	001021				BNE 42\$	; OF THE LONG INTEGER LOW & HIGH
12068	046520	023727	050012	076157		CMP TINST,#76157	; IS REVERSED FROM THE REGISTER ORDER
12069	046526	001404				BEQ 43\$	
12070	046530	023727	050012	076177		CMP TINST,#76177	
12071	046536	001011				BNE 42\$	
12072	046540	011112			43\$:	MOV (R1),(R2)	
12073	046542	011113				MOV (R1),(R3)	
12074	046544	162701	000002			SUB #2,R1	
12075	046550	011162	000002			MOV (R1),2(R2)	
12076	046554	011163	000002			MOV (R1),2(R3)	
12077	046560	000410				BR 44\$	
12078	046562	011162	000002		42\$:	MOV (R1),2(R2)	;COPY BUFFER DESCRIPTORS FROM REGISTER
12079	046566	011163	000002			MOV (R1),2(R3)	; DESCRIPTORS.
12080	046572	162701	000002			SUB #2,R1	
12081	046576	011112				MOV (R1),(R2)	
12082	046600	011113				MOV (R1),(R3)	
12083	046602	062702	000004		44\$:	ADD #4,R2	;UPDATE BUFFER DESCRIPTOR POINTERS
12084	046606	062703	000004			ADD #4,R3	
12085	046612	000732				BR 41\$	;RETURN TO FILL NEXT DESCRIPTOR WORDS.
12086	046614	062737	000002	001742	5\$:	ADD #2,FLOPTR	;LOOK AT NEXT FLOW COMMAND (=00)
12087	046622	005737	002272			TST RPTFLG	;IS TEST BEING REPEATED??
12088	046626	001025				BNE 6\$	;BRANCH IF YES (DONT UPDATE EMUL BUFFER
12089							; DESCRIPTORS - THEY ARE ALREADY UP-TO-DATE).
12090	046630	013703	001646			MOV EBADR,R3	;INITIALIZE POINTER TO EMULATION BUFFER.
12091	046634	005037	002232			CLR NXFLD	;INITIALIZE FIELD PTR TO 1ST FIELD
12092	046640	004737	054032		51\$:	JSR PC,RFN	;GET NEXT FIELD CONTENTS
12093	046644	005701				TST R1	;ALL RESULT EMULATOR BUFFER DESCRIPTORS UPDATED?
12094	046646	001415				BEQ 6\$	;BRANCH IF YES
12095	046650	020127	000016			CMP R1,#16	;DESCRIPTOR TO BE UPDATED?
12096	046654	001407				BEQ 52\$	;BRANCH IF NO
12097	046656	062701	003706			ADD #ERNR,R1	;FORM POINTER INTO EMUL. RESULT STACK
12098	046662	011113				MOV (R1),(R3)	;COPY EMUL. RESULT INTO BUFFER
12099	046664	162701	000002			SUB #2,R1	
12100	046670	011163	000002			MOV (R1),2(R3)	
12101	046674	062703	000004		52\$:	ADD #4,R3	;UPDATE EMUL. BUFFER POINTER
12102	046700	000757				BR 51\$	;RETURN TO WORK ON NEXT EMUL. RESULT.
12103							
12104							

12105	046702	013737	001720	003630	6\$:	MOV PAT0,TR0	;INITIALIZE TRX'S TO #PATTERN
12106	046710	013737	001722	003632		MOV PAT1,TR1	
12107	046716	013737	001724	003634		MOV PAT2,TR2	
12108	046724	013737	001726	003636		MOV PAT3,TR3	
12109	046732	013737	001730	003640		MOV PAT4,TR4	
12110	046740	013737	001732	003642		MOV PAT5,TR5	
12111							
12112	046746	062737	000002	001742	7\$:	ADD #2,FLOPTR	;OVERWRITE WITH #PATTERN ALL ERNRS EXCEPT
12113							; THOSE THAT CONTAIN A REGISTER RESULT.
12114							;LOOK AT NEXT FLOW COMMAND (=00)
12115	046754	013737	003706	002252		MOV ER0R,TER0R	;COPY ERNRS INTO TEMP SPACE AT TERNR
12116	046762	013737	003710	002254		MOV ER1R,TER1R	
12117	046770	013737	003712	002256		MOV ER2R,TER2R	
12118	046776	013737	003714	002260		MOV ER3R,TER3R	
12119	047004	013737	003716	002262		MOV ER4R,TER4R	
12120	047012	013737	003720	002264		MOV ER5R,TER5R	
12121							
12122	047020	013737	001720	003706	8\$:	MOV PAT0,ER0R	;OVERWRITE ERNR STACK WITH # PATTERN
12123	047026	013737	001722	003710		MOV PAT1,ER1R	
12124	047034	013737	001724	003712		MOV PAT2,ER2R	
12125	047042	013737	001726	003714		MOV PAT3,ER3R	
12126	047050	013737	001730	003716		MOV PAT4,ER4R	
12127	047056	013737	001732	003720		MOV PAT5,ER5R	
12128							
12129	047064	005037	002232		9\$:	CLR NXFLD	;INITIALIZE FIELD PTR TO 1ST FIELD
12130	047070	004737	054032		91\$:	JSR PC,RFNX	;GET NEXT FIELD CONTENTS
12131	047074	020127	000016			CMP R1,#16	;END OF REGISTER RESULT LIST?
12132	047100	001407				BEQ 10\$	;BRANCH IF YES
12133	047102	010102				MOV R1,R2	
12134	047104	062701	003706			ADD #ERNR,R1	
12135	047110	062702	002252			ADD #TER0R,R2	
12136	047114	011211				MOV (R2),(R1)	;COPY REGISTER RESULT BACK INTO ERNR STACK
12137							; FOR TEMPORARY STORAGE.
12138	047116	000764				BR 91\$	;RETURN TO WORK ON NEXT REGISTER RESULT.
12139	047120	000137	047170		10\$:	JMP LTSTOP	
12140							
12141	047124	013737	001674	050014	REGOP:	MOV KBR5,TINST+2	
12142	047132	013737	001672	050016		MOV KHALT,TINST+4	
12143	047140	013737	001672	050020		MOV KHALT,TINST+6	
12144	047146	013737	001672	050022		MOV KHALT,TINST+10	
12145	047154	013737	001672	050024		MOV KHALT,TINST+12	
12146	047162	013737	001672	050026		MOV KHALT,TINST+14	
12147							
12148	047170	010637	002230		LTSTOP:	MOV SP,TSP	;SAVE STACK POINTER
12149	047174	022737	076026	050012		CMP #076026,TINST	;IS INST A L2D6?
12150	047202	001003				BNE 2\$	;BRANCH IF NO
12151	047204	013706	003644		3\$:	MOV TR6,SP	;LOAD DESC 6 INST - USE TR6 CONTENTS AS SP
12152	047210	000411				BR 1\$	
12153	047212	022737	076066	050012	2\$:	CMP #076066,TINST	;IS INST A L3D6?
12154	047220	001771				BEQ 3\$	;BRANCH IF YES
12155	047222	012737	120602	003722		MOV #CSTACK,ER6R	
12156	047230	012706	120602			MOV #CSTACK,SP	
12157	047234	010637	003644		1\$:	MOV SP,TR6	;LOAD TR6 FOR INPUT REG DISPLAY
12158	047240	013737	003724	003646		MOV ECCR,TCC	;SETUP CONDITION CODES

12159	047246	013701	050012			MOV TINST,R1		:IS INST A L2D OR L3D?
12160	047252	042701	000007			BIC #7,R1		
12161	047256	022701	076020			CMP #076020,R1		:IF YES SETUP CONDITION CODES=EXPECTED CODES.
12162	047262	001405				BEQ 4\$		
12163	047264	022701	076060			CMP #076060,R1		
12164	047270	001402				BEQ 4\$		
12165	047272	005137	003646			COM TCC		:OTHERWISE SETUP CC=COMPLEMENT OF EXPECTED CC.
12166	047276	004737	054746		4\$:	JSR PC,SELREG		:SELECT REGISTER SET AND SETUP CONTENTS OF
12167								: REGISTER SET NOT SELECTED.
12168	047302	012737	000240	050030		MOV #NOP,TINRET		:RESTORE NOP TO CIS INST RETURN POINT
12169	047310	004737	056134			JSR PC,SELMD		:SELECT MODE & I/D ENABLES; LOAD MMR3 TO
12170								: REFLECT D-SPACE ENABLE/DIS SELECTION
12171	047314	005737	002160			TST MODE		:TEST MODE = KERNEL?
12172	047320	001403				BEQ 17\$		:BRANCH IF YES
12173	047322	012737	000000	050030		MOV #HALT,TINRET		:LOAD HALT AT INST UNDER TEST RETURN ADDRESS
12174								: (ALLOWS TRAPPING BACK TO KERNEL MODE
12175								: AFTER CIS INST EXECUTION).
12176	047330	005737	002156		17\$:	TST MMFLG		:IS MEMORY MGMT AVAILABLE?
12177	047334	001411				BEQ 8\$		:BRANCH IF NO
12178	047336	004737	056360			JSR PC,SETPDR		:SETUP MEM MGMT PDR'S
12179								:NOTE: PAR'S ARE SETUP AT BEGINNING OF PROG
12180								: KERNEL,USER & SUPV SPACES ARE ALWAYS MAPPED
12181								: TO SAME PHYSICAL MEMORY)
12182	047342	012737	000240	047774		MOV #NOP,TOMM		:ALLOW MEMORY MANAGEMENT TURN ON
12183	047350	012737	000240	050040		MOV #NOP,TOFMM		:ALLOW MEMORY MANAGEMENT TURN OFF
12184	047356	000406				BR 9\$		
12185	047360	012737	000403	047774	8\$:	MOV #403,TOMM		:INHIBIT MEM MGMT TURN ON
12186	047366	012737	000402	050040		MOV #402,TOFMM		
12187	047374	013700	003630		9\$:	MOV TR0,R0		:LOAD TEST OPERANDS INTO REGISTERS
12188	047400	010037	002566			MOV R0,STATR0		:SAVE STATE OF CISP
12189	047404	013701	003632			MOV TR1,R1		
12190	047410	010137	002570			MOV R1,STATR1		
12191	047414	013702	003634			MOV TR2,R2		
12192	047420	010237	002572			MOV R2,STATR2		
12193	047424	013703	003636			MOV TR3,R3		
12194	047430	010337	002574			MOV R3,STATR3		
12195	047434	013704	003640			MOV TR4,R4		
12196	047440	010437	002576			MOV R4,STATR4		
12197	047444	013705	003642			MOV TR5,R5		
12198	047450	010537	002600			MOV R5,STATR5		
12199	047454	010637	002602			MOV SP,STATR6		
12200	047460	005737	002160			TST MODE		:IS MODE = KERNEL?
12201	047464	001003				BNE 5\$		:BRANCH IF NO
12202	047466	162737	000006	002602		SUB #6,STATR6		:ADJUST SAVED STACK POINTER TO ACCOUNT FOR
12203								: INTERRUPT STACK PUSH (KERNEL = CIS INST STACK)
12204	047474	042737	177760	003646	5\$:	BIC #177760,TCC		:FORM PSW TO BE USED UPON ENTRY TO CIS INST
12205	047502	017737	132156	001752		MOV @TPSW,TSTPSW		
12206	047510	042737	170017	001752		BIC #170017,TSTPSW		
12207	047516	053737	003646	001752		BIS TCC,TSTPSW		
12208	047524	022737	000001	002160		CMP #1,MODE		:TEST MODE = SUPERVISOR?
12209	047532	001004				BNE 6\$		:BRANCH IF NO
12210	047534	052737	040000	001752		BIS #040000,TSTPSW		:SET CURRENT MODE - SUPV IN TSTPSW
12211	047542	000407				BR 7\$		
12212	047544	022737	000003	002160	6\$:	CMP #3,MODE		:TEST MODE = USER?

12213	047552	001015				BNE 10\$		;BRANCH IF NO
12214	047554	052737	140000	001752		BIS #140000,TSTPSW		;SET CURR MODE = USER IN TSTPSW
12215	047562	013777	001752	132074	7\$:	MOV TSTPSW,@TSPW		;MODE = USER + SUPV; SWITCH TO TEST MODE
12216	047570	013706	002602			MOV STATR6,SP		;SETUP TEST MODE R6
12217	047574	042777	140000	132062		BIC #140000,@TSPW		;SWITCH BACK TO KERNEL
12218	047602	013706	002230			MOV TSP,SP		;SETUP KERNEL MODE R6
12219	047606	013737	001752	002564	10\$:	MOV TSTPSW,STATPS		
12220	047614	005037	002546			CLR INTCT		;CLEAR INTERRUPT COUNT
12221	047620	042737	040000	002132		BIC #40000,FATAL		;CLEAR INTERRUPT INDICATOR IN FATAL ERROR WORD
12222	047626	005037	002540			CLR PROGCT		;CLEAR PROGRESS COUNT
12223	047632	022777	000001	132076		CMP #1,@SWR		;CHECK FOR REQUEST TO STOP ON A TEST
12224	047640	001027				BNE TOLTC		;DON'T STOP
12225	047642	005737	002216			TST STOPTF		;HAS TEST BEEN ENTERED?
12226	047646	001015				BNE 11\$		;BRANCH IF YES
12227	047650	012737	177777	002216		MOV #177777,STOPTF		;SET FLAG TO INDICATE THAT TEST HAS BEEN ENTERED
12228	047656	010046				MOV RO,-(SP)		;SAVE RO - PRINT CALL DESTROYS IT
12229	047660					PRINTB #ASKST		;ASK USER FOR TEST # TO STOP ON
(6)	047660	012746	013034			MOV #ASKST,-(SP)		
(3)	047664	010600				MOV SP,RO		
(4)	047666	004737	065410			JSR PC,FPRINT		
12230	047672	012600				MOV (SP)+,RO		;RESTORE RO
12231	047674	104403				RDDEC		;READ DECIMAL #
12232	047676	012637	002220			MOV (SP)+,STOPT		;SAVE STOP TEST #
12233	047702	023737	002220	001420	11\$:	CMP STOPT,TOTTC		;IS CURRENT TEST = STOP TEST
12234	047710	001003				BNE TOLTC		;BRANCH IF NO
12235	047712	012737	177777	002042		MOV #177777,NOERDS		;SET FLAGS TO CAUSE CURRENT TEST TO BE ;DISPLAYED AND USER TO BE QUERIED FOR FURTHER ACTION.
12236								;OVERWRITTEN WITH A NOP IF LTC IS USED FOR ; INTERRUPT TESTING
12237	047720	000414				TOLTC: BR TOPC2		;RESET LTC COUNTER
12238								;SYNC UP TO LTC
12239	047722	013737	003030	003032		MOV LCNT,VLCNT		;ENABLE LTC INTR
12240	047730	004737	062250			JSR PC,LTC SUP		;KILL MOST OF TIME BEFORE EXPECTED
12241	047734	052777	000100	133064		BIS #100,@LKS		; INTERRUPT
12242	047742	005337	003032		1\$:	DEC VLCNT		
12243								
12244	047746	001375				BNE 1\$		
12245	047750	000410				BR PREINS		
12246	047752	000403				TOPC2: BR TOPC1		;OVERWRITTEN WITH A NOP IF LATENCY IS BEING TESTED
12247	047754	052777	000001	132546		BIS #001,@PC2CSR		;TURN ON P-CLK2
12248	047762	000403				TOPC1: BR PREINS		;OVERWRITTEN WITH A NOP IF INTERRUPTABILITY IS B
12249	047764	052777	000101	132526		BIS #101,@PC1CSR		;TURN ON P-CLK1
12250	047772	000240				PREINS: NOP		;REPLACED WITH A HALT IF OPERATOR REQUESTED
12251								; 'HALT AT CIS INST'
12252	047774	000403				TOMM: BR GO		;OVERWRITTEN WITH A NOP IF MEM MGMT TEST STATE = ON
12253	047776	052737	000001	177572		BIS #1,@MMRO		;TURN ON MEMORY MGMT
12254	050004	013777	001752	131652	GO:	MOV TSTPSW,@TSPW		;SET PSW TO DESIRED STATE (PRIOR TO THIS MODE=KERNEL)
12255	050012	000000			TINST:	.WORD 0		;EXECUTE CIS INST UNDER TEST.
12256	050014	000405				BR TINRET		
12257	050016	000000				HALT		
12258	050020	000000				HALT		;IF PROGRAM STOPS AT ANY ONE OF THESE HALTS,
12259	050022	000000				HALT		; THEN THE CIS INSTRUCTION EXECUTION
12260	050024	000000				HALT		; RETURNED WITH INCORRECT PC.
12261	050026	000000				HALT		
12262	050030	000240				TINRET: NOP		;OVERWRITTEN WITH A HALT IF MODE = USER OR SUPV
12263								;NOTE: HALT IS USED TO TRAP BACK TO KERNEL MODE



12312	050274	022701	003646			CMP #TRN+16,R1	
12313	050300	001433				BEQ CCKK	;DONE?
12314	050302	004737	051264			JSR PC,ADJR	;NO - ADJUST AGAIN
12315	050306	004737	053754			JSR PC,RF3	;FORM POINTER TO 3RD TEST OPERAND TO BE ADJUSTED
12316	050312	022701	003646			CMP #TRN+16,R1	
12317	050316	001424				BEQ CCKK	;DONE?
12318	050320	004737	051264			JSR PC,ADJR	;NO - ADJUST
12319	050324	004737	054004			JSR PC,RF4	;FORM POINTER TO 4TH TEST OPERAND TO BE ADJUSTED
12320	050330	022701	003646			CMP #TRN+16,R1	
12321	050334	001415				BEQ CCKK	;DONE?
12322	050336	004737	051264			JSR PC,ADJR	;NO - ADJUST
12323	050342	000412				BR CCKK	
12324	050344	062737	000002	001742	ADJI:	ADD #2,FLOPTR	;LOOK AT NEXT FLOW COMMAND (=00)
12325	050352	004737	053666			JSR PC,RF1	;FORM PTR TO INLINE TYPE INST REG OPERAND
12326							; TO BE ADJUSTED.
12327	050356	022701	003646			CMP #TRN+16,R1	;IF OPERAND = 7 THEN NO REG OPERANDS TO ADJUST.
12328	050362	001402				BEQ CCKK	;DONE?
12329	050364	004737	051264			JSR PC,ADJR	;NO ADJUST SPECIFIED OPERAND.
12330							;NOTE: INLINE TYPE INST NEVER REQUIRE
12331							; THAT MORE THAN 1 OPERAND BE ADJUSTED.
12332	050370	042737	177760	003724	CCKK:	BIC #177760,ECCR	;CLEAR OUT ALL BUT CONDITION CODES
12333	050376	005737	036210			TST EZDF	;CONDITION UNDER TEST = DIVP BY 0?
12334	050402	001406				BEQ 1\$	;BRANCH IF NO
12335	050404	042737	000014	003666		BIC #14,TCCR	;MASK OUT ALL BUT C & V COND. CODES
12336	050412	042737	000014	003724		BIC #14,ECCR	
12337	050420	004737	070006		1\$:	JSR PC,RECCC	;RECORD CONDITION CODE USAGE
12338	050424	023737	003724	003666		CMP ECCR,TCCR	;CHECK CONDITION CODE RESULTS
12339	050432	001403				BEQ REGCK	
12340	050434	012737	177777	002144		MOV #177777,ERRCC	;SET CONDITION CODE ERROR FLAG
12341	050442	005737	036210		REGCK:	TST EZDF	;CONDITION UNDER TEST = DIVP BY 0?
12342	050446	001104				BNE EZDBCK	;BRANCH IF YES
12343	050450	023737	003706	003650		CMP ER0R,TR0R	;CHECK REGISTER RESULTS
12344	050456	001030				BNE REGERR	
12345	050460	023737	003710	003652		CMP ER1R,TR1R	
12346	050466	001024				BNE REGERR	
12347	050470	023737	003712	003654		CMP ER2R,TR2R	
12348	050476	001020				BNE REGERR	
12349	050500	023737	003714	003656		CMP ER3R,TR3R	
12350	050506	001014				BNE REGERR	
12351	050510	023737	003716	003660		CMP ER4R,TR4R	
12352	050516	001010				BNE REGERR	
12353	050520	023737	003720	003662		CMP ER5R,TR5R	
12354	050526	001004				BNE REGERR	
12355	050530	023737	003722	003664		CMP ER6R,TR6R	
12356	050536	001403				BEQ BUFCK	
12357	050540	012737	177777	002146	REGERR:	MOV #177777,ERRREG	;REGISTER ERROR - SET REGISTER ERROR FLAG
12358	050546	032737	000004	002140	BUFCK:	BIT #4,SPHAND	;SPECIAL HANDLING?
12359	050554	001077				BNE TFORE	;IF YES SKIP BUFFER CHECK
12360	050556	013701	001640			MOV TBADR,R1	;CHECK BUFFER RESULTS
12361	050562	013702	001646			MOV EBADR,R2	
12362	050566	005737	002160			TST MODE	;IS MODE = KERNEL?
12363	050572	001024				BNE 1\$	;BRANCH IF NO
12364	050574	023727	047762	000240		CMP TOPC1,#NOP	;INTERRUPTABILITY BEING TESTED?
12365	050602	001404				BEQ 3\$	;BRANCH IF YES

12366	050604	023727	047720	000240		CMP TOLTC,#NOP	
12367	050612	001014				BNE 1\$	:BRANCH IF NO
12368	050614	023727	050012	076026	3\$:	CMP TINST,#76026	:IS INST UNDER TEST A L2D6 OR L3D6
12369	050622	001404				BEQ 2\$	
12370	050624	023727	050012	076066		CMP TINST,#76066	
12371	050632	001004				BNE 1\$	
12372	050634	062701	000006		2\$:	ADD #6,R1	:YES - SPECIAL CASE
12373	050640	062702	000006			ADD #6,R2	:L2D6 OR L3D6 WITH INTERRUPTABILITY TEST.MODE=KERNEL.
12374							:DUE TO THE INTERRUPT OCCURRING IMMEDIATELY
12375							:AFTER THE L2D6 OR L3D6 INST WHEN R6 POINTS
12376							:TO TBADR (TEST BUFFER), THE FIRST 2 TO 3 WORDS OF
12377							:TBADR WILL GET DESTROYED. DONT COMPARE
12378							:FIRST 3 WORDS OF BUFFERS.
12379	050644	023701	001754		1\$:	MP TBEND,R1	:REACH END OF BUFFER YET?
12380	050650	101441				BLOS TFORE	:YES
12381	050652	122122				CMPB (R1)+,(R2)+	:NO - COMPARE NEXT WORD
12382	050654	001773				BEQ 1\$	
12383	050656	000421				BR BUFERR	
12384							
12385	050660	013701	001640		EZDBCK:	MOV TBADR,R1	:CHECK BUFFER RESULTS - ZERO DIVP CASE
12386	050664	013702	001646			MOV EBADR,R2	:DON'T COMPARE ON DST STRING
12387	050670	023701	001754		2\$:	CMP TBEND,R1	:REACH END OF BUFFER YET?
12388	050674	101427				BLOS TFORE	:YES
12389	050676	020237	036212			CMP R2,EZDBEG	:AT BEGINNING OF DST STRING?
12390	050702	001005				BNE 1\$	:BRANCH IF NO
12391	050704	063702	036214			ADD EZDEND,R2	:ADJUST POINTERS TO END OF DST STRING
12392	050710	063701	036214			ADD EZDEND,R1	
12393	050714	000765				BR 2\$	
12394	050716	122122			1\$:	CMPB (R1)+,(R2)+	:COMPARE NEXT BYTE
12395	050720	001763				BEQ 2\$	
12396	050722	012737	177777	002150	BUFERR:	MOV #177777,ERRBUF	:BUFFER ERROR - SET BUFFER ERROR FLAG
12397	050730	005301				DEC R1	
12398	050732	005302				DEC R2	
12399	050734	010137	002174			MOV R1,AEADR	
12400	050740	010237	002200			MOV R2,EMADR	
12401	050744	111137	002176			MOVB (R1),AEDTA	
12402	050750	111237	002202			MOVB (R2),EMDTA	
12403	050754	005037	002056		TFORE:	CLR ERRS	:CLEAR ERROR INDICATOR
12404	050760	005737	002144			TST ERRCC	:IF COMPARISON HAS TURNED UP ERRORS
12405							: ENTER ERROR DISPLAY ROUTINE.
12406	050764	001020				BNE CISERR	
12407	050766	005737	002146			TST ERRREG	
12408	050772	001015				BNE CISERR	
12409	050774	005737	002150			TST ERRBUF	
12410	051000	001012				BNE CISERR	
12411	051002	005737	002152			TST ERRSTK	
12412	051006	001007				BNE CISERR	
12413	051010	005037	002272			CLR RPTFLG	:CLEAR THE REPEAT TEST FLAG
12414	051014	005737	002042			TST NOERDS	:DISPLAY EVEN THOUGH THERE WAS NO ERROR?
12415	051020	001017				BNE NOER	:BRANCH IF YES
12416	051022	000137	051326			JMP SWOPC	
12417							
12418	051026				CISERR:		
12419	051026	005237	002054			INC ERRCT	:INCREMENT ERROR COUNT









12573	051776	042701	177077			BIC #177077,R1	: BITS 6,7,8 OF PKPTW.
12574	052002	005301				ASL R1	
12575	052004	006301				ASL R1	
12576	052006	000301				SWAB R1	
12577	052010	010137	002446			MOV R1,S3TYPE	
12578							
12579	052014	005737	002452			TST MIXTYP	:MIX TYPES WITHIN INST?
12580	052020	001016				BNE 61\$	:BRANCH IF YES
12581	052022	023727	002454	000666		CMP PKPTW,#666	:ALL TYPES TESTED?
12582	052030	001327			2\$:	BNE TYPSET	:BRANCH IF NO
12583	052032	005037	002454		12\$:	CLR PKPTW	
12584	052036	005037	002442			CLR S1TYPE	
12585	052042	005037	002444			CLR S2TYPE	
12586	052046	005037	002446			CLR S3TYPE	
12587	052052	000137	052600			JMP CFINL	:EXIT TO TEST IN-LINE INST
12588							
12589	052056	023727	002456	000001	61\$:	CMP NDESC,#1	:IS THERE ONLY 1 DESC FOR THIS INST
12590	052064	001007				BNE 7\$	:BRANCH IF NO
12591	052066	013701	002454			MOV PKPTW,R1	
12592	052072	042701	177770			BIC #177770,R1	
12593	052076	020127	000006			CMP R1,#6	:ALL TYPES TESTED?
12594	052102	000752				BR 2\$	
12595							
12596	052104	023727	002456	000002	7\$:	CMP NDESC,#2	:ARE THERE 2 DESC FOR THIS INST?
12597	052112	001022				BNE 8\$	:BRANCH IF NO
12598	052114	013701	002454			MOV PKPTW,R1	
12599	052120	042701	177770			BIC #177770,R1	
12600	052124	020127	000006			CMP R1,#6	
12601	052130	001267				BNE TYPSET	
12602	052132	062737	000002	002454		ADD #2,PKPTW	
12603	052140	013701	002454			MOV PKPTW,R1	
12604	052144	042701	177707			BIC #177707,R1	
12605	052150	020127	000060			CMP R1,#60	:ALL TYPE MIXTURES TESTED?
12606	052154	001267				BNE 11\$	
12607	052156	000725				BR 12\$	
12608							
12609	052160	023727	002456	000003	8\$:	CMP NDESC,#3	:ARE THERE 3 DESC FOR THIS INST?
12610	052166	001401				BEQ 81\$	:BRANCH IF YES
12611	052170	000000				HALT	:# OF DESC FOR INST UNDER TEST DOES
12612							: NOT MAKE SENSE.
12613	052172	013701	002454		81\$:	MOV PKPTW,R1	
12614	052176	042701	177770			BIC #177770,R1	
12615	052202	020127	000006			CMP R1,#6	
12616	052206	001240				BNE TYPSET	
12617	052210	062737	000002	002454		ADD #2,PKPTW	
12618	052216	013701	002454			MOV PKPTW,R1	
12619	052222	042701	177707			BIC #177707,R1	
12620	052226	020127	000060			CMP R1,#60	
12621	052232	001240				BNE 11\$	
12622	052234	062737	000020	002454		ADD #20,PKPTW	
12623	052242	013701	002454			MOV PKPTW,R1	
12624	052246	042701	177077			BIC #177077,R1	
12625	052252	020127	000600			CMP R1,#600	:ALL TYPE MIXTURES TESTED?
12626	052256	001226				BNE 11\$	

12627	052260	000664				BR 12\$	
12628							
12629	052262	005237	002454			MIXDPZ: INC PKPTW	
12630	052266	005737	002452			TST MIXTYP	;MIX TYPES WITHIN INST?
12631	052272	001407				BEQ 1\$	;BRANCH IF NO
12632	052274	032737	000007	002454		BIT #7,PKPTW	;YES - IF PACK DATA TYPE=0,CHANGE IT TO 6.
12633	052302	001003				BNE 1\$	
12634	052304	052737	000006	002454		BIS #6,PKPTW	
12635	052312	013701	002454		1\$:	MOV PKPTW,R1	;SET STRING 1 DESC TYPE FROM
12636	052316	042701	177770			BIC #177770,R1	; BITS 0,1,2 OF PKPTW
12637	052322	010137	002442			MOV R1,S1TYPE	
12638							
12639	052326	013701	002454			MOV PKPTW,R1	;SET STRING 2 DESC TYPE FROM
12640	052332	042701	177707			BIC #177707,R1	; BITS 3,4,5 OF PKPTW
12641	052336	006201				ASR R1	
12642	052340	006201				ASR R1	
12643	052342	006201				ASR R1	
12644	052344	010137	002444			MOV R1,S2TYPE	
12645							
12646	052350	005737	002452			TST MIXTYP	;MIX TYPES WITHIN INST?
12647	052354	001016				BNE 61\$	;BRANCH IF YES
12648	052356	023727	002454	000010		CMP PKPTW,#10	;ALL TYPES TESTED?
12649	052364	001016			60\$:	BNE 62\$	;BRANCH IF NO
12650	052366	012737	000006	002454		MOV #06,PKPTW	
12651	052374	012737	000006	002442		MOV #6,S1TYPE	;INITIALIZE S1TYPE & S2TYPE FOR NEXT TEST CONDITION
12652	052402	005037	002444			CLR S2TYPE	
12653	052406	000137	052600			JMP CFINL	;EXIT TO TEST IN-LINE INST
12654							
12655	052412	023727	002454	000066	61\$:	CMP PKPTW,#66	
12656	052420	000761				BR 60\$	
12657	052422	000137	051710		62\$:	JMP TYPSET	
12658							
12659	052426	005237	002454			MIXDZP: INC PKPTW	
12660	052432	005737	002452			TST MIXTYP	;MIX TYPES WITHIN INST??
12661	052436	001412				BEQ 1\$	;BRANCH IF NO
12662	052440	013701	002454			MOV PKPTW,R1	;YES - IF ZONED DATA TYPE=6,CHANGE IT TO 0.
12663	052444	042701	177770			BIC #177770,R1	
12664	052450	020127	000006			CMP R1,#6	
12665	052454	001003				BNE 1\$	
12666	052456	062737	000002	002454		ADD #2,PKPTW	
12667	052464	013701	002454		1\$:	MOV PKPTW,R1	;SET STRING 1 DESC TYPE FROM
12668	052470	042701	177770			BIC #177770,R1	; BITS 0,1,2 OF PKPTW
12669	052474	010137	002442			MOV R1,S1TYPE	
12670							
12671	052500	013701	002454			MOV PKPTW,R1	;SET STRING 2 DESC TYPE FROM
12672	052504	042701	177707			BIC #177707,R1	; BITS 3,4,5 OF PKPTW
12673	052510	006201				ASR R1	
12674	052512	006201				ASR R1	
12675	052514	006201				ASR R1	
12676	052516	010137	002444			MOV R1,S2TYPE	
12677	052522	005737	002452			TST MIXTYP	;MIX TYPES WITHIN INST?
12678	052526	001016				BNE 61\$	;BRANCH IF YES
12679	052530	023727	002454	000066		CMP PKPTW,#66	;ALL TYPES TESTED?
12680	052536	001016			60\$:	BNE 62\$	;BRANCH IF NO

12681	052540	012737	000060	002454		MOV #60,PKPTW	
12682	052546	005037	002442			CLR S1TYPE	
12683	052552	012737	000006	002444		MOV #6,S2TYPE	
12684	052560	000137	052600			JMP CFINL	;EXIT TO TEST IN-LINE INST
12685							
12686	052564	022737	000100	002454	61\$:	CMP #100,PKPTW	
12687	052572	000761				BR 60\$	
12688	052574	000137	051710		62\$:	JMP TYPSET	
12689							
12690	052600	032737	000100	050012	CFINL:	BIT #100,TINST	;INST TYPE (REG OR IN-LINE)?
12691	052606	001022				BNE NXTTC	;BRANCH IF IN-LINE TYPE
12692	052610	005737	001760			TST RANDOM	;RANDOM EXERCISE MODE?
12693	052614	001407				BEQ 1\$	;BRANCH IF NO
12694	052616	004737	063532			JSR PC,RN	;GET A RANDOM #
12695	052622	042700	177770			BIC #177770,R0	;LOOK AT 3 LEAST SIGN BITS
12696	052626	020027	000003			CMP R0,#3	;IF THEY ARE = TO 3 (ARBITRARY CONSTANT) THEN DO IN-LINE
12697	052632	001010				BNE NXTTC	
12698	052634	052737	000100	050012	1\$:	BIS #100,TINST	;SWITCH REG OP-CODE AT TINST TO IN-LINE OP-CODE.
12699	052642	052737	000100	002132		BIS #100,FATAL	;SET IN-LINE FIELD IN FATAL ERROR INDICATOR WORD
12700	052650	000137	041324			JMP NTC	;REPEAT TEST CONDITION WITH IN-LINE OPCODE.
12701	052654	042737	000100	050012	NXTTC:	BIC #100,TINST	
12702	052662	042737	000100	002132		BIC #100,FATAL	;CLR IN-LINE FIELD IN FATAL ERROR INDICATOR
12703	052670	005737	042032			TST MTYPE	;IF TYPE = 0 THEN READY FOR NEXT INPUT TABLE ENTRY
12704	052674	001402				BEQ 1\$	;BRANCH IF TYPE NOT = 0
12705	052676	000137	053174			JMP RFNITE	
12706	052702	013737	001656	002126	1\$:	MOV IPNU,VIP	;SETUP A POINTER TO LAST PARAMETER IN
12707	052710	063737	001656	002126		ADD IPNU,VIP	; INPUT TABLE ENTRY.
12708	052716	013703	002126		UPTP:	MOV VIP,R3	
12709	052722	062703	001566			ADD #PTP,R3	;R3 POINTS TO PARAMETER TABLE POINTER
12710	052726	013701	002134			MOV INPTP,R1	
12711	052732	063701	002126			ADD VIP,R1	
12712	052736	062701	000002			ADD #2,R1	
12713	052742	005711				TST (R1)	; (R1) POINTS TO 1ST ENTRY IN PARAMERE TABLE
12714	052744	001464				BEQ TNXP	
12715	052746	017137	000000	002130		MOV @(R1),PTW1	;PTW1 CONTAINS TYPE,SIZE, AND # OF ENTRIES
12716							; IN PARAMETER TABLE
12717	052754	042737	177400	002130		BIC #177400,PTW1	;STRIP OFF TYPE AND ENTRY SIZE
12718	052762	122737	000001	002130		CMPB #1,PTW1	;# OF ENTRIES = 1?
12719	052770	001452				BEQ TNXP	;YES - NO UPDATING. TRY NEXT PARAMETER
12720	052772	005737	002130			TST PTW1	;# OF ENTRIES = 0?
12721	052776	001447				BEQ TNXP	;YES - NO UPDATING. TRY NEXT PARAMETER
12722	053000	017102	000000			MOV @(R1),R2	;PARAMETER TABLE CONTAINS MORE THAN 1 ENTRY.
12723	053004	042702	037777			BIC #037777,R2	;LOOK AT ENTRY TYPE
12724	053010	005702				TST R2	
12725	053012	001401				BEQ FLE	;FIXED LENGTH ENTRIES
12726	053014	000447				BR VLE	;VARIABLE LENGTH ENTRIES
12727							
12728	053016				FLE:		;PARAMETER TABLE CONTAINS FIXED
12729							; LENGTH ENTRIES.
12730	053016	017137	000000	002130		MOV @(R1),PTW1	
12731	053024	000337	002130			SWAB PTW1	
12732	053030	113704	002130			MOVB PTW1,R4	;LOAD R4 WITH ENTRY SIZE IN WORDS
12733	053034	006304				ASL R4	;CONVERT SIZE IN WORDS TO BYTES
12734	053036	005002				CLR R2	

12735	053040	000337	002130			SWAB PTW1		
12736	053044	042737	177400	002130		BIC #177400,PTW1		;PTW1 CONTAINS # OF ENTRIES IN
12737								; PARAMETER TABLE.
12738	053052	005337	002130		1\$:	DEC PTW1		;CALCULATE SIZE OF PARAMETER TABLE
12739	053056	005737	002130			TST PTW1		; SIZE = ENTRY SIZE X # OF ENTRIES
12740	053062	001402				BEQ GPTE SZ		;CALCULATION COMPLETE - R2 CONTAINS SIZE
12741								; OF PARAMETER TABLE (MINUS 1 ENTRY)
12742	053064	060402				ADD R4,R2		
12743	053066	000771				BR 1\$		
12744								
12745	053070						GPTE SZ:	
12746	053070	061102				ADD (R1),R2		;UPDATE R2 TO CONTAIN ADDRESS OF
12747	053072	062702	000002			ADD #2,R2		; LAST ENTRY IN PARAMETER TABLE
12748	053076	021302				CMP (R3),R2		
12749	053100	001403				BEQ RESPTP		;CURRENT PTP POINTS TO LAST ENTRY IN
12750								; PARAMETER TABLE. RESET TO 1ST ENTRY
12751	053102	060413				ADD R4,(R3)		;STILL MORE ENTRIES TO TRY IN PARAMETER
12752								; TABLE. UPDATE PTP TO NEXT ENTRY.
12753	053104	000137	041324			JMP NTC		;TRY NEXT TEST CONDITON FOR SAME CIS INST.
12754								
12755	053110						RESPTP:	;RESET PTP BACK TO 1ST ENTRY IN TABLE
12756	053110	011113				MOV (R1),(R3)		
12757	053112	062713	000002			ADD #2,(R3)		
12758	053116	162737	000002	002126		TNXP: SUB #2,VIP		;BACKUP TO NEXT PARAMETER TABLE POINTER
12759								; AND UPDATE IT.
12760	053124	005737	002126			TST VIP		
12761	053130	001421				BEQ RFNITE		;ALL TEST CONDITIONS ASSOCIATED WITH CURRENT
12762								; INPUT TABLE ENTRY EXERCISED.
12763	053132	000671				BR UPTP		
12764								
12765	053134						VLE:	;PARAMETER TABLE CONTAINS VARIABLE LENGTH ENTRIES
12766	053134	113704	002130			MOVB PTW1,R4		;R4 CONTAINS # OF ENTRIES IN PARAMETER TABLE
12767	053140	005304				DEC R4		;DETERMINE ADDRESS OF LAST ENTRY IN PARAMETER TABLE
12768	053142	011102				MOV (R1),R2		;R2 POINTS TO 1ST WORD IN PARAMETER TABLE
12769	053144	005722			1\$:	TST (R2)+		;SEARCH FOR 0 WORDS
12770	053146	001376				BNE 1\$		
12771	053150	005304				DEC R4		;AT LAST ENTRY YET?
12772	053152	001374				BNE 1\$		;NO
12773	053154	021302				CMP (R3),R2		;YES - IS CURRENT PTP POINTING TO LAST ENTRY?
12774	053156	001754				BEQ RESPTP		;YES - RESET PTP TO 1ST ENTRY
12775	053160	011304				MOV (R3),R4		;NO - UPDATE PTP TO NEXT ENTRY.
12776	053162	005724			2\$:	TST (R4)+		
12777	053164	001376				BNE 2\$		
12778	053166	010413				MOV R4,(R3)		
12779	053170	000137	041324			JMP NTC		;TRY NEXT TEST CONDITION FOR SAME INPUT
12780								; TABLE ENTRY.
12781								



```

12861 053434 000137 036760          JMP START
12863 053440          HERE:
12865 053440 005737 002074          TST FSRUN          ;NORMAL FIELD SERVICE RUN?
12866 053444 001445          BEQ NFS           ;BRANCH IF NO
12867 053446 005737 002206          TST QVMODE        ;RUNNING IN QV MODE
12868 053452 001411          BEQ 1$           ;BRANCH IF NO
12869 053454 005737 001750          TST N200M        ;PROG STARTED AT LOC 200?
12870 053460 001404          BEQ 2$           ;BRANCH IF NO
12871 053462 005037 002206          CLR QVMODE
12872 053466 000137 037320          JMP COMST
12873 053472 000137 036766          2$: JMP QVST
12874 053476 104400          1$: TYPE
12875 053500 016227          FSHDR2           ;PRINT ENTERING RANDOM MODE HEADER
12876 053502 104400          TYPE
12877 053504 016263          FSHDR3
12878 053506 104400          TYPE
12879 053510 016344          FSHDR4
12880 053512 104400          TYPE
12881 053514 016430          FSHDR5
12883 053516 012737 000001 001660          MOV #1,INCSQ1    ;INITIALIZE BUFFER INITIALIZATION CONSTANTS
12884 053524 012737 000002 001662          MOV #2,INCSQ2
12885 053532 012737 177777 001760          MOV #177777,RANDOM ;SET RANDOM MODE FLAG
12886 053540 012737 072144 072220          MOV #IDUM,INPTBL ;POINT TO DUMMY INPUT TABLE
12887 053546 013737 072220 002134          MOV INPTBL,INPT
12888 053554 000137 040706          JMP NITE
12889 053560 000137 037154          NFS: JMP DVTST
12890
12891 053564          FC00:           ;FLOW COMMAND = 00 -IGNORE ENTRY.
12892 053564 000137 042250          JMP FCRTN
12893
12894
12895          ;SUBROUTINES TO EXTRACT VARIOUS DATA FIELDS FROM FLOW TABLE ENTRY.
12896
12897 053570          PF1:           ;LOAD PTPTR WITH CONTANTS OF PTP
12898          ; SPECIFIED IN THE PF1 FIELD OF FLOW
12899          ; TABLE ENTRY.
12900 053570 017701 126146          MOV @FLOPTR,R1   ;FLOPTR POINTS TO FLOW TABLE ENTRY
12901 053574 032701 004000          BIT #004000,R1  ;SAVE STRING INSERTED FOR POSSIBLE ERROR PRINTOUT?
12902 053600 0C1405          BEQ 1$
12903 053602 042701 004000          BIC #004000,R1  ;YES
12904 053606 012737 177777 002476          MOV #177777,SAVSRF ;SET SAVE STRING FLAG
12905 053614 042701 170077          1$: BIC #170077,R1 ;STRIP OFF ALL BUT DESIRED FIELD
12906 053620 006301          ASL R1
12907 053622 006301          ASL R1
12908 053624 000301          SWAB R1         ;ADJUST DESIRED FIELD TO BIT 0
12909 053626 006301          ASL R1
12910 053630 062701 001566          ADD #PTP,R1     ;ADD # IN FIELD TO TOP OF PTP TABLE
12911 053634 011137 002136          MOV (R1),PTPTR ;SAVE CONTENTS OF DESIRED PTP IN PTPTR
12912 053640 000207          RTS PC
12913
12914 053642          PF2:           ;SAME AS PF1 SUBROUTINE EXCEPT FOR FIELD.
12915 053642 017701 126074          MOV @FLOPTR,R1
12916 053646 042701 177700          BIC #177700,R1
12917 053652 006301          ASL R1
  
```



12918	053654	062701	001566	ADD #PTP,R1	
12919	053660	011137	002136	MOV (R1),PTPTR	
12920	053664	000207		RTS PC	
12921					
12922	053666			RF1:	;LOAD R1 WITH ADDRESS OF TEST
12923					; OPERAND SPECIFIED.
12924	053666	004737	053700	JSR PC,RF1X	;GET FIELD ONE CONTENTS
12925	053672	062701	003630	ADD #TRN,R1	;ADD IN ADDRESS OF TEST OPERAND TABLE
12926	053676	000207		RTS PC	
12927					
12928	053700			RF1X:	;GET FIELD ONE CONTENTS FROM FLOW TABLE ENTRY
12929	053700	017701	126036	MOV @FLOPTR,R1	;FLOPTR POINTS TO FLOW TABLE ENTRY
12930	053704	042701	170777	BIC #170777,R1	;STRIP OFF ALL BUT DESIRED FIELD
12931	053710	006201		ASR R1	
12932	053712	000301		SWAB R1	;RIGHT ADJUST FIELD
12933	053714	006301		ASL R1	
12934	053716	000207		RTS PC	
12935					
12936	053720			RF2:	;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD.
12937	053720	004737	053732	JSR PC,RF2X	
12938	053724	062701	003630	ADD #TRN,R1	
12939	053730	000207		RTS PC	
12940					
12941	053732			RF2X:	;SAME AS RF2X SUBROUTINE EXCEPT FOR FIELD.
12942	053732	017701	126004	MOV @FLOPTR,R1	
12943	053736	042701	177077	BIC #177077,R1	
12944	053742	006301		ASL R1	
12945	053744	006301		ASL R1	
12946	053746	000301		SWAB R1	
12947	053750	006301		ASL R1	
12948	053752	000207		RTS PC	
12949					
12950	053754			RF3:	;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD
12951	053754	004737	053766	JSR PC,RF3X	
12952	053760	062701	003630	ADD #TRN,R1	
12953	053764	000207		RTS PC	
12954					
12955	053766			RF3X:	;SAME AS RF1X SUBROUTINE EXCEPT FOR FIELD
12956	053766	017701	125750	MOV @FLOPTR,R1	
12957	053772	042701	177707	BIC #177707,R1	
12958	053776	006201		ASR R1	
12959	054000	006201		ASR R1	
12960	054002	000207		RTS PC	
12961					
12962	054004			RF4:	;SAME AS RF1 SUBROUTINE EXCEPT FOR FIELD
12963	054004	004737	054016	JSR PC,RF4X	
12964	054010	062701	003630	ADD #TRN,R1	
12965	054014	000207		RTS PC	
12966					
12967	054016			RF4X:	;SAME AS RF1X SUBROUTINE EXCEPT FOR FIELD
12968	054016	017701	125720	MOV @FLOPTR,R1	
12969	054022	042701	177770	BIC #177770,R1	
12970	054026	006301		ASL R1	
12971	054030	000207		RTS PC	

12972										
12973	054032					RFNX:				:SUBROUTINE TO RETURN CONTENTS OF FLOW
12974										: TABLE ENTRY FIELD POINTED TO BY THE
12975										: CONTENTS OF NXFLD+1.
12976	054032	005237	002232							
12977	054036	022737	000001	002232						:FIELD REQUESTED = 1?
12978	054044	001003								:BRANCH IF NO
12979	054046	004737	053700							:GET CONTENTS OF FIELD ONE.
12980	054052	000426								:EXIT
12981	054054	022737	000002	002232	1\$:					:FIELD REQUESTED=2?
12982	054062	001003								:BRANCH IF NO
12983	054064	004737	053732							:GET CONTENTS OF FIELD 2
12984	054070	000417								:EXIT
12985	054072	022737	000003	002232	2\$:					:FIELD REQUESTED = 3?
12986	054100	001003								:BRANCH IF NO
12987	054102	004737	053766							:GET CONTENTS OF FIELD 3
12988	054106	000410								:EXIT
12989	054110	022737	000004	002232	3\$:					:FIELD REQUESTED = 4?
12990	054116	001003								:BRANCH IF NO
12991	054120	004737	054016							:GET CONTENTS OF FIELD 4
12992	054124	000401								:EXIT
12993	054126	005001			9\$:					:FIELD REQUESTED INVALID - RETURN 0.
12994	054130	000207			10\$:					
12995										
12996	054132					RFN:				:SUBROUTINE TO RETURN CONTENTS OF FLOW
12997										: TABLE ENTRY FIELD PLUS TEST OPERAND TABLE
12998										: OFFSET POINTED TO BY THE
12999										: CONTENTS OF NXFLD+1.
13000	054132	005237	002232							
13001	054136	022737	000001	002232						:FIELD REQUESTED = 1?
13002	054144	001003								:BRANCH IF NO
13003	054146	004737	053666							:GET CONTENTS OF FIELD ONE PLUS TRN OFFSET.
13004	054152	000427								:EXIT
13005	054154	022737	000002	002232	1\$:					:FIELD REQUESTED=2?
13006	054162	001003								:BRANCH IF NO
13007	054164	004737	053720							:GET CONTENTS OF FIELD 2 PLUS TRN OFFSET.
13008	054170	000420								:EXIT
13009	054172	022737	000003	002232	2\$:					:FIELD REQUESTED = 3?
13010	054200	001003								:BRANCH IF NO
13011	054202	004737	053754							:GET CONTENTS OF FIELD 3 PLUS TRN OFFSET.
13012	054206	000411								:EXIT
13013	054210	022737	000004	002232	3\$:					:FIELD REQUESTED = 4?
13014	054216	001003								:BRANCH IF NO
13015	054220	004737	054004							:GET CONTENTS OF FIELD 4 PLUS TRN OFFSET.
13016	054224	000402								:EXIT
13017	054226	012701	003630		9\$:					:FIELD REQUESTED INVALID - RETURN #TRN.
13018	054232	000207			10\$:					
13019										
13020	054234					FC15:				:FLOW COMMAND = 15 - SETUP SPECIAL HANDLING WORD
13021										: BIT 0 = 1 MEANS SKIP 04 FLOW COMMANDS
13022										: BIT 1 = 1 MEANS 07 FLOW COMMAND IGNORE BIT 15
13023										: OF STRING LENGTH
13024										: BIT 2 = 1 MEANS DON'T CHECK BUFFER RESULTS
13025	054234	004737	053570							:FORM PARAMETER TABLE POINTER TO SPECIAL HANDLING REQUES

13026	054240	017737	125672	002140	MOV @PTPTR,SPHAND	;COPY SPECIAL HANDLING REQUEST INTO SPECIAL HANDLING WOR
13027	054246	000137	042250		JMP FCRTN	
13028						

```

13030 .SBTTL          SETUP MACHINE DEPENDENT CONSTANTS
13031 :
13032 :;SIZE FOR PROCESSOR TYPE - SETUP MACHINE DEPENDENT CONSTANTS BASED UPON RESULT.
13033 :
13034 :SIZEPT:
13035 054252 012737 054562 000010      MOV #3$,@#RESVEC      ;SETUP RESERVED INST TRAP VECTOR
13036 :                                     ; IN CASE MACHINE UNDER TEST DOES NOT HAVE MFPT
13037 054260 005037 000012      CLR @#RESVEC+2
13038 054264 000007      MFPT                  ;WHAT IS THE PROCESSOR TYPE?
13039 054266 123700 001710      CMPB EL2324,RO
13040 054272 001525      BEQ 6$                ;BRANCH IF PROCESSOR = 11/23 OR 11/24
13041 054274 123700 001704      CMPB EL74,RO
13042 054300 001025      BNE 1$                ;BRANCH IF NOT AN 11/74 TYPE PROCESSOR
13043 054302 005737 001122      TST $PASS
13044 054306 001002      BNE 10$
13045 054310 104400      TYPE                  ;TYPE 11/74 TYPE PROCESSOR (1ST PASS ONLY)
13046 054312 017002      MPT74
13047 054314 012737 177777 002156 10$:  MOV #177777,MMFLG      ;SET MEM MANAGEMENT FLAG SIGNALING THAT SYS
13048 :                                     ; UNDER TEST HAS 11/70 TYPE MEM MGMT
13049 054322 012737 000003 002164      MOV #3,NMODES        ;PROC UNDER TEST HAS 3 MODES (K,S,U)
13050 054330 012737 177777 002060      MOV #177777,TWOSET   ;PROCESSOR UNDER TEST HAS 2 REGISTER SETS
13051 054336 012737 000002 002212      MOV #2,PTQV          ;INPUT TABLE ENTRY TYPE WORD BIT 1
13052 :                                     ; IDENTIFIES TABLE RUN IN QV MODE FOR 11/74
13053 :                                     ; (1=RUN,0=SKIP)
13054 054344 012737 000003 003036      MOV #3,LTCDLY        ;INITIALIZE COUNTER USED BY LINE TIME CLOCK
13055 :                                     ; ROUTINES - COUNT IS A MEASURE OF
13056 :                                     ; TIME REMAINING BEFORE INTERRUPT FROM
13057 :                                     ; LTC WHEN STARTING EXECUTION OF CIS
13058 :                                     ; INSTRUCTION UNDER TEST.
13059 054352 000566      BR 4$
13060 054354 123700 001706      1$:  CMPB EL44,RO          ;IS THIS AN 11/44?
13061 054360 001024      BNE 2$                ;BRANCH IF NO
13062 054362 005737 001122      TST $PASS
13063 054366 001002      BNE 12$
13064 054370 104400      TYPE                  ;TYPE 11/44 PROCESSOR (1ST PASS ONLY)
13065 054372 016760      MPT44
13066 054374 012737 177777 002156 12$:  MOV #177777,MMFLG      ;SET MEM MGMT FLAG
13067 054402 012737 000003 002164      MOV #3,NMODES        ;PROC UNDER TEST HAS 3 MODES
13068 054410 005037 002060      11$:  CLR TWOSET           ;INITIALIZE FOR SINGLE REGISTER SET
13069 054414 012737 000002 002212      MOV #2,PTQV          ;INPUT TABLE ENTRY TYPE WORD BIT 2
13070 :                                     ; IDENTIFIES TABLE RUN IN QV MODE FOR 11/44
13071 054422 012737 000013 003036      MOV #13,LTCDLY       ;INITIALIZE COUNTER USED FOR LTC
13072 054430 000537      BR 4$
13073 054432 005037 002156      2$:  CLR MMFLG
13074 054436 005737 001122      TST $PASS
13075 054442 001002      BNE 13$
13076 054444 104400      TYPE                  ;TYPE NO MEM MGMT AVAILABLE (1ST PASS ONLY)
13077 054446 017064      NMM
13078 054450 012737 000240 060110 13$:  MOV #NOP,PCIS2        ;OVERWRITE ACCESS TO MEM MGMT REGISTERS
13079 054456 012737 000240 060112      MOV #NOP,PCIS2+2
13080 054464 012737 000240 060134      MOV #NOP,PCIS1
13081 054472 012737 000240 060136      MOV #NOP,PCIS1+2
13082 054500 012737 000405 062044      MOV #405,LTCIS
13083 054506 012737 000240 057202      MOV #NOP,ILLSER

```

13084	054514	012737	000240	057204		MOV #NOP,ILLSER+2	
13085	054522	012737	000240	057062		MOV #NOP,HLTSER	
13086	054530	012737	000240	057064		MOV #NOP,HLTSER+2	
13087	054536	012737	000002	002164		MOV #2,NMODES	
13088	054544	000721				BR 11\$	
13089							
13090							
13091							
13092	054546	005737	001122		6\$:	TST \$PASS	
13093	054552	001005				BNE 7\$	
13094	054554	104400				TYPE	
13095	054556	017031				MPT2324	
13096	054560	000402				BR 7\$	
13097	054562	005726			3\$:	TST (SP)+	
13098	054564	005726				TST (SP)+	
13099	054566	012737	054634	000004	7\$:	MOV #33\$,@#ERRVEC	
13100	054574	005737	172340			TST @#KIPARO	
13101	054600	005737	177640			TST @#UIPARO	
13102	054604	012737	054642	000004		MOV #34\$,@#ERRVEC	
13103	054612	005737	172240			TST @#SIPARO	
13104	054616	000705				BR 2\$	
13105	054620	012737	054650	000004	5\$:	MOV #35\$,@#ERRVEC	
13106	054626	005737	172360			TST @#KDPARO	
13107	054632	000677				BR 2\$	
13108	054634	005726			33\$:	TST (SP)+	
13109	054636	005726				TST (SP)+	
13110	054640	000674				BR 2\$	
13111	054642	005726			34\$:	TST (SP)+	
13112	054644	005726				TST (SP)+	
13113	054646	000764				BR 5\$	
13114	054650	005726			35\$:	TST (SP)+	
13115	054652	005726				TST (SP)+	
13116	054654	005737	001122			TST \$PASS	
13117	054660	001002				BNE 14\$	
13118	054662	104400				TYPE	
13119	054664	016675				MPT34	
13120	054666	012737	000002	002164	14\$:	MOV #2,NMODES	
13121	054674	012737	177777	002156		MOV #177777,MMFLG	
13122	054702	005037	002060			CLR TWOSET	
13123	054706	012737	000002	002212		MOV #2,PTQV	
13124	054714	012737	177777	002154		MOV #177777,PT34	
13125	054722	012737	000013	003036		MOV #13,LTC DLY	
13126	054730	012737	057202	000010	4\$:	MOV #ILLSER,@#RESVEC	
13127	054736	012737	057062	000004		MOV #HLTSER,@#ERRVEC	
13128	054744	000207				RTS PC	

;MACHINE TYPE UNKNOWN - DEFUAULT TO SINGLE  
; REG SET ,NO MEM MGMT, 2 PROC MODES (K & U)  
; AND A LTC DELAY OF 13  
;(NEW MACHINES MAY REQUIRE CHANGES HERE)

;TYPE 11/23 OR 11/24 PROCESSOR (1ST PASS ONL/)

;FIX UP STACK

;DOES MACHINE UNDER TEST HAVE (11/34 TYPE) MEM MGMT?  
;IF SO NO TRAP HERE  
;IF SO NO TRAP HERE

;IF SO, TRAP HERE

;IF SO, TRAP HERE

;FIX UP STACK

;FIX UP STACK

;FIX UP STACK

;TYPE 'MEMORY MANAGEMENT ON SYS UNDER TEST'(1ST PASS ONI

;PROC = 11/23 OR 11/24 OR 11/34

;RESTORE RESERVED INST TRAP CATCHER  
;RESTORE HALT SERVICE TRAP CATCHER

```

13130          .SBTTL          REGISTER SET SELECTION ROUTINES
13131
13132          ;
13133          ;IF PROCESSOR UNDER TEST HAS 2 REGISTER SETS
13134          ;SELECT REGISTER SET TO BE USED BASED ON LEAST SIGNIFICANT BIT OF
13135          ;TEST # (TOTTC), AND LOAD REGISTER SET WHICH WAS NOT
13136          ;SELECTED WITH A FIXED PATTERN <NOTREG>.
13137          ;
13138          ;
13139          054746          SELREG:
13140          054746 005737 002060          TST TWOSET          ;DOES MACHINE UNDER TEST HAVE 2 REG SETS?
13141          054752 001506          BEQ 1$          ;BRANCH IF NO
13142          054754 032737 000001 001420          BIT #1,TOTTC          ;PROCESSOR IS AN 11/74 - HAS 2 REG SETS;USE
13143          ;          ; LEAST SIGNIF. BIT OF TEST COUNT TO LOAD
13144          ;          ; PSW BIT 11 (REG SET BIT).
13145          054762 001041          BNE 2$          ;BRANCH TO USE REGISTER SET 1.
13146          054764 052777 004000 124672          BIS #4000,@TPSW          ;CIS INST WILL BE TESTED USING GPR SET 0.
13147          054772 042737 001000 002132          BIC #1000,FATAL          ;INDICATE GPR SET 0 IN FATAL ERROR WORD
13148          ;          ;LOAD SET 1 WITH PATTERN IN NOTREG.
13149          055000 013700 001702          MOV NOTREG,R0
13150          055004 013701 001702          MOV NOTREG,R1
13151          055010 013702 001702          MOV NOTREG,R2
13152          055014 013703 001702          MOV NOTREG,R3
13153          055020 013704 001702          MOV NOTREG,R4
13154          055024 013705 001702          MOV NOTREG,R5
13155          055030 042777 004000 124626          BIC #4000,@TPSW          ;SET REGISTER SET TO 0
13156          055036 042777 004000 125452          BIC #4000,@PCLK1P          ;ASSURE CORRECT REGISTER USAGE ON INTERRUPT
13157          055044 042777 004000 125752          BIC #4000,@LTCIP
13158          055052 042737 004000 000006          BIC #4000,@#6
13159          055060 005037 002052          CLR REGSET          ;SET REG SET INDICATOR TO 0
13160          055064 000441          BR 1$          ;BRANCH TO RETURN
13161          055066 042777 004000 124570 2$:          BIC #4000,@TPSW          ;CIS INST WILL BE TESTED USING GPR SET 1.
13162          055074 052737 001000 002132          BIS #1000,FATAL          ;INDICATE GPR SET 1 IN FATAL ERROR WORD
13163          ;          ;LOAD SET 0 WITH PATTERN IN NOTREG.
13164          055102 013700 00 702          MOV NOTREG,R0
13165          055106 013701 001702          MOV NOTREG,R1
13166          055112 013702 001702          MOV NOTREG,R2
13167          055116 013703 001702          MOV NOTREG,R3
13168          055122 013704 001702          MOV NOTREG,R4
13169          055126 013705 001702          MOV NOTREG,R5
13170          055132 052777 004000 124524          BIS #4000,@TPSW          ;SET REGISTER SET TO 1
13171          055140 052777 004000 125350          BIS #4000,@PCLK1P          ;ASSURE CORRECT REGISTER USAGE ON INTERRUPT
13172          055146 052777 004000 125650          BIS #4000,@LTCIP
13173          055154 052737 004000 000006          BIS #4000,@#6
13174          055162 012737 000001 002052          MOV #1,REGSET          ;SET REG SET INDICATOR TO 1
13175          055170 000207          1$:          RTS PC
13176          ;
13177          ;
13178          ;ROUTINE TO VERIFY THAT REGISTER SET WHICH WAS NOT SELECTED
13179          ;(PROVIDED PROCESSOR UNDER TEST HAS 2) DID NOT GET CHANGED, AND
13180          ;SWITCH TO REGISTER SET 0.
13181          ;
13182          ;
13183          055172          CKUREG:

```

```

13184 055172 005737 002060          TST TWOSET          ;PROCESSOR UNDER TEST HAVE TWO REGISTER SETS?
13185 055176 001435          BEQ 1$              ;BRANCH IF NO
13186 055200 005737 002052          TST REGSET         ;DETERMINE WHICH REG SET WAS IN USE
13187 055204 001426          BEQ 2$              ;BRANCH IF REG SET 0 WAS USED
13188 055206 042777 004000 124450  BIC #4000,@TPSW    ;VERIFY CONTENTS OF REG SET 0
13189 055214 020037 001702          CMP R0,NOTREG
13190 055220 001030          BNE 4$
13191 055222 020137 001702          CMP R1,NOTREG
13192 055226 001025          BNE 4$
13193 055230 020237 001702          CMP R2,NOTREG
13194 055234 001022          BNE 4$
13195 055236 020337 001702          CMP R3,NOTREG
13196 055242 001017          BNE 4$
13197 055244 020437 001702          CMP R4,NOTREG
13198 055250 001014          BNE 4$
13199 055252 020537 001702          CMP R5,NOTREG
13200 055256 001011          BNE 4$
13201 055260 000404          BR 1$
13202 055262 052777 004000 124374 2$:  BIS #4000,@TPSW    ;VERIFY CONTENTS OF REG SET 1
13203 055270 000751          BR 3$
13204 055272 042777 004000 124364 1$:  BIC #4000,@TPSW    ;SET REG SET TO ZERO
13205 055300 000207          RTS PC              ;RETURN
13206 055302          4$:
(6) 055302 012746 012167          MOV #HLTMSG,-(SP)
(3) 055306 010600          MOV SP,R0
(4) 055310 004737 065410          JSR PC,FPRINT
13207 055314 012737 055376 002170  MOV #100$,HLTLOC
13208 055322 004737 055402          JSR PC,IDINFO      ;IDENTIFY FAILING INST
13209 055326          PRINTB #FORM38,REGSET
(7) 055326 013746 002052          MOV REGSET,-(SP)
(6) 055332 012746 014415          MOV #FORM38,-(SP)
(3) 055336 010600          MOV SP,R0
(4) 055340 004737 065410          JSR PC,FPRINT
13210 055344          PRINTB #FORM39,NOTREG,R0,R1,R2,R3,R4,R5
(13) 055344 010546          MOV R5,-(SP)
(12) 055346 010446          MOV R4,-(SP)
(11) 055350 010346          MOV R3,-(SP)
(10) 055352 010246          MOV R2,-(SP)
(9) 055354 010146          MOV R1,-(SP)
(8) 055356 010046          MOV R0,-(SP)
(7) 055360 013746 001702          MOV NOTREG,-(SP)
(6) 055364 012746 014474          MOV #FORM39,-(SP)
(3) 055370 010600          MOV SP,R0
(4) 055372 004737 065410          JSR PC,FPRINT
13211 055376 000000          100$: HALT          ;REGISTER SET ERROR;PRESS CONTINUE SWITCH TO CONTINUE
13212          ; TESTING
13213 055400 000734          BR 1$
13214
13215 055402          IDINFO: PRINTB #TRPINF,HLTLOC,MODE,DEN,TINST,TOTTCH,TOTTC
(12) 055402 013746 001420          MOV TOTTC,-(SP)
(11) 055406 013746 001416          MOV TOTTCH,-(SP)
(10) 055412 013746 050012          MOV TINST,-(SP)
(9) 055416 013746 002162          MOV DEN,-(SP)
(8) 055422 013746 002160          MOV MODE,-(SP)

```

(7)	055426	013746	002170	MOV	HLTLOC,-(SP)
(6)	055432	012746	012304	MOV	#PRINF,-(SP)
(3)	055436	010600		MOV	SP,R0
(4)	055440	004737	065410	JSR	PC,FPRINT
13216	055444	000207		RTS	PC
13217					



```

13219
13220
13221
13222
13223
13224
13225
13226
13227 055446
13228 055446 005737 002156
13229 055452 001002
13230 055454 000137 056132
13231 055460 005737 002154
13232
13233 055464 001142
13234 055466 005037 172516
13235 055472 012737 000000 177660
13236 055500 012737 000200 177662
13237 055506 012737 000400 177664
13238 055514 012737 000600 177666
13239 055522 012737 001000 177670
13240 055530 012737 001200 177672
13241 055536 012737 001400 177674
13242 055544 012737 177600 177676
13243
13244 055552 012737 000000 172240
13245 055560 012737 000200 172242
13246 055566 012737 000400 172244
13247 055574 012737 000600 172246
13248 055602 012737 001000 172250
13249 055610 012737 001200 172252
13250 055616 012737 001400 172254
13251 055624 012737 177600 172256
13252
13253 055632 012737 000000 172260
13254 055640 012737 000200 172262
13255 055646 012737 000400 172264
13256 055654 012737 000600 172266
13257 055662 012737 001000 172270
13258 055670 012737 001200 172272
13259 055676 012737 001400 172274
13260 055704 012737 177600 172276
13261
13262 055712 012737 000000 172360
13263 055720 012737 000200 172362
13264 055726 012737 000400 172364
13265 055734 012737 000600 172366
13266 055742 012737 001000 172370
13267 055750 012737 001200 172372
13268 055756 012737 001400 172374
13269 055764 012737 177600 172376
13270
13271 055772 012737 000000 177640
13272 056000 012737 000200 177642

```

.SBTTL MEMORY MANAGEMENT SUBROUTINES

:MEMORY MANAGEMENT SUBROUTINES

:SETUP PAR'S

SETPAR:

2\$:

3\$:

```

TST MMFLG
BNE 2$
JMP 1$
TST PT34
BNE 3$
CLR @MMR3
MOV #0,@UDPAR0
MOV #200,@UDPAR1
MOV #400,@UDPAR2
MOV #600,@UDPAR3
MOV #1000,@UDPAR4
MOV #1200,@UDPAR5
MOV #1400,@UDPAR6
MOV #177600,@UDPAR7

MOV #0,@SIPAR0
MOV #200,@SIPAR1
MOV #400,@SIPAR2
MOV #600,@SIPAR3
MOV #1000,@SIPAR4
MOV #1200,@SIPAR5
MOV #1400,@SIPAR6
MOV #177600,@SIPAR7

MOV #0,@SDPAR0
MOV #200,@SDPAR1
MOV #400,@SDPAR2
MOV #600,@SDPAR3
MOV #1000,@SDPAR4
MOV #1200,@SDPAR5
MOV #1400,@SDPAR6
MOV #177600,@SDPAR7

MOV #0,@KDPAR0
MOV #200,@KDPAR1
MOV #400,@KDPAR2
MOV #600,@KDPAR3
MOV #1000,@KDPAR4
MOV #1200,@KDPAR5
MOV #1400,@KDPAR6
MOV #177600,@KDPAR7

MOV #0,@UIPAR0
MOV #200,@UIPAR1

```

```

;SETUP PAR'S FOR USER,SUPV, AND KERNEL I & D SPACES
;DOES SYSTEM UNDER TEST HAVE MEMORY MANAGEMENT?
;BRANCH IF YES

```

```

;IS THIS AN 11/34 TYPE PROCESSOR
;(I.E. K, U MODES AND 18 BIT MEM MGMT)
;BRANCH IF YES
;CLEAR OUT D-SPACE ENABLES
;SETUP USER D PAR'S

```

:SETUP SUPERVISOR I PAR'S

:SETUP SUPERVISOR D PAR'S

:SETUP KERNEL D PAR'S

:SETUP USER I PAGE ADDRESS REGISTERS

```

13273 056006 012737 000400 177644      MOV #400,@#UIPAR2
13274 056014 012737 000600 177646      MOV #600,@#UIPAR3
13275 056022 012737 001000 177650      MOV #1000,@#UIPAR4
13276 056030 012737 001200 177652      MOV #1200,@#UIPAR5
13277 056036 012737 001400 177654      MOV #1400,@#UIPAR6
13278 056044 012737 177600 177656      MOV #177600,@#UIPAR7
13279
13280 056052 012737 000000 172340      MOV #0,@#KIPAR0      ;SETUP KERNEL I PAR'S
13281 056060 012737 000200 172342      MOV #200,@#KIPAR1
13282 056066 012737 000400 172344      MOV #400,@#KIPAR2
13283 056074 012737 000600 172346      MOV #600,@#KIPAR3
13284 056102 012737 001000 172350      MOV #1000,@#KIPAR4
13285 056110 012737 001200 172352      MOV #1200,@#KIPAR5
13286 056116 012737 001400 172354      MOV #1400,@#KIPAR6
13287 056124 012737 177600 172356      MOV #177600,@#KIPAR7
13288
13289 056132 000207      1$:      RTS PC
13290
13291      ;
13292      ;SELECT MODE AND D-SPACE ENABLE/DISABLE
13293      ;
13294      ;BOTH MODE AND D-SPACE ENABLE ARE SELECTED RANDOMLY FOR EACH TEST
13295      ;IF EXECUTING IN RANDOM MODE OR IN A NORMAL FIELD SERVICE TYPE RUN.
13296      ;IF IN A DESIGN VERIFICATION TYPE RUN (ST @ 204) AND NOT RANDOM MODE
13297      ;THEN THESE VARIABLES WERE SET VIA OPERATOR DIAGLOG ABOVE.
13298      ;REGARDLESS OF THE RUN TYPE, IF MEMORY MANAGEMENT IS AVAILABLE
13299      ;(MMFLG=NONZERO) THEN MMR3 IS LOADED TO PROPER D-SPACE STATE.
13300      ;
13301      SELMD:
13302 056134 005737 002074      TST FSRUN      ;NORMAL FFIELD SERVICE TYPE RUN?
13303 056140 001003      BNE 10$      ;BRANCH IF YES
13304 056142 005737 001760      TST RANDOM      ;RANDOM TESTING?
13305 056146 001443      BEQ 11$      ;BRANCH IF NO
13306 056150 022737 000001 002164 10$:      CMP #1,NMODES      ;DOES SYSTEM UNDER TEST HAVE MORE THAN ONE
;      ; PROCESSOR MODE?
13307      BNE 2$      ;BRANCH IF YES
13308 056156 001003      CLR MODE
13309 056160 005037 002160      BR 4$
13310 056164 000422
13311 056166 004737 063532      2$:      JSR PC,RN      ;GENERATE A RANDOM #
13312 056172 042700 177774      BIC #177774,R0      ;USE BITS 1 & 0 TO SELECT MODE
;      ; (I.E 00=KERNEL,01=SUPV,11=USER)
13313
13314 056176 022700 000002      CMP #2,R0      ;ILLEGAL MODE (10)?
13315 056202 001771      BEQ 2$      ;YES - TRY AGAIN
13316 056204 022737 000002 002164      CMP #2,NMCMODES      ;DOES SYSTEM UNDER TEST HAVE ONLY 2 PROC MODES?
13317 056212 001005      BNE 6$      ;BRANCH IF NO
13318 056214 022700 000001      CMP #1,R0      ;ON 2 MODE MACHINE MODES ASSUMED TO BE
;      ; KERNEL & USER
13319      BNE 6$      ;BRANCH IF MODE IS LEGAL
13320 056220 001002
13321 056222 052700 000002      BIS #2,R0      ;IF RANDOM MODE = SUPV THEN SWITCH IT TO USER
13322 056226 010037 002160      6$:      MOV R0,MODE      ;MODE VALID - SET INTO MODE WORD
13323 056232 005037 002162      4$:      CLR DEN
13324 056236 004737 063532      JSR PC,RN      ;GENERATE A RANDOM #
13325 056242 032700 000001      BIT #1,R0      ;USE BIT 0 OF RANDOM NUMBER TO SELECT D ENABLE
13326 056246 001403      BEQ 11$      ;BRANCH TO DISABLE D SPACE

```



```

13381          ;SETUP KERNEL MODE PDR'S
13382
13383 056462 012737 077406 172316      MOV #77406,@#KIPDR7      ;ALLOW R/W ACCESS OF I/O PAGE
13385 056470 012737 077402 172310      MOV #77402,@#KIPDR4      ; AND R ONLY ACCESS OF PHYSICAL ADDRESS
13386 056476 012737 077402 172306      MOV #77402,@#KIPDR3      ; 20 TO 120K
13388 056504 012737 077402 172304      MOV #77402,@#KIPDR2
13389 056512 012737 077402 172302      MOV #77402,@#KIPDR1
13390 056520 005737 002160      TST MODE                ;IS MODE = KERNEL?
13391 056524 001404      BEQ 31$                 ;BRANCH IF YES
13392 056526 012737 077406 172300      MOV #77406,@#KIPDR0
13393 056534 000403      BR 32$
13394 056536 012737 077402 172300 31$:  MOV #77402,@#KIPDR0
13395 056544 005737 002160 32$:  TST MODE                ;IF MODE= KERNEL & D-SPACE IS DISABLED
13396 056550 001021      BNE 5$                 ; THEN ALLOW R/W OF STACK & TEST BUFFER AREA
13397 056552 005737 002162      TST DEN
13398 056556 001004      BNE 4$
13400 056560 012737 077406 172312      MOV #77406,@#KIPDR5
13405 056566 000455      BR 11$
13406 056570 012737 077406 172336 4$:  MOV #77406,@#KDPDR7      ;IF MODE IS KERNEL & D-SPACE IS ENABLED
13408 056576 012737 077406 172332      MOV #77406,@#KDPDR5      ; THEN SETUP KERNEL D-SPACE PDRS
13413 056604 012737 077402 172320      MOV #77402,@#KDPDR0      ;ALLOW R/W ACCESS OF I/O PAGE AND TEST BUFFER
13414          ; AREA; R-ONLY ACCESS OF PHYS 0-20K
13418 056612 000443      BR 11$
13419 056614 022737 000001 002160 5$:  CMP #1,MODE              ;SETUP SUPERVISOR MODE PDR'S
13420 056622 001016      BNE 7$                 ;BRANCH IF TEST MODE IS NOT SUPERVISOR
13421 056624 012737 077402 172204      MOV #77402,@#SIPDR2      ;ALLOW R ONLY ACCESS OF SPACE INCLUDING
13422          ; CIS INST TO BE EXECUTED
13426 056632 005737 002162      TST DEN                  ;IF D-SPACE IS NOT ENABLED ALLOW
13427 056636 001004      BNE 6$                 ; R/W ACCESS OF TEST BUFFER AREA IN SUPERVISOR
13428          ; I-SPACE
13430 056640 012737 077406 172212      MOV #77406,@#SIPDR5
13435 056646 000425      BR 11$
13437 056650 012737 077406 172232 6$:  MOV #77406,@#SDPDR5      ;D-SPACE IS ENABLED; ALLOW R/W ACCESS
13442 056656 000421      BR 11$                 ; TO TEST BUFFER AREA IN SUPERVISOR D-SPACE
13443 056660 022737 000003 002160 7$:  CMP #3,MODE              ;SETUP USER MODE PDR'S
13444 056666 001015      BNE 11$                 ;BRANCH IF TEST MODE IS NOT USER
13445 056670 012737 077402 177604      MOV #77402,@#UIPDR2      ;ALLOW R ONLY ACCESS OF SPACE INCLUDING
13446          ; CIS INST TO BE EXECUTED
13450 056676 005737 002162      TST DEN                  ;IF D-SPACE IS NOT ENABLED ALLOW R/W
13451 056702 001004      BNE 10$                 ; ACCESS OF TEST BUFFER AREA IN USER
13453 056704 012737 077406 177612      MOV #77406,@#UIPDR5      ; I SPACE
13458 056712 000403      BR 11$
13460 056714 012737 077406 177632 10$: MOV #77406,@#UDPDR5      ;D-SPACE IS ENABLED; ALLOW R/W ACCESS
13465          ; TO TEST BUFFER AREA IN USER D-SPACE
13466 056722 005737 001760 11$:  TST RANDOM              ;ASSURE THAT XLATION BUFFER (MOVTC) IS IN READABLE SPACE
13467          ;IN RANDOM MODE?
13468 056726 001454      BEQ 12$                 ;BRANCH IF NO
13469 056730 022737 076032 050012      CMP #76032,TINST        ;IS INST UNDER TEST = MOVTC?
13470 056736 001404      BEQ 20$                 ;BRANCH IF YES
13471 056740 022737 076132 050012      CMP #76132,TINST
13472 056746 001044      BNE 12$
13473 056750 005737 002162 20$:  TST DEN                  ;IS D-SPACE ENABLED?
13474 056754 001017      BNE 13$                 ;BRANCH IF YES
13475 056756 005737 002160      TST MODE                ;IS MODE = KERNEL?

```

```

13476 056762 001436          BEQ 12$          ;BRANCH IF YES - NO PDR CHANGE REQUIRED FOR XLATION BUFF
13477 056764 022737 000001 002160    CMP #1,MODE     ;IS MODE = SUPERVISOR?
13478 056772 001004          BNE 14$          ;BRANCH IF NO
13479 056774 012737 077402 172202    MOV #77402,@#SIPDR1 ;ALLOW R-ONLY ACCESS TO XLATION BUFFER
13480 057002 000426          BR 12$
13481 057004 012737 077402 177602 14$:  MOV #77402,@#UIPDR1 ;USER MODE- ALLOW R-ONLY ACCESS TO XLATION BUFFER
13482 057012 000422          BR 12$
13483 057014 005737 002160          13$:  TST MODE
13484 057020 001004          BNE 15$
13485 057022 012737 077402 172322    MOV #77402,@#KDPDR1 ;KERNEL MODE,D-EN - ALLOW R-ONLY ACCESS TO XLATI
13486 057030 000413          BR 12$
13487 057032 022737 000001 002160 15$:  CMP #1,MODE
13488 057040 001004          BNE 16$
13489 057042 012737 077402 172222    MOV #77402,@#SDPDR1 ;SUPERVISOR MODE,D-EN - ALLOW READ OF XLATION BU
13490 057050 000403          BR 12$
13491 057052 012737 077402 177622 16$:  MOV #77402,@#UDPDR1 ;USER MODE,D-EN - ALLOW R-ONLY ACCESS OF XLATION
13492 057060 000207          12$:  RTS PC
13493
13494
13495
13496
13497          ;HALT SERVICE ROUTINE (USED FOR TRAPS CAUSED BY HALTS IN SUPERVISOR OR USER MODE
13498
13499 057062 005037 177572          HLTSER: CLR @#MMRO ;TURN OFF MEM MGMT - OVERWRITTEN WITH NOPS
13500
13501 057066 021627 050032          CMP (SP),#TINRET+2 ; IF NO MEM MGMT ON SYSTEM (REF SIZEPT ROUTINE)
13502 057072 001434          BEQ 1$          ;WAS HALT AT CIS INST RETURN LOC?
13503 057074 011637 002166          MOV (SP),TRPLOC ;BRANCH IF YES
13504 057100 162737 000002 002166    SUB #2,TRPLOC   ;GET TRAP LOCATION
13505 057106          PRINTB #TRAP4 ;PRINT - TRAP TO LOC 4
(6) 057106 012746 012132          MOV #TRAP4,-(SP)
(3) 057112 010600          MOV SP,RO
(4) 057114 004737 065410          JSR PC,FPRINT
13506 057120          PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTCH,TOTTC ;PRINT - ADDITIONAL TRAP
(12) 057120 013746 001420          MOV TOTTC,-(SP)
(11) 057124 013746 001416          MOV TOTTCH,-(SP)
(10) 057130 013746 050012          MOV TINST,-(SP)
(9) 057134 013746 002162          MOV DEN,-(SP)
(8) 057140 013746 002160          MOV MODE,-(SP)
(7) 057144 013746 002166          MOV TRPLOC,-(SP)
(6) 057150 012746 012304          MOV #TRPINF,-(SP)
(3) 057154 010600          MOV SP,RO
(4) 057156 004737 065410          JSR PC,FPRINT
13507 057162 000000          HALT
13508 057164 016637 000002 003666 1$:  MOV 2(SP),TCCR ;SAVE CIS INST RETURN CONDITION CODES
13509 057172 005726          TST (SP)+
13510 057174 005726          TST (SP)+
13511 057176 000137 050054          JMP SUHRET
13512
13513
13514          ;ILLEGAL INSTRUCTION TRAP SERVICE ROUTINE
13515
13516
13517 057202 005037 177572          ILLSER: CLR @#MMRO ;TURN OFF MEM MGMT - OVERWRITTEN WITH NOPS

```

```

13518                                     ; IF NO MEM MGMT ON SYSTEM (REF SIZEPT ROUTINE)
13519 057206 005737 002154             TST PT34                    ;11/34 TYPE PROCESSOR?
13520 057212 001403                   BEQ 2$                      ;BRANCH IF NO
13521                                     ;NOTE:ON 11/34 HALT IN USER MODE TRAPS TO 10 (NOT 4)
13522 057214 021627 050032             CMP (SP),#TINRET+2         ;WAS HALT AT CIS INST RETURN LOC?
13523 057220 001452                   BEQ 3$                      ;BRANCH IF YES
13524 057222 022737 000001 001420 2$:  CMP #1,TOTTC                ;WAS ILLEGAL INST TRAP DURING TEST #1 ?
13525 057230 001011                   BNE 1$                      ;BRANCH IF NO
13526 057232 021627 050014             CMP (SP),#TINST+2        ;WAS ILLEGAL INST THE CIS INST?
13527 057236 001006                   BNE 1$                      ;BRANCH IF NO
13528 057240                           PRINTB #CISQ               ;PRINT CISP PRESENT? WARNING
      (6) 057240 012746 012007         MOV #CISQ,-(SP)
      (3) 057244 010600                 MOV SP,R0
      (4) 057246 004737 065410         JSR PC,FPRINT
13529 057252 000000                   HALT
13530 057254 011637 002166             MOV (SP),TRPLOC           ;GET LOCATION THAT CAUSED TRAP
13531 057260 162737 000002 002166 1$:  SUB #2,TRPLOC
13532 057266                           PRINTB #TRAP10             ;PRINT TRAP TO 10
      (6) 057266 012746 012150         MOV #TRAP10,-(SP)
      (3) 057272 010600                 MOV SP,R0
      (4) 057274 004737 065410         JSR PC,FPRINT
13533 057300                           PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTC ;PRINT ADDITIONAL TRAP I
      (12) 057300 013746 001420        MOV TOTTC,-(SP)
      (11) 057304 013746 001416        MOV TOTTC,-(SP)
      (10) 057310 013746 050012        MOV TINST,-(SP)
      (9) 057314 013746 002162         MOV DEN,-(SP)
      (8) 057320 013746 002160         MOV MODE,-(SP)
      (7) 057324 013746 002166         MOV TRPLOC,-(SP)
      (6) 057330 012746 012304         MOV #TRPINF,-(SP)
      (3) 057334 010600                 MOV SP,R0
      (4) 057336 004737 065410         JSR PC,FPRINT
13534 057342 000000                   HALT
13535 057344 000002                   RTI
13536 057346 016637 000002 003666 3$:  MOV 2(SP),TCCR            ;SAVE CIS INST RETURN CONDITON CODES
13537 057354 005726                   TST (SP)+
13538 057356 005726                   TST (SP)+
13539 057360 000137 050054             JMP SUHRET
13540
13541
13542                                     ;
13543                                     ;MEMORY MANAGEMENT VIOLATION TRAP SERVICE ROUTINE
13544                                     ;
MMVIOL: CLR @#MMRO                    ;TURN OFF MEM MGMT
13545 057364 005037 177572             MOV (SP),TRPLOC           ;GET LOCATION WHICH CAUSED TRAP
13546 057370 011637 002166             SUB #2,TRPLOC
13547 057374 162737 000002 002166     PRINTB #MMVMSG           ;PRINT MEMORY MANAGEMENT VIOL
13548 057402                           MOV #MMVMSG,-(SP)
      (6) 057402 012746 012112         MOV SP,R0
      (3) 057406 010600                 MOV SP,R0
      (4) 057410 004737 065410         JSR PC,FPRINT
13549 057414                           PRINTB #TRPINF,TRPLOC,MODE,DEN,TINST,TOTTC
      (12) 057414 013746 001420        MOV TOTTC,-(SP)
      (11) 057420 013746 001416        MOV TOTTC,-(SP)
      (10) 057424 013746 050012        MOV TINST,-(SP)
      (9) 057430 013746 002162         MOV DEN,-(SP)

```

```

(8) 057434 013746 002160
(7) 057440 013746 002166
(6) 057444 012746 012304
(3) 057450 010600
(4) 057452 004737 065410
13550 057456 000000
13551 057460 000002
13552
13553
13554
13555
13556 057462
13557 057462 105777 122200
13558 057466 100375
13559 057470 117737 122216 064776
13560 057476 042737 177600 064776
13561 057504 004737 064760
13562 057510 123727 064776 000113
13563 057516 001411
13564 057520 123727 064776 000123
13565 057526 001410
13566 057530 123727 064776 000125
13567 057536 001410
13568 057540 000414
13569 057542 005037 002160
13570 057546 000407
13571 057550 012737 000001 002160
13572 057556 000403
13573 057560 012737 000003 002160
13574 057566 062716 000002
13575 057572 000207
13576
13577

```

```

MOV MODE,-(SP)
MOV TRPLOC,-(SP)
MOV #TRPINF,-(SP)
MOV SP,RO
JSR PC,FPRINT
HALT
RTI

```

ACCEPT ASCII (K,S, OR U) FROM TTY AND SETUP MODE WORD

KSORU:

```

1$: TSTB @TKS ;WAIT FOR A CHARACTER
BPL 1$
MOVB @TKB,RCHAR ;READ & SAVE CHAR
BIC #^C177,RCHAR ;GET RID OF JUNK IF ANY
JSR PC,ECHAR ;ECHO CHARACTER
CMPB RCHAR,#113 ;IS CHAR A 'K'
BEQ 6$ ;BRANCH IF YES
CMPB RCHAR,#123 ;IS CHAR AN 'S'
BEQ 3$ ;BRANCH IF YES
CMPB RCHAR,#125 ;IS CHAR A 'U' ?
BEQ 4$ ;BRANCH IF YES
BR 5$ ;CHAR IS ILLEGAL :RETURN TO CALL +2
6$: CLR MODE ;SET MODE TO KERNEL (0)
BR 2$
3$: MOV #1,MODE ;SET MODE WORD TO SUPERVISOR (1)
BR 2$
4$: MOV #3,MODE ;SET MODE WORD TO USER (3)
2$: ADD #2,(SP)
5$: RTS PC

```

```

13579          .SBTTL          PROGRAMMABLE CLOCK SERVICE ROUTINE
13581          ;NOTE: LOCATIONS 57670-60070 ARE RESERVED FOR STACK USAGE DURING
13582          ;          EXECUTION OF CIS STACK PROBEAHEAD MEMORY MGMT ABORT TESTS.
13583
13584          060100          . =60100
13586
13587          ; KW11-P (PROGRAMMABLE CLOCK) INTERRUPT SERVICE ROUTINES (ALWAYS ENTERED IN KERNEL MODE)
13588
13589          060100          PCIS3:          ;P CLK SERVICE ROUTINE USED FOR INTR DURING
13590          ;          EXECUTION OF 'STATE DISTURBING' DIVPI
13591          ;          INSTRUCTION BELOW.
13592          060100 042777 000001 122412          BIC #1,@PC1CSR          ;TURN OFF PCLK1
13593          060106 000002          RTI
13594
13595          060110 005037 177572          PCIS2: CLR @#MMRO          ;P CLK INTR SERVICE ROUTINE USED WHEN
13596          ;          TESTING LATENCY
13597          ;THIS CLR INSTRUCTION TURNS OFF MEMORY MANAGEMENT
13598          ; THE CLR IS OVERWRITTEN WITH NOPS IF NOT 11/44
13599          ; (SEE SIZEPT ROUTINE).
13600          060114 017737 122414 002542          MOV @PC2CTR,LATCT          ;SAVE P-CLK2 COUNTER FOR LATENCY CALCULATION
13601          060122 042777 000001 122400          BIC #1,@PC2CSR          ;TURN OFF PCLK2
13602          060130 005077 122376          CLR @PC2CSB          ;CLEAR PCLK2 COUNTER
13603
13604          060134 005037 177572          PCIS1: CLR @#MMRO          ;NORMAL P CLK INTERRUPT SERVICE ROUTINE
13605          ;THIS CLR TURNS OFF MEM MGMT.
13606          ;THIS CLR GETS OVERWRITTEN WITH NOPS IF NOT 11/44
13607          060140 042777 000001 122352          BIC #1,@PC1CSR          ;TURN OFF P-CLK1
13608          060146 004737 061206          JSR PC,SGPRO6          ;SAVE GENERAL PURPOSE REGS 0-6
13609          060152 021627 050012          CMP (SP),#TINST          ;INTERRUPTED THE TEST INST???
13610          060156 001154          BNE 1$          ;BRANCH IF NO
13611          060160 032766 000400 000002          BIT #400,2(SP)          ;IS PSW BIT 8 SET?
13612          060166 001407          BEQ 4$          ;BRANCH IF NO
13613          060170 005237 002546          INC INTCT          ;UPDATE INTERRUPT COUNT
13614          060174 052737 040000 002132          BIS #40000,FATAL          ;SET INTERRUPT INDICATOR IN FATAL ERROR WORD
13615          060202 004737 061040          JSR PC,RECLAT          ;RECORD LATENCY
13616          060206 004737 051434          4$: JSR PC,STATCG          ;HAS THE 'STATE' OF CIS INST CHANGED?
13617          060212 000504          BR 5$          ;NO RETURN
13618          060214 005237 002540          INC PROGCT          ;YES RETURN - UPDATE PROGRESS COUNT
13619          060220 032766 000400 000002          BIT #400,2(SP)          ;IS PSW BIT 8 SET?
13620          060226 001025          BNE 6$          ;BRANCH IF YES
13621          060230          PRINTB #HLTMSG
13622          (6) 060230 012746 012167          MOV #HLTMSG,-(SP)
13623          (3) 060234 010600          MOV SP,R0
13624          (4) 060236 004737 065410          JSR PC,FPRINT
13625          060242 012737 060300 002170          MOV #100$,HLTLOC
13626          060250 004737 055402          JSR PC,IDINFO          ;IDENTIFY FAILING INST
13627          060254          PRINTB #FORM42          ;MSG: CIS INST WAS SUSPENDED TO SERVICE INTR
13628          (6) 060254 012746 014745          MOV #FORM42,-(SP)
13629          (3) 060260 010600          MOV SP,R0
13630          (4) 060262 004737 065410          JSR PC,FPRINT
13631          060266          PRINTB #FORM43          ;MSG: PSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT
13632          (6) 060266 012746 015034          MOV #FORM43,-(SP)
13633          (3) 060272 010600          MOV SP,R0
13634          (4) 060274 004737 065410          JSR PC,FPRINT

```



```

13626 060300 000000          100$: HALT                ;BIT 8 OF PSW SHOULD HAVE BEEN SET!!!
13627                                     ;PRESS CONTINUE TO PROCEED WITH TESTING
13628
13629 060302          6$:                                     ;INST IS IN A NEW PART OF ITS OPERATION
13630 060302 012737 000001 002556  MOV #1,INTRVL          ;SET P-CLK INTERVAL TO MINIMUM = 1
13631 060310 004737 061316      JSR PC,SAVST          ;SAVE CIS INST STATE
13632 060314 004737 061152          11$: JSR PC,DIC           ;ALLOW INTERRUPT DURING THIS DIVPI IF REQUESTED
13633 060320 076175      DIVPI          ;DISTURB INTERNAL CISP STATE BY
13634 060322 003152      DIVDS          ; EXECUTING A DIVP IN-LINE INST.
13635 060324 003152      DIVDS
13636 060326 003156      DIVDD
13637 060330 042777 000001 122162  BIC #1,@PC1CSR        ;TURN OFF PCLK1
13638 060336 013777 002036 122150  MOV SPCV,@PCLK1V      ;RESTORE PCLK VECTOR
13639 060344 013737 002040 120400  MOV TPRECS,PRECSK     ;RESTORE 65TH STACK WORD
13640 060352 004737 061264      JSR PC,RGPRO6         ;RESTORE GENERAL PURPOSE REGISTERS 0-6
13641 060356 013777 002556 122136  MOV INTRVL,@PC1CSB    ;SET INTERVAL
13642 060364 005737 002544      TST LATEN           ;LATENCY TESTING?
13643 060370 001403      BEQ 7$              ;BRANCH IF NO
13644 060372 052777 000001 122130  BIS #1,@PC2CSR        ;TURN ON P-CLK2
13645 060400 005737 002156          7$: TST MMFLG          ;TESTING WITH MEM MGMT ON?
13646 060404 001403      BEQ 8$              ;BRANCH IF NO
13647 060406 052737 000001 177572  BIS #1,@MMRO         ;TURN ON MEM MGMT
13648 060414 052777 000001 122076  8$: BIS #1,@PC1CSR    ;TURN ON P-CLK1
13649 060422 000002          14$: RTI                ;RETURN FROM SERVICE
13650
13651 060424 005237 002556          5$: INC INTRVL          ;INCREASE P-CLK1 INTERVAL
13652 060430 023737 002556 002562  CMP INTRVL,MAXIVL     ;IS INTERVAL GREATER THAN SOME PRESET
13653                                     ; MAXIMUM ALLOWED?
13654 060436 103420          BLO 12$            ;BRANCH IF NO
13655 060440          PRINTB #HLTMSG
    (6) 060440 012746 012167      MOV #HLTMSG,-(SP)
    (3) 060444 010600      MOV SP,R0
    (4) 060446 004737 065410      JSR PC,FPRINT
13656 060452 012737 060476 002170  MOV #101$,HLTLOC
13657 060460 004737 055402      JSR PC,IDINFO        ;IDENTIFY FAILING INST
13658 060464          PRINTB #NOPROG
    (6) 060464 012746 012206      MOV #NOPROG,-(SP)
    (3) 060470 010600      MOV SP,R0
    (4) 060472 004737 065410      JSR PC,FPRINT
13659 060476 000000          101$: HALT                ;CIS INST DID NOT MAKE PROGRESS -
13660                                     ; INTERVAL TILL INTERRUPT EXCEEDS USER
13661                                     ; DEFINED MAXIMUM ALLOWED.
13662 060500 005737 002540          12$: TST PROGCT        ;HAS PROGRESS BEEN MADE ON THIS INST PREVIOUSLY?
13663 060504 001445      BEQ 13$            ;BRANCH IF NO
13664 060506 000702          BR 11$              ;DISTURB INTERNAL STATE; THEN RETURN
13665                                     ; FOR MORE PROGRESS
13666 060510 101054          1$: BHI 15$            ;BRANCH IF INST ALREADY HAS COMPLETED (PC>TINST)
13667 060512 032766 000400 000002  BIT #400,2(SP)       ;DID NOT REACH INST YET.
13668                                     ;VERIFY THAT BIT 8 OF PSW IS NOT SET.
13669 060520 001435          BEQ 17$            ;BRANCH IF BIT 8 = 0.
13670 060522          PRINTB #HLTMSG
    (6) 060522 012746 012167      MOV #HLTMSG,-(SP)
    (3) 060526 010600      MOV SP,R0
    (4) 060530 004737 065410      JSR PC,FPRINT
  
```

13671	060534	012737	000102	002170		MOV #102,HLTLOC	
13672	060542	004737	055402			JSR PC,IDINFO	;IDENTIFY FAILING INST
13673	060546					PRINTB #FORM44	;MSG: BIT 8 OF PSW SET WITH PC < CIS INST PC
(6)	060546	012746	015115			MOV #FORM44,-(SP)	
(3)	060552	010600				MOV SP,R0	
(4)	060554	004737	065410			JSR PC,FPRINT	
13674	060560					PRINTB #FORM45	;MSG: SUSPECT THAT CIS INST BACKED UP TOO FAR
(6)	060560	012746	015170			MOV #FORM45,-(SP)	
(3)	060564	010600				MOV SP,R0	
(4)	060566	004737	065410			JSR PC,FPRINT	
13675	060572					PRINTB #FORM46	;MSG: WHEN SERVICING INTERRUPT
(6)	060572	012746	015247			MOV #FORM46,-(SP)	
(3)	060576	010600				MOV SP,R0	
(4)	060600	004737	065410			JSR PC,FPRINT	
13676	060604	000000			102\$:	HALT	;BIT 8 OF PSW SET WITH PC < CIS INST PC.
13677	060606	042766	000400	000002		BIC #400,2(SP)	;SUSPECT THAT CIS INST BACKED UP PC TOO FAR
13678							; WHEN SERVICING INTERRUPT.
13679							;PRESS CONTINUE TO PROCEED WITH TESTING
13680	060614	005237	002556		17\$:	INC INTRVL	
13681	060620	013777	002556	121674	13\$:	MOV INTRVL,@PC1CSB	;INCREASE INTERVAL
13682	060626	004737	061264			JSR PC,RGPR06	;RESTORE REGISTERS
13683	060632	005726				TST (SP)+	;FIX UP STACK POINTER
13684	060634	005726				TST (SP)+	
13685	060636	000137	047752			JMP TOPC2	;RETURN TO TURN ON PCLK-1 POINT
13686							
13687	060642	004737	061040		15\$:	JSR PC,RECLAT	;RECORD LATENCY
13688	060646	032737	000100	050012		BIT #100,TINST	;IS INST UNDER TEST AN IN-LINE INST
13689	060654	001430				BEQ 20\$	;BRANCH IF NO
13690	060656	021637	002214			CMP (SP),ICOMPC	;VERIFY THAT PC HAS BEEN ADJUSTED TO POINT
13691							; TO NEXT INST
13692	060662	103025				BHIS 20\$	;BRANCH IF PC IS OK
13693	060664					PRINTB #HLTMSG	
(6)	060664	012746	012167			MOV #HLTMSG,-(SP)	
(3)	060670	010600				MOV SP,R0	
(4)	060672	004737	065410			JSR PC,FPRINT	
13694	060676	012737	060734	002170		MOV #103\$,HLTLOC	
13695	060704	004737	055402			JSR PC,IDINFO	;IDENTIFY FAILING INST
13696	060710					PRINTB #FORM48	;MSG: IN-LINE CIS INST COMPLETED WITH PC
(6)	060710	012746	015401			MOV #FORM48,-(SP)	
(3)	060714	010600				MOV SP,R0	
(4)	060716	004737	065410			JSR PC,FPRINT	
13697	060722					PRINTB #FORM49	;MSG: POINTING AT IN-LINE OPERANDS RATHER
(6)	060722	012746	015457			MOV #FORM49,-(SP)	
(3)	060726	010600				MOV SP,R0	
(4)	060730	004737	065410			JSR PC,FPRINT	
13698							; THAN NEXT INST.
13699	060734	000000			103\$:	HALT	;PRESS CONTINUE TO PROCEED WITH TESTING
13700	060736	004737	061264		20\$:	JSR PC,RGPR06	;RESTORE REGISTERS
13701	060742	032766	000400	000002		BIT #400,2(SP)	;CIS INST COMPLETE - VERIFY THAT PSW BIT 8
13702							; IS NOT SET
13703	060750	001420				BEQ 16\$	
13704	060752					PRINTB #HLTMSG	
(6)	060752	012746	012167			MOV #HLTMSG,-(SP)	
(3)	060756	010600				MOV SP,R0	

```

(4) 060760 004737 065410 JSR PC,FPRINT
13705 060764 012737 061010 002170 MOV #104$,HLTLOC
13706 060772 004737 055402 JSR PC,IDINFO ;IDENTIFY FAILING INST
13707 060776 PRINTB #FORM47 ;MSG: CIS INST COMPLETED BUT PSW BIT 8 STILL SET
(6) 060776 012746 015320 MOV #FORM47,-(SP)
(3) 061002 010600 MOV SP,R0
(4) 061004 004737 065410 JSR PC,FPRINT
13708 061010 000000 104$: HALT ;CIS INST COMPLETED BUT PSW BIT 8 STILL SET
13709 ;PRESS CONTINUE TO PROCEED WITH TESTING
13710 061012 042766 000400 000002 16$: BIC #400,2(SP)
13711 061020 012777 000001 121474 MOV #1,@PC1CSB ;SET UP FOR NEXT PASS
13712 061026 012737 000001 002556 MOV #1,INTRVL
13713 061034 000137 060422 JMP 14$
13714
13715 ;KW11-P INTERRUPT SERVICE ROUTINE SUBROUTINES
13716 ;
13717 ;ROUTINE TO RECORD INTERRUPT LATENCY
13718 ;
13719 061040 163737 002556 002542 RECLAT: SUB INTRVL,LATCT ;CALCULATE LATENCY
13720 061046 005737 002560 TST STOPLA ;STOP ON EXCESSIVE LATENCY?
13721 061052 001424 BEQ 2$ ;BRANCH IF NO
13722 061054 023737 002560 002542 CMP STOPLA,LATCT ;IS LATENCY EXCESSIVE?
13723 061062 101020 BHI 2$ ;BRANCH IF NO
13724 061064 PRINTB #HLTMSG
(6) 061064 012746 012167 MOV #HLTMSG,-(SP)
(3) 061070 010600 MOV SP,R0
(4) 061072 004737 065410 JSR PC,FPRINT
13725 061076 012737 000100 002170 MOV #100,HLTLOC
13726 061104 004737 055402 JSR PC,IDINFO ;IDENTIFY FAILING INST
13727 061110 PRINTB #LATEXC
(6) 061110 012746 012254 MOV #LATEXC,-(SP)
(3) 061114 010600 MOV SP,R0
(4) 061116 004737 065410 JSR PC,FPRINT
13728 061122 000000 100$: HALT ;LATENCY EXCEEDED USER DEFINED
13729 ; 'MAXIMUM ALLOWABLE'
13730
13731 061124 013701 002276 2$: MOV OCTIC,R1 ;RECORD LATENCY
13732 061130 006301 ASL R1
13733 061132 062701 004106 ADD #ILATEN,R1 ;FORM POINTER INTO INST LATENCY TABLE
13734 061136 021137 002542 CMP (R1),LATCT ;IS LATENCY BIGGER THAN THAT ALREADY
13735 ; RECORDED FOR INST?
13736 061142 101002 BHI 1$ ;BRANCH IF NO
13737 061144 013711 002542 MOV LATCT,(R1) ;SAVE NEW LATENCY VALUE
13738 061150 000207 1$: RTS PC
13739
13740 ;ROUTINE TO TURN ON P-CLK DURING DIVPI 'STATE DISTURBING' INST
13741 ;
13742 061152 017737 121336 002036 DIC: MOV @PCLK1V,SPCV ;SAVE CONTENTS OF P CLK INTERR VECTOR
13743 061160 013737 120400 002040 MOV PRECSK,IPRECS ;SAVE CONTENTS OF 65TH STACK WORD
13744 061166 012777 060100 121320 MOV #PCIS3,@PCLK1V ;SFTUP INTR VECTOR
13745 061174 000207 DI: RTS PC ;OVERWRITTEN WITH A NOP IF USER REQUESTS
13746 061176 052777 000001 121314 BIS #1,@PC1CSR ;TURN ON P-CLK1 - ENABLE INTR DURING
13747 ; SUBSEQUENT (STATE DISTURBING) CIS INST
13748 061204 000207 RTS PC

```

```

13749
13750      ;ROUTINE TO SAVE GENERAL PURPOSE REGISTERS 0 THROUGH 6.
13751      ;
13752      ;SGPR06:
13753      061206 010037 003004      MOV R0,SGPRO
13754      061212 010137 003006      MOV R1,SGPR1
13755      061216 010237 003010      MOV R2,SGPR2
13756      061222 010337 003012      MOV R3,SGPR3
13757      061226 010437 003014      MOV R4,SGPR4
13758      061232 010537 003016      MOV R5,SGPR5
13759      061236 032737 030000 177776 BIT #30000,PSW      ;WAS PREVIOUS MODE USER OR SUPV?
13760      061244 001404      BEQ 1$      ;BRANCH IF NO
13761      061246 006506      MFPI SP      ;GET PREVIOUS MODE SP
13762      061250 012637 003020      MOV (SP)+,SGPR6      ;STORE PREVIOUS MODE SP IN SGPR6
13763      061254 000402      BR 2$
13764      061256 010637 003020 1$:      MOV SP,SGPR6
13765      061262 000207 2$:      RTS PC
13766
13767
13768      ;ROUTINE TO RESTORE GENERAL PURPOSE REGISTERS 0 THROUGH 5
13769      ;
13770      ;RGPR06:
13771      061264 013700 003004      MOV SGPRO,R0
13772      061270 013701 003006      MOV SGPR1,R1
13773      061274 013702 003010      MOV SGPR2,R2
13774      061300 013703 003012      MOV SGPR3,R3
13775      061304 013704 003014      MOV SGPR4,R4
13776      061310 013705 003016      MOV SGPR5,R5
13777
13778      061314 000207      RTS PC      ;NOTE NO NEED TO RESTORE R6 BECAUSE IT HAS NOT CHANGED.
13779
13780
13781      ;ROUTINE TO SAVE STATE OF CISP - STATE=STACK POINTER,GENERAL PURPOSE REGISTER
  
```

```

13783 ; CONTENTS & STACK CONTENTS.
13784 ;
13785 061316 SAVST:
13786 061316 016637 000004 002564 MOV 4(SP),STATPS ;SAVE PSW STATE
13787 061324 013737 003004 002566 MCV SGPRO,STATRO ;SAVE STATE OF GENERAL PURPOSE REGS 0-6
13788 061332 013737 003006 002570 MOV SGPR1,STATR1
13789 061340 013737 003010 002572 MOV SGPR2,STATR2
13790 061346 013737 003012 002574 MOV SGPR3,STATR3
13791 061354 013737 003014 002576 MOV SGPR4,STATP4
13792 061362 013737 003016 002600 MOV SGPR5,STATR5

```

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
PROGRAMMABLE CLOCK SERVICE ROUTINE

M 16  
PAGE 93 SEQUENCE 207

13794 061370 013737 003020 002602  
13795 061376 013700 003020

MOV SGPR6,STATR6  
MOV SGPR6,R0

;SAVE STACK CONTENTS

PDP-11 CIS INST EXERCISER      MACY11 27(655) 29-SEP-80 09:22      N 16  
CZKEEB.P11      PROGRAMMABLE CLOCK SERVICE ROUTINE      PAGE 94 SEQUENCE 208

13797 061402 062700 000006

ADD #6,R0

13799 061406 020027 120602  
13800 061412 103007  
13801 061414 012702 003004  
13802 061420 012701 120602  
13803 061424 014142  
13804 061426 020100  
13805 061430 103375  
13806 061432 000207  
13807  
13808  
13809  
13810

          CMP R0,#CSTACK          ;DID CIS INST PUSH ANYTHING ONTO STACK?  
          BHS 1\$                  ;BRANCH IF NO  
          MOV #SCSTK,R2          ;COPY USED PORTION OF STACK INTO A  
          MOV #CSTACK,R1          ;  SAVE AREA.  
2\$:      MOV -(R1),-(R2)  
          CMP R1,R0              ;ALL OF USED PORTION OF STACK COPIED?  
          BHS 2\$                  ;BRANCH IF NO  
1\$:      RTS PC  
:  
:ROUTINE TO CHECK FOR A CISP STATE CHANGE  
:  RETURNS TO CALL ON NO CHANGE; CALL+2 ON CHANGE



PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
PROGRAMMABLE CLOCK SERVICE ROUTINE

C 1  
PAGE 96 SEQUENCE 210

13812

13813 061434

13814 061434 026637 000004 002564

STATCG:

(MP 4(SP),STATPS

;DID PSW CHANGE?

13816	061442	001054			BNE STHASC	;BRANCH IF YES
13817	061444	023737	003020	002602	CMP SGPR6,STATR6	;DID STACK POINTER CHANGE?
13818	061452	001050			BNE STHASC	;BRANCH IF YES
13819	061454	023737	003004	002566	CMP SGPRO,STATRO	;DID ANY OF THE GENERAL PURPOSE REGISTER
13820						;CONTENTS CHANGE?
13821	061462	001044			BNE STHASC	;BRANCH IF RO CHANGED

```

13823 061464 023737 003006 002570      CMP SGPR1,STATR1
13824 061472 001040                      BNE STHASC          ;BRANCH IF R1 HAS CHANGED
13825 061474 023737 003010 002572      CMP SGPR2,STATR2
13826 061502 001034                      BNE STHASC          ;BRANCH IF R2 HAS CHANGED
13827 061504 023737 003012 002574      CMP SGPR3,STATR3
13828 061512 001030                      BNE STHASC          ;BRANCH IF R3 HAS CHANGED
13829 061514 023737 003014 002576      CMP SGPR4,STATR4
13830 061522 001024                      BNE STHASC          ;BRANCH IF R4 HAS CHANGED
13831 061524 023737 003016 002600      CMP SGPR5,STATR5
13832 061532 001020                      BNE STHASC          ;BRANCH IF R5 HAS CHANGED
13833 061534 013700 003020      MOV SGPR6,R0      ;DID THE STACK CONTENTS CHANGE
13834 061540 062700 000006      ADD #6,R0
13835 061544 020027 120602      CMP R0,#CSTACK   ;DID ANYTHING GET PUSHED ONTO THE STACK?
13836 061550 103010                      BHS 1$            ;BRANCH IF NO
13837 061552 012702 003004      MOV #SCSTK,R2
13838 061556 012701 120602      MOV #CSTACK,R1
13839 061562 024142          2$:  CMP -(R1),-(R2)   ;DID ANY OF THE INFORMATION ON THE STACK GET CHANGED?
13840 061564 001003                      BNE STHASC          ;BRANCH IF YES
13841 061566 020100                      CMP R1,R0          ;ALL OF STACK CHECKED?
13842 061570 103374                      BHS 2$            ;BRANCH IF NO
13843 061572 000402          1$:  BR NOSCHG
13844 061574 062716 000002      STHASC: ADD #2,(SP)
13845 061600 000207      NOSCHG: RTS PC
13846
13847
13848 061602      ;ROUTINE TO CHECK FOR AND SETUP P-CLK 1
      PC1CK:          ;TEST FOR P-CLKS PRESENT

```

```

13850 061602 013701 002536      MOV TIMEOUT,R1      ;SAVE TIME OUT VECTOR CONTENTS
13851 061606 011146      MOV (R1),-(SP)
13852 061610 012721 061676      MOV #1$, (R1)+
13853 061614 011146      MOV (R1),-(SP)
13854 061616 005011      CLR (R1)
13855
13856 061620 005777 120674      TST @PC1CSR        ;ATTEMPT ACCESS P-CLK1 CSR
13857                                ;PCLK1 IS RESPONDING
13858 061624 012777 060134 120662  MOV #PCIS1,@PCLK1V ;SET UP P-CLK1 INTERRUPT SERVICE VECTOR
13859 061632 005077 120662      CLR @PC1CSR        ;CLEAR P-CLK1 CSR
13860 061636 012777 000000 120654  MOV #000,@PC1CSR   ;SET P-CLK1 FOR SINGLE INT,COUNT DOWN,100K HZ
13861                                ; CLOCK, INT ENABLE
13862 061644 012777 000001 120650  MOV #1,@PC1CSB     ;SET COUNTER TO 1 (1 MICRO SEC INTERVAL
13863                                ; WITH 1 MHZ EXTERNAL CLOCK.
13864 061652 012737 000001 002556  MOV #1,INTRVL      ;SAVE INTERVAL SETTING
13865 061660 013737 002550 047762  MOV KNOPI,TOPCI    ;OVERWRITE BRANCH TO ALLOW TURNING ON OF
13866                                ; PCLK1 PRIOR TO CIS INST EXECUTION
13867 061666 062766 000004 000004  ADD #4,4(SP)       ;ADJUST STACK FOR 'EXISTS RETURN'
13868 061674 000402      BR 2$
13869 061676 005726      1$: TST (SP)+         ;FIX UP STACK POINTER
13870 061700 005726      TST (SP)+
13871 061702 013701 002536      2$: MOV TIMEOUT,R1   ;RESTORE TIME OUT INTR VECTOR TO ORIGINAL STATE
13872 061706 012661 000002      MOV (SP)+,2(R1)
13873 061712 012611      MOV (SP)+,(R1)
13874 061714 000207      RTS PC
13875
13881                                ;ROUTINE TO CHECK FOR AND SETUP P-CLK2
13882 061716 013701 002536  PC2CK: MOV TIMEOUT,R1 ;SAVE TIME OUT VECTOR CONTENTS
13883 061722 011146      MOV (R1),-(SP)
13884 061724 012721 062024      MOV #1$, (R1)+
13885 061730 011146      MOV (R1),-(SP)
13886 061732 005011      CLR (R1)
13887 061734 005777 120570      TST @PC2CSR        ;ATTEMPT ACCESS OF PCLK 2 CSR
13888 061740 012737 177777 002544  MOV #177777,LATEN  ;SET LATENCY TESTING FLAG
13889 061746 012700 004106      MOV #ILATEN,RO     ;CLEAR INTERRUPT LATENCY TABLE
13890 061752 005020      12$: CLR (RO)+
13891 061754 020027 004176      CMP RO,#LATEND+2
13892 061760 001374      BNE 12$
13893 061762 012777 060110 120524  MOV #PCIS2,@PCLK1V
13894 061770 005077 105534      CLR @PC2CSR        ;CLEAR P-CLK2 CSR

```

13896	061774	012777	000020	120526		MOV #20,@PC2CSR		;SET P-CLK2 FOR INT. DISABLE, COUNT UP
13897								; 100KHZ CLOCK
13898	062002	005077	120524			CLR @PC2CSB		;SET COUNTER TO 0
13899	062006	013737	002552	047752		MOV KNOP2, TOPC2		;OVERWRITE BRANCH TO ALLOW TURNING ON OF
13900								; PCLK2 PRIOR TO CIS INST EXECUTION
13901	062014	062766	000004	000004		ADD #4,4(SP)		;ADJUST STACK FOR 'EXISTS RETURN'
13902	062022	000402				BR 2\$		
13903	062024	005726			1\$:	TST (SP)+		;FIX UP STACK POINTER
13904	062026	005726				TST (SP)+		
13905	062030	013701	002536		2\$:	MOV TIMEOUT,R1		;RESTORE TIME OUT INTERRUPT VECTOR TO
13906	062034	012661	000002			MOV (SP)+,2(R1)		; ORIGINAL STATE
13907	062040	012611				MOV (SP)+,(R1)		
13908	062042	000207				RTS PC		



LINE TIME CLOCK ROUTINES

13921	062072	021627	050012			CMR (SP),#TINST	; INTERRUPTED THE CIS INST UNDER TEST?
13922	062076	001053				BNE EXLTCS	; NO - EXIT LTC SERVICE
13923	062100	004737	061434			JSR PC,STATCG	; HAS THE STATE OF CIS INST CHANGED?
13924	062104	000450				BR EXLTCS	; NO RETURN - EXIT LTC SERVICE
13925	062106	032766	000400	000002		BIT #400,2(SP)	; YES RETURN - IS PSW BIT 8 SET?
13926	062114	001025				BNE 1\$	; BRANCH IF YES
13927	062116					PRINTB #HLTMSG	
(6)	062116	012746	012167			MOV #HLTMSG,-(SP)	
(3)	062122	010600				MOV SP,R0	
(4)	062124	004737	065410			JSR PC,FPRINT	
13928	062130	012737	062166	002170		MOV #100\$,HLTLOC	
13929	062136	004737	055402			JSR PC,IDINFO	; IDENTIFY FAILING INST
13930	062142					PRINTB #FORM42	; MSG: CIS INST WAS SUSPENDED TO SERVICE INTR
(6)	062142	012746	014745			MOV #FORM42,-(SP)	
(3)	062146	010600				MOV SP,R0	
(4)	062150	004737	065410			JSR PC,FPRINT	
13931	062154					PRINTB #FORM43	; MSG: PSW BIT 8 SHOULD HAVE BEEN SET BUT WAS NOT
(6)	062154	012746	015034			MOV #FORM43,-(SP)	
(3)	062160	010600				MOV SP,R0	
(4)	062162	004737	065410			JSR PC,FPRINT	
13932	062166	000000			100\$:	HALT	
13933							
13934	062170	005237	002546		1\$:	INC INTCT	; UPDATE INTERRUPT COUNT
13935	062174	052737	040000	002132		BIS #40000,FATAL	; SET INTR INDICATION IN FATAL ERROR WORD
13936	062202	013737	120400	002040		MOV PRECSK,TPRECS	; SAVE CONTENTS OF 65TH STACK WORD
13937	062210	076175				DIVPI	; DISTURB INTERNAL CISP STATE BY
13938	062212	003152				DIVDS	; EXECUTING A DIVP IN-LINE INST.
13939	062214	003152				DIVDS	
13940	062216	003156				DIVDD	
13941	062220	013737	002040	120400		MOV TPRECS,PRECSK	; RESTORE 65TH STACK WORD
13942	062226	004737	061264		EXLTCS:	JSR PC,RGPR06	; RESTORE GENERAL PURPOSE REGS 0-6
13943	062232	005737	002156			TST MMFLG	; TESTING WITH MEM MGMT ON ?
13944	062236	001403				BEQ 1\$	; BRANCH IF NO
13945	062240	013737	002210	177572		MOV MMSTAT,@#MMR0	; TURN ON MEM MGMT
13946	062246	000002			1\$:	RTI	; RETURN FROM SERVICE

```
13948
13949
13950           ;LTC SYNC UP ROUTINE
13951           ;
13952 062250           ;LTC SUP:
13953 062250 012777 062272 120544      MOV #LSUPR,@LTCIV      ;SETUP LTC INTERRUPT VECTOR
13954 062256 005077 120544             CLR @LKS
13955 062262 052777 000100 120536     BIS #100,@LKS        ;ENABLE INTERRUPTS
13956
13957 062270 000001           1$:      WAIT                ;WAIT FOR CLOCK SIGNAL
13958 062272 005077 120530           LSUPR:  CLR @LKS        ;DISABLE LTC INTERRUPTS
13959 062276 012777 062044 120516     MOV #LTCIS,@LTCIV    ;RESTORE LTC INTERRUPT VECTOR
13960 062304 005726             TST (SP)+           ;FIX UP STACK
13961 062306 005726             TST (SP)+
13962 062310 000207             RTS PC
13963
```



```
13965
13966           ;LTC - DETERMINE COUNT PER CLOCK TICK
13967           ;
13968           ;LTCINT:
13969 062312 012777 062340 120502      MOV #LTCINT,@LTCIV      ;SETUP LTC INTERRUPT SERVICE
13970 062312 005077 120502          CLR @LKS
13971 062324 052777 000100 120474    BIS #100,@LKS         ;ENABLE INTERRUPTS
13972 062332 005237 003030          1$: INC LCNT           ;COUNT TILL LTC INTERRUPTS
13973 062336 000775                  BR 1$
13974 062340 005077 120462          LTCINT: CLR @LKS      ;DISABLE LTC INTERRUPTS
13975 062344 163737 003036 003^30   SUB LTCDIY,LCNT       ;INTERRUPT RETURN
13976 062352 012777 062044 120442   MOV #LTCIS,@LTCIV    ;RESTORE LTC INTERRUPT VECTOR
13977 062360 005726                  TST (SP)+
13978 062362 005726                  TST (SP)+             ;FIX UP STACK
13979 062364 000207                  RTS PC
13980
13981
13982           ; LTC - ROUTINE TO CHECK FOR LINE TIME CLOCK ON SYSTEM
13983           ;
13984           ;LTCPC:
13985 062366 013701 002536          MOV TIMEOUT,R1        ;SAVE TIME OUT VECTOR
13986 062372 011146                  MOV (R1),-(SP)       ;SETUP INTERRUPT VECTOR
13987 062374 012721 062432          MOV #1$, (R1)+
13988 062400 011146                  MOV (R1),-(SP)
13989 062402 005011                  CLR (R1)
13990 062404 005777 120416          TST @LKS             ;ATTEMPT ACCESS OF LTC
13991 062410 005077 120412          CLR @LKS            ;CLEAR LTC CSR
```

13993	062414	013737	003034	047'20	MOV KNOP4,TOLTC	;OVERWRITE BRANCH TO ALLOW TURNING ON OF
13994						; OF LTC PRIOR TO CIS INST EXECUTION.
13995	062422	062766	000004	000004	ADD #4,4(SP)	;ADJUST RETURN TO CALL + 4
13996	062430	000402			BR 2\$	
13997	062432	005726			1\$: TST (SP)+	;FIX UP STACK POINTER
13998	062434	005726			TST (SP)+	
13999	062436	013701	002536		2\$: MOV TIMEOUT,R1	;RESTORE TIME OUT INTR VECTOR TO
14000	062442	012661	000002		MOV (SP)+,2(R1)	; ORIGINAL STATE
14001	062446	012611			MOV (SP)+,(R1)	
14002	062450	000207			RTS PC	
14003						
14004					;;RANDOM EXERCISE MODE SUBROUTINES	
14005						
14006						
14007	062452				SRNGST:	;SUBROUTINE TO SAVE RANDOM # GEN STATE
14008						;AT START OF EACH TEST
14009	062452	013737	063612	001776	MOV RNCON,STRNC	
14010	062460	013737	063614	002000	MOV RP1,STRP1	
14011	062466	013737	063616	002002	MOV RP2,STRP2	
14012	062474	000207			RTS PC	
14013						
14014	062476				SRNGSX:	;SUBROUTINE TO SAVE RANDOM # GEN. STATE X.
14015	062476	013737	063612	002004	MOV RNCON,SXRNC	
14016	062504	013737	063614	002006	MOV RP1,SXRP1	
14017	062512	013737	063616	002010	MOV RP2,SXRP2	
14018	062520	000207			RTS PC	
14019						
14020	062522				RRNGSX:	;SUBROUTINE TO RESTORE RANDOM # GEN STATE X.
14021	062522	013737	002004	063612	MOV SXRNC,RNCON	

14023	062530	013737	002006	063614
14024	062536	013737	002010	063616
14025	062544	000207		
14026				
14027	062546			
14028	062546	013737	063612	002012

MOV SGRP1,RP1  
MOV SGRP2,RP2  
RTS PC

SRNGSY:  
MOV RNCON,SYRNC

;SUBROUTINE TO SAVE RANDOM # GEN STATE Y.

14030	062554	013737	063614	002014		MOV RP1,SYRP1	
14031	062562	013737	063616	002016		MOV RP2,SYRP2	
14032	062570	000207				RTS PC	
14033							
14034	062572				RRNGSY:		;SUBROUTINE TO RESTORE RANDOM # GEN STATE Y.
14035	062572	013737	002012	063612		MOV SYRNC,RNCON	
14036	062600	013737	002014	063614		MOV SYRP1,RP1	
14037	062606	013737	002016	063616		MOV SYRP2,RP2	
14038	062614	000207				RTS PC	
14039							
14040	062616				SRNGSW:		;SUBROUTINE TO SAVE RANDOM # GEN STATE W.
14041	062616	013737	063612	002020		MOV RNCON,SWRNC	
14042	062624	013737	063614	002022		MOV RP1,SWRP1	
14043	062632	013737	063616	002024		MOV RP2,SWRP2	
14044	062640	000207				RTS PC	
14045							
14046	062642				RRNGSW:		;SUBROUTINE TO RESTORE RANDOM # GEN STATE W.
14047	062642	013737	002020	063612		MOV SWRNC,RNCON	
14048	062650	013737	002022	063614		MOV SWRP1,RP1	
14049	062656	013737	002024	063616		MOV SWRP2,RP2	
14050	062664	000207				RTS PC	
14051							
14052	062666				SRNGSV:		;SUBROUTINE TO SAVE RANDOM # GEN STATE V.
14053	062666	013737	063612	002026		MOV RNCON,SVRNC	
14054	062674	013737	063614	002030		MOV RP1,SVRP1	
14055	062702	013737	063616	002032		MOV RP2,SVRP2	
14056	062710	000207				RTS PC	
14057							
14058	062712				RRNGSV:		;SUBROUTINE TO RESTORE RANDOM # GEN STATE V.
14059	062712	013737	002026	063612		MOV SVRNC,RNCON	
14060	062720	013737	002030	063614		MOV SVRP1,RP1	

14062 062726 013737 002032 063616 MOV SVRP2,RP2

14064 062734 000207  
14065

RTS PC

```
14067      .SBTTL          RANDOM MODE SUBROUTINES
14068      :
14069      :SUBROUTINE TO GENERATE A RANDOM CIS INST AND LOAD ITS IDENTIFIER
14070      : (REFERENCE THE OINST TABLE) INTO THE FIRST WORD OF THE DUMMY
14071      : INPUT TABLE. THIS ROUTINE ONLY GENERATES CIS INSTS WHICH HAVE A NON-ZERO
14072      : ENCODING IN THE OINST TABLE. ZERO OINST TABLE ENTRIES ASSOC WITH CIS INST
14073      : NOT TO BE INCLUDED IN THE RANDOM EXERCISING.
14074      :
14075      :GENRI:
14076      :   JSR PC,RN          ;GET A RANDOM #
14077      :   BIC #BS128,RO     ;MASK OFF ALL BUT LEAST SIGNIF 5 BITS
14078      :   INC RO
14079      :   CMP RO,#31        ;VALID IDENTIFIERS = 1 TO 31
14080      :   BHI GENRI        ;BRANCH IF IDENTIFIER IS INVALID
```

```

14082 062756 010037 072144      MOV R0, IDUM      ;LOAD INST IDENTIFIER INTO DUMMY INPUT TABLE
14083 062762 006300              ASL R0
14084 062764 062700 003726      ADD #OINST, R0   ;VERIFY THAT GENERATED INST IS
14085 062770 005710              TST (R0)         ;A MEMBER OF THE SET OF CIS INSTS
14086 062772 001761              BEQ GENRI        ;TO BE RANDOMLY EXERCISED.
14087 062774 000207              RTS PC
14088
14089                          ;ROUTINE TO LOAD UP DUMMY INPUT TABLE USING RANDOM NUMBER GENERATOR.
14090                          ;ROUTINE USES THE RANDOM EXERCISE MASK TABLES TO LIMIT OPERANDS
14091                          ;(LENGTHS, ADDRESSES, ETC) TO THE PROPER RANGE.
14092
14093 062776 012702 072150      LDINPT: MOV #IDUM+4, R2      ;SETUP POINTER INTO DUMMY TABLE
14094 063002 013701 072144      MOV IDUM, R1
14095 063006 006301              ASL R1
14096 063010 062701 004176      ADD #MINST, R1
14097 063014 011101              MOV (R1), R1
14098 063016 012137 001764      1$: MOV (R1)+, PMASK        ;GET MASK FOR GIVEN INPUT PARAMETER
14099 063022 022737 125252 001764  CMP #EOT, PMASK   ;IS MASK=END OF MASK TABLE (EOT)
14100 063030 001431              BEQ IDFLD        ;BRANCH IF YES
14101 063032 022737 152525 001764  CMP #DSCPTR, PMASK ;DOES MASK INDICATE THAT INPUT
14102                          ;PARAMETER IS A DESCRIPTOR POINTER?
14103 063040 001003              BNE 2$          ;BRANCH IF NO
14104 063042 012722 002510      MOV #RANDSC, (R2)+ ;FILL TABLE ENTRY WITH A RANDOM
14105 063046 000763              BR 1$          ;DESCRIPTOR POINTER.
14106 063050 032737 100000 001764  2$: BIT #100000, PMASK ;MASK AND OFFSET?
14107 063056 001010              BNE 3$          ;BRANCH IF NO(MASK ONLY).
14108 063060 004737 063532      JSR PC, RN      ;GENERATE A RANDOM #
14110 063064 042700 176000      BIC #BS4, R0    ;MASK WITH 176000
14115 063070 063700 001764      ADD PMASK, R0   ;ADD IN OFFSET
14116 063074 010022              MOV R0, (R2)+   ;STORE INPUT PARAMETER
14117 063076 000747              BR 1$
14118 063100 004737 063532      3$: JSR PC, RN      ;GENERATE A RANDOM NUMBER
14119 063104 043700 001764      BIC PMASK, R0   ;MASK TO VALID RANGE
14120 063110 010022              MOV R0, (R2)+   ;STORE INPUT PARAMETER IN DUMMY TABLE
14121 063112 000741              BR 1$
14122 063114 005022              IDFLD: CLR (R2)+   ;CLEAR REMAINDER OF DUMMY INPUT TABLE
14123 063116 020227 072216      CMP R2, #IDUME
14124 063122 001374              BNE IDFLD
14125 063124 022737 000020 072144  CMP #20, IDUM    ;IS RANDOM MODE INST = ASHP OR ASHN?
14126 063132 001404              BEQ 1$          ;BRANCH IF YES
14127 063134 022737 000030 072144  CMP #30, IDUM
14128 063142 001007              BNE 2$          ;BRANCH IF NO
14129 063144 123727 072155 000011  1$: CMPB IDUM+11, #11 ;YES - LIMIT ROUND DIGIT TO 0 - 9
14130 063152 101403              BLOS 2$
14131 063154 142737 000010 072155  BICB #10, IDUM+11 ;CONVERT INVALID DIGIT TO A VALID ONE
14132 063162 022737 000003 072144  2$: CMP #3, IDUM    ;IS RANDOM INST = MOVTC
14133 063170 001006              BNE 3$          ;BRANCH IF NO
14134 063172 013737 072200 002172  MOV IDUM+34, IRXLT ;ADJUST IP15 FOR PROPER LEVEL OF INDIRECTING
14135 063200 012737 002172 072200  MOV #IRXLT, IDUM+34
14136 063206 000207              3$: RTS PC
14137
14138                          ;ROUTINE TO LOAD MISCELLANEOUS CONSTANTS USING RANDOM NUMBER GENERATOR
14139
14140 063210      LDCON:

```



14141 063210 004737 063532  
14142 063214 010037 001660  
14143 063220 004737 063532  
14144 063224 010037 001662  
14145 063230 000207

JSR PC,RN  
MOV R0,INCSQ1  
JSR PC,RN  
MOV R0,INCSQ2  
RTS PC

;LOAD TEST BUFFER INCREMENTING SEQUENCE  
; SEED WITH A RANDOM #

14146  
14147  
14148  
14149  
14150  
14151  
14152  
14153  
14154  
14155  
14156

:ROUTINE TO ACKNOWLEDGE OPERATOR REQUESTS  
:CNTL T - DISPLAY CURRENT TEST #(DECIMAL) THEN RETURN TO CALL+6  
:CNTL C - RETURN TO CALL+2  
:CNTL D - SET DISPLAY AND NO QUERY SWITCH. THEN RETURN TO CALL + 6  
:CNTL E - SET DISPLAY SWITCH. THEN RETURN TO CALL + 6  
:CNTL N - CLEAR DISPLAY SWITCH. THEN RETURN TO CALL + 6  
:CNTL O - TOGGLE PROGRESS DISPLAY SWITCH. THEN RETURN TO CALL +6  
:OTHER - RETURN TO CALL+6

14157 063232  
14158 063232 005737 111202  
14159 063236 001406  
14160 063240 013737 111202 064776  
14161 063246 005037 111202  
14162 063252 000406  
14163 063254 105777 116406  
14164 063260 100063  
14165 063262 117737 116424 064776  
14166 063270 042737 177600 064776  
14167 063276 023727 064776 000003  
14168 063304 001453  
14169 063306 023727 064776 000024  
14170 063314 001003  
14171 063316 004737 063436  
14172 063322 000442  
14173 063324 023727 064776 000004  
14174 063332 001007  
14175 063334 012737 177777 002042  
14176 063342 012737 177777 002204  
14177 063350 000427  
14178 063352 023727 064776 000005  
14179 063360 001004  
14180 063362 012737 177777 002042  
14181 063370 000417  
14182 063372 023727 064776 000017  
14183 063400 001003  
14184 063402 005137 002044  
14185 063406 000410  
14186 063410 023727 064776 000016  
14187 063416 001004  
14188 063420 005037 002042  
14189 063424 005037 002204  
14190 063430 062716 000004  
14191 063434 000207

EXTBK:  
TST XOCHAR ;DID TYPE ROUTINE XON/XOFF CHECK FIND A CHAR  
BEQ 7\$ ;BRANCH IF NO  
MOV XOCHAR,RCHAR  
CLR XOCHAR  
BR 8\$  
7\$: TSTB @TKS ;CHAR THERE?  
BPL 1\$ ;NO - EXIT ROUTINE  
MOV @TKB,RCHAR ;READ AND SAVE TTY CHAR  
8\$: BIC #^C177,RCHAR ;GET RID OF JUNK IF ANY  
CMP RCHAR,#003 ;IS CHAR A CNTL C?  
BEQ 2\$ ;BRANCH IF YES  
CMP RCHAR,#024 ;IS CHAR A CNTL T?  
BNE 3\$  
JSR PC,IDINST  
BR 1\$  
3\$: CMP RCHAR,#004 ;IS CHAR A CNTL D?  
BNE 5\$  
MOV #177777,NOERDS ;SET DISPLAY SWITCH  
MOV #177777,QRVFLG ;SET QUERY FOR DISPLAY BUFFER FLAG  
BR 1\$  
5\$: CMP RCHAR,#005 ;IS CHAR A CNTL E?  
BNE 6\$ ;BRANCH IF NO  
MOV #177777,NOERDS ;SET DISPLAY SWITCH  
BR 1\$  
6\$: CMP RCHAR,#017 ;IS CHAR A CNTL O?  
BNE 4\$ ;BRANCH IF NO  
COM PROGD ;TOGGLE PROGRESS DISPLAY SWITCH  
BR 1\$  
4\$: CMP RCHAR,#016 ;IS CHAR A CNTL N?  
BNE 1\$  
CLR NOERDS ;YES - CLEAR NO ERROR DISPLAY SWITCH  
CLR QRVFLG ;CLEAR QUERY SWITCH  
1\$: ADD #4,(SP) ;RETURN TO CALL+6  
2\$: RTS PC

14192  
14193  
14194

:ROUTINE TO DISPLAY CURRENT INST AND TEST #

```

14195 063436 IDINST:
14196 063436 012737 177777 001762 MOV #177777,CTACT ;SET CONTROL T ACTIVE FLAG
14197 063444 PRINTB #FORM21 ;PRINT A CRLF
(6) 063444 012746 014074 MOV #FORM21,-(SP)
(3) 063450 010600 MOV SP,R0
(4) 063452 004737 065410 JSR PC,FPRINT
14198 063456 004777 116460 JSR PC,@EMPTR ;PRINT INST & TEST #
14199 063462 000207 RTS PC
14200
14201
14202 ;ROUTINE TO RANDOMIZE PACKED STRING DATA TYPE
14203
14204 063464 RPTYPE:
14205 063464 004737 063532 JSR PC,RN ;GET A RANDOM #
14206 063470 032700 000001 BIT #1,R0 ;USE BIT 0 OF THE RANDOM # TO SELECT BETWEEN
14207 063474 001403 BEQ 1$ ; THE TWO TYPES FOR PACKED STRINGS (6,7).
14208 063476 012700 000007 MOV #7,R0
14209 063502 000402 BR 10$
14210 063504 012700 000006 1$: MOV #6,R0
14211 063510 000207 10$: RTS PC
14212
14213 ;ROUTINE TO RANDOMIZE ZONED STRING DATA TYPES
14214
14215 063512 RZTYPE:
14216 063512 004737 063532 JSR PC,RN ;GET A RANDOM #
14217 063516 042700 177770 BIC #177770,R0 ;USE BITS 0,1 & 2 TO SELECT BETWEEN
14218 063522 020027 000005 CMP R0,#5 ; THE 6 TYPES FOR ZONED STRINGS
14219 063526 101371 BHI RZTYPE
14220 063530 000207 RTS PC
14221
14222 ;ROUTINE TO GENERATE A PSEUDO RANDOM NUMBER
14223
14224 ;INPUTS: NONE
14225 ;OUTPUTS: PSEUDO RANDOM VALUE IN R0
14226
14227
14228 RN:
14229 063532 MOV RP1,R0
14230 063536 000241 CLC
14231 063540 005337 063612 DEC RNCON
14232 063544 006100 ROL R0
14233 063546 006100 ROL R0
14234 063550 063700 063612 ADD RNCON,R0
14235 063554 063700 063616 ADD RP2,R0
14236 063560 010037 063614 MOV R0,RP1
14237 063564 006100 ROL R0
14238 063566 006100 ROL R0
14239 063570 063700 063616 ADD RP2,R0
14240 063574 006100 ROL R0
14241 063576 006100 ROL R0
14242 063600 010037 063616 MOV R0,RP2
14243 063604 013700 063614 MOV RP1,R0
14244 063610 000207 RTS PC
14245 063612 000000 RNCON: .WORD 0 ;RANDOM # GENERATOR SEEDS

```

14246 063614 001233  
14247 063616 007622  
14248 063620 000000  
14249 063622 001233  
14250 063624 007622  
14251

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

14253					.SBTTL	MESSAGE PRINT ROUTINES
14254					:+++++	
14255					:	
14256					:ERROR MESSAGE PRINT ROUTINES	
14257					: RETURNS TO CALL +2 FOR REPEAT TEST	
14258					: RETURNS TO CALL +6 FOR NORMAL RETURN	
14259					:	
14260					:-----	
14261	063626				INSERR:	
14262	063626	004777	116310		JSR PC,@EMPTR	;PRINT ERROR MESSAGE HEADER
14263	063632	032737	000100	050012	BIT #100,TINST	;INST UNDER TEST TYPE?
14264	063640	001406			BEQ 11\$	;BRANCH IF REGISTER TYPE
14265	063642				PRINTB #INMEM	;IN-LINE TYPE
(6)	063642	012746	011120		MOV #INMEM,-(SP)	
(3)	063646	010600			MOV SP,R0	
(4)	063650	004737	065410		JSR PC,FPRINT	
14266	063654	000405			BR 12\$	
14267	063656				11\$: PRINTB #INREG	
(6)	063656	012746	011072		MOV #INREG,-(SP)	
(3)	063662	010600			MOV SP,R0	
(4)	063664	004737	065410		JSR PC,FPRINT	
14268	063670				12\$: PRINTB #FORM13,ER0,ER1,ER2,ER3,ER4,ER5,TR6,<B,TCC>	
(14)	063670	005046			CLR -(SP)	
(14)	063672	153716	003646		BISB TCC,(SP)	
(13)	063676	013746	003644		MOV TR6,-(SP)	
(12)	063702	013746	003702		MOV ER5,-(SP)	
(11)	063706	013746	003700		MOV ER4,-(SP)	
(10)	063712	013746	003676		MOV ER3,-(SP)	
(9)	063716	013746	003674		MOV ER2,-(SP)	
(8)	063722	013746	003672		MOV ER1,-(SP)	
(7)	063726	013746	003670		MOV ER0,-(SP)	
(6)	063732	012746	011174		MOV #FORM13,-(SP)	
(5)	063736	010600			MOV SP,R0	
(4)	063740	004737	065410		JSR PC,FPRINT	
14269	063744				PRINTB #EMOUT	
(6)	063744	012746	011146		MOV #EMOUT,-(SP)	
(3)	063750	010600			MOV SP,R0	
(4)	063752	004737	065410		JSR PC,FPRINT	
14270	063756				PRINTB #FORM14,ER0R,ER1R,ER2R,ER3R,ER4R,ER5R,ER6R,<B,ECCR>	
(14)	063756	005046			CLR -(SP)	
(14)	063760	153716	003724		BISB ECCR,(SP)	
(13)	063764	013746	003722		MOV ER6R,-(SP)	
(12)	063770	013746	003720		MOV ER5R,-(SP)	
(11)	063774	013746	003716		MOV ER4R,-(SP)	
(10)	064000	013746	003714		MOV ER3R,-(SP)	
(9)	064004	013746	003712		MOV ER2R,-(SP)	
(8)	064010	013746	003710		MOV ER1R,-(SP)	
(7)	064014	013746	003706		MOV ER0R,-(SP)	
(6)	064020	012746	011254		MOV #FORM14,-(SP)	
(3)	064024	010600			MOV SP,R0	
(4)	064026	004737	065410		JSR PC,FPRINT	
14271	064032	005737	002146		TST ERRREG	;WAS THERE A REGISTER ERROR?
14272	064036	001440			BEQ 1\$	
14273	064040				PRINTB #ACOUT	;YES - PRINT OUT DISCREPANCIES

(6)	064040	012746	011334	MOV	#ACOUT,-(SP)	
(3)	064044	010600		MOV	SP,RO	
(4)	064046	004737	065410	JSR	PC,FPRINT	
14274	064052	012701	003650	MOV	#TROR,R1	
14275	064056	012702	003706	MOV	#EROR,R2	
14276	064062	021122		5\$:	CMP	(R1),(R2)+ ;COMPARE ACTUAL WITH EMULATOR REGS.
14277	064064	001412			BEQ	2\$
14278	064066	011137	002270		MOV	(R1),TERR ;NOT EQUAL - PRINT ACTUAL
14279	064072				PRINTB	#FORM15,TERR
(7)	064072	013746	002270		MOV	TERR,-(SP)
(6)	064076	012746	011362		MOV	#FORM15,-(SP)
(3)	064102	010600			MOV	SP,RO
(4)	064104	004737	065410		JSR	PC,FPRINT
14280	064110	000405			BR	3\$
14281	064112			2\$:	PRINTB	#FORM16 ;EQUAL - PRINT SPACES
(6)	064112	012746	011371		MOV	#FORM16,-(SP)
(3)	064116	010600			MOV	SP,RO
(4)	064120	004737	065410		JSR	PC,FPRINT
14282	064124	020127	003664	3\$:	CMP	R1,#TR6R ;ALL REGISTERS COMPARED?
14283	064130	001424			BEQ	4\$ ;BRANCH IF YES
14284	064132	062701	000002		ADD	#2,R1
14285	064136	000751			BR	5\$ ;LOOK AT NEXT REGISTER
14286	064140	005737	002144	1\$:	TST	ERRCC ;WAS THERE A CONDITION CODE ERROR?
14287	064144	001431			BEQ	6\$
14288	064146				PRINTB	#ACOUT ;YES - PRINT ACTUAL COND. CODES
(6)	064146	012746	011334		MOV	#ACOUT,-(SP)
(3)	064152	010600			MOV	SP,RO
(4)	064154	004737	065410		JSR	PC,FPRINT
14289	064160				PRINTB	#FORM17,<B,TCCR>
(7)	064160	005046			CLR	-(SP)
(7)	064162	153716	003666		BISB	TCCR,(SP)
(6)	064166	012746	011375		MOV	#FORM17,-(SP)
(5)	064172	010600			MOV	SP,RO
(4)	064174	004737	065410		JSR	PC,FPRINT
14290	064200	000413			BR	6\$
14291	064202	005737	002144	4\$:	TST	ERRCC
14292	064206	001410			BEQ	6\$
14293	064210				PRINTB	#FORM18,<B,TCCR>
(7)	064210	005046			CLR	-(SP)
(7)	064212	153716	003666		BISB	TCCR,(SP)
(6)	064216	012746	011405		MOV	#FORM18,-(SP)
(3)	064222	010600			MOV	SP,RO
(4)	064224	004737	065410		JSR	PC,FPRINT
14294	064230	004737	066346	6\$:	JSR	PC,PRNIB ;GO CHECK FOR POSSIBLE NIBBLE PRINTOUT.
14295	064234	005037	002274		CLR	FILLS2
14296	064240	005737	002150		TST	ERRBUF ;WAS THERE A BUFFER ERROR?
14297	064244	001422			BEQ	LODT
14298	064246				PRINTB	#EBUFO,EMADR,EMDTA ;YES PRINT FIRST BUFFER
(8)	064246	013746	002202		MOV	EMDTA,-(SP)
(7)	064252	013746	002200		MOV	EMADR,-(SP)
(6)	064256	012746	011411		MOV	#EBUFO,-(SP)
(3)	064262	010600			MOV	SP,RO
(4)	064264	004737	065410		JSR	PC,FPRINT
14299	064270				PRINTB	#ABUFO,AEADR,AEDTA ; BYTE DISCREPANCY.

(8)	064270	013746	002176	MOV	AEDTA,-(SP)		
(7)	064274	013746	002174	MOV	AEADR,-(SP)		
(6)	064300	012746	011446	MOV	#ABUFO,-(SP)		
(3)	064304	010600		MOV	SP,RO		
(4)	064306	004737	065410	JSR	PC,FPRINT		
14300	064312						
14301	064312	005737	002204	LODT:	TST	QRYFLG	;INHIBIT BUFFER QUERY?
14302	064316	001110			BNE	ERMDON	;YES
14303	064320	104400			TYPE		
14304	064322	011477			QDISP		
14305	064324	004737	064546		JSR	PC,YORN	;DISPLAY BUFFER?
14306	064330	000137	064514		JMP	3\$	;CONTINUE (C) RETURN
14307	064334	000137	064350		JMP	1\$	;DISPLAY MEMORY (D) RETURN
14308	064340	000137	064544		JMP	LERMD	;REPEAT TEST (R) RETURN
14309	064344	000137	064534		JMP	RTSUPV	;RESTART (S) RETURN
14310	064350	104400		1\$:	TYPE		;PRINT 'ADDR(S)?'
14311	064352	013703			AST		
14312	064354	004737	065000		JSR	PC,RANGE	;GET RANGE OF LOCATIONS TO DISPLAY
14313	064360	000754			BR	LODT	;NO MORE DISPLAY REQUESTED-RETURN
14314	064362				PRINTB	#ADDHDR	;PRINT BYTE HEADER
(6)	064362	012746	013755		MOV	#ADDHDR,-(SP)	
(3)	064366	010600			MOV	SP,RO	
(4)	064370	004737	065410		JSR	PC,FPRINT	
14315	064374	004737	065334	2\$:	JSR	PC,FILLPB	;FILL PRINT BUFFER
14316	064400	000763			BR	1\$	;RANGE EXHAUSTED RETURN
14317	064402				PRINTB	#FORM19,BAD,<B,PB0>,<B,PB1>,<B,PB2>,<B,PB3>	
(11)	064402	005046			CLR	-(SP)	
(11)	064404	153716	003145		BISB	PB3,(SP)	
(10)	064410	005046			CLR	-(SP)	
(10)	064412	153716	003144		BISB	PB2,(SP)	
(9)	064416	005046			CLR	-(SP)	
(9)	064420	153716	003143		BISB	PB1,(SP)	
(8)	064424	005046			CLR	-(SP)	
(8)	064426	153716	003142		BISB	PB0,(SP)	
(7)	064432	013746	002226		MOV	BAD,-(SP)	
(6)	064436	012746	013716		MOV	#FORM19,-(SP)	
(3)	064442	010600			MOV	SP,RO	
(4)	064444	004737	065410		JSR	PC,FPRINT	
14318	064450				PRINTB	#FORM20,<B,PB4>,<B,PB5>,<B,PB6>,<B,PB7>	
(10)	064450	005046			CLR	-(SP)	
(10)	064452	153716	003151		BISB	PB7,(SP)	
(9)	064456	005046			CLR	-(SP)	
(9)	064460	153716	003150		BISB	PB6,(SP)	
(8)	064464	005046			CLR	-(SP)	
(8)	064466	153716	003147		BISB	PB5,(SP)	
(7)	064472	005046			CLR	-(SP)	
(7)	064474	153716	003146		BISB	PB4,(SP)	
(6)	064500	012746	014041		MOV	#FORM20,-(SP)	
(3)	064504	010600			MOV	SP,RO	
(4)	064506	004737	065410		JSR	PC,FPRINT	
14319	064512	000730			BR	2\$	
14320	064514	005737	002216	3\$:	TST	STOPTF	;IS THE STOP TEST FLAG SET?
14321	064520	001404			BEQ	4\$	;BRANCH IF NO
14322	064522	005037	002216		CLR	STOPTF	;CLEAR FLAG WHICH WILL CAUSE PROG TO

```

14323
14324 064526 005037 002042
14325 064532 000402
14326 064534 062716 000004
14327 064540 062716 000004
14328 064544 000207
14329
14330
14331
14332
14333 064546
14334 064546 105777 115114
14335 064552 100375
14336 064554 117737 115132 064776
14337 064562 032737 000100 064776
14338 064570 001403
14339 064572 042737 000040 064776
14340 064600 042737 177600 064776
14341 064606 023727 064776 000123
14342 064614 001450
14343 064616 023727 064776 000110
14344 064624 001003
14345 064626 005037 047772
14346
14347 064632 000407

```

```

CLR NOERDS
BR ERMDON
4$:
RTSUPV: ADD #4,(SP)
ERMDON: ADD #4,(SP)
LERMD: RTS PC
; TO QUERY FOR STOP TEST # AGAIN
; RETURN TO RESTART AT LOC 'START'
; NORMAL RETURN TO CALL +6
; REPEAT TEST RETURN TO CALL +2
; SUBROUTINE TO ACCEPT Y,N,C,P,S,D OR H RESPONSE FROM TTY. RETURNS TO CALL +2
; ON N OR C RESPONSE; CALL +4 ON A Y OR D RESPONSE; CALL +6 ON AN R OR H RESPONSE;
; AND CALL +10 ON AN S RESPONSE.
YORN:
1$: TSTB @TKS ;WAIT FOR A CHARACTER
BPL 1$
MOVB @TKB,RCHAR ;READ & SAVE CHAR
BIT #100,RCHAR ;CONVERT LOWER CASE INPUT TO UPPER
BEQ 12$ ; CASE LETTERS
BIC #40,RCHAR
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$

```

14349	064634	023727	064776	000122	6\$:	CMP	RCHAR,#122	;IS CHAR R
14350	064642	001011				BNE	7\$	;BRANCH IF NO
14351	064644	013737	001670	047772		MOV	KNOP,PREINS	;RESTORE NOP TO INST IMMED BEFOR CIS INST UNDER TEST
14352	064652	012737	177777	002272	10\$:	MOV	#177777,RPTFLG	;SET REPEAT TEST FLAG
14353	064660	005337	002054			DEC	ERRCI	;DECREMENT ERROR COUNT SO THAT ERROR COUNT
14354								; DOESN'T ADVANCE ON REPEAT OF TEST
14355	064664	000426				BR	4\$	
14356	064666	023727	064776	000131	7\$:	CMP	RCHAR,#131	;IS CHAR = Y
14357	064674	001424				BEQ	2\$	
14358	064676	023727	064776	000104		CMP	RCHAR,#104	;IS CHAR = D?
14359	064704	001420				BEQ	2\$	
14360	064706	023727	064776	000116		CMP	RCHAR,#116	;NO - IS CHAR = N
14361	064714	001404				BEQ	11\$	
14362	064716	023727	064776	000103		CMP	RCHAR,#103	;IS CHAR = C?
14363	064724	001310				BNE	1\$	
14364	064726	013737	001670	047772	11\$:	MOV	KNOP,PREINS	;RESTORE NOP TO INST IMMED BEFORE CIS INST UNDER TEST
14365	064734	000406				BR	3\$	;YES - RETURN - CALL +2
14366	064736	062716	000004		5\$:	ADD	#4,(SP)	
14367	064742	062716	000004		4\$:	ADD	#4,(SP)	
14368	064746	062716	000004		2\$:	ADD	#4,(SP)	;CHAR = Y OR D SETUP RETURN - CALL +4
14369	064752	004737	064760		3\$:	JSR	PC,ECHAR	;ECHO CHARACTER - WAIT FOR



14371	064756	000207			RTS	PC	:	PRINTER READY
14372							:	LOAD CHAR TO BE TYPED INTO DATA REG.
14373	064760	105777	114730	ECHAR:	TSTB	@TPS	:	SUBROUTINE TO PRINT CHAR IN 'RCHAR'
14374	064764	100375			BPL	ECHAR	:	WAIT UNTIL PRINTER IS READY
14375	064766	113777	064776	114722	MOVB	RCHAR,@TPB	:	LOAD CHAR INTO DATA REG
14376	064774	000207			RTS	PC		

```

14378
14379 064776 000000 RCHAR: .WORD 0
14380
14381 ;SUBROUTINE TO GET RANGE OF LOCATIONS TO DISPLAY.
14382 ; RETURNS TO CALL +2 ON NO MORE DISPLAY REQUESTED - USER
14383 ; RESPONDED WITH IMMEDIATE 'CR'.
14384 ; NORMAL RETURN IS TO CALL +4 WITH LOWER DISPLAY LIMIT
14385 ; IN 'RLL' AND UPPER DISPLAY LIMIT IN 'RUL'.
14386 065000 RANGE: ;NORMAL RETURN = CALL +4.
14387 065000 004737 065102 JSR PC,ACCOCT ;GET RANGE LOWER LIMIT
14388 065004 000207 RTS PC ;RETURN - EXIT DISPLAY
14389 065006 000411 BR 1$ ;RETURN - SINGLE LIMIT SPECIFIED
14390 065010 012637 002222 MOV (SP)+,RLL ;NORMAL RETURN - SAVE LOWER LIMIT
14391 065014 004737 065102 JSR PC,ACCOCT ;GET RANGE UPPER LIMIT
14392 065020 000207 RTS PC ;RETURN - EXIT DISPLAY
14393 065022 000411 BR 2$ ;NORMAL RETURN - SAVE UPPER LIMIT
14394 065024 104400 TYPE ;RETURN - TYPE? <CR><LF>REENTER:
14395 065026 015552 QUES
14396 065030 000763 BR RANGE ;TRY AGAIN
14397 065032 012637 002222 1$: MOV (SP)+,RLL ;SINGLE LIMIT SPECIFIED - SAVE AS
14398 065036 013737 002222 002224 MOV RLL,RUL ; BOTH LOWER & UPPER LIMIT
14399 065044 000402 BR 3$ ;EXIT
14400 065046 012637 002224 2$: MOV (SP)+,RUL ;SAVE UPPER LIMIT
14401 065052 042737 000007 002222 3$: BIC #7,RLL ;ROUND OFF RANGE TO GROUP OF
14402 065060 042737 000007 002224 BIC #7,RUL ;TEN BYTES.
14403 065066 062737 000010 002224 ADD #10,RUL
14404 065074 062716 000002 ADD #2,(SP) ;EXIT TO CALL +4
14405 065100 000207 RTS PC
14406
14407 ;SUBROUTINE TO ACCEPT OCTAL # FROM TTY. RETURNS TO CALL +2 ON INITIAL CR.
14408 ; RETURNS TO CALL +4 ON <CR> OR / WITH LIMIT ON STACK. RETURNS TO CALL +6 ON <-> WITH
14409 ; LIMIT ON STACK.
14410 065102 ACCOCT:
14411 065102 005046 CLR -(SP) ;CLEAR STORAGE FOR OCTAL #
14412 065104 105777 114556 1$: TSTB @TKS ;CHAR THERE?
14413 065110 100375 BPL 1$ ;NO - WAIT
14414 065112 117746 114574 MOVB @TKB,-(SP) ;SAVE THE CHAR
14415 065116 042716 177600 BIC #^C177,(SP) ;STRIP-OFF THE ASCII
14416 065122 022726 000015 CMP #15,(SP)+ ;IS IT A 'CR'?
14417 065126 001005 BNE -12$
14418 065130 104400 TYPE ;YES - ECHO CR & LF
14419 065132 015566 XCRLF
14420 065134 062706 000002 ADD #2,SP ;RETURN TO CALL +2
14421 065140 000207 RTS PC
14422 065142 024627 000055 2$: CMP -(SP),#55 ;IS CHAR = '-'
14423 065146 001462 BEQ 6$ ;BRANCH IF YES
14424 065150 021627 000057 3$: CMP (SP),#57 ;IS CHAR A / ?
14425 065154 001403 BEQ 31$
14426 065156 021627 000015 CMP (SP),#15 ;IS CHAR A <CR>?
14427 065162 001016 BNE 4$
14428 065164 104400 31$: TYPE ;YES - ECHO/<CR> AND <LF>
14429 065166 015571 SLCRLF
14430 065170 016616 000002 7$: MOV 2(SP),(SP) ;SWAP POSITION OF OCTAL #
14431 065174 016666 000004 000002 MOV 4(SP),2(SP) ;AND RETURN PC ON STACK

```

```

14432 065202 011666 000004      MOV      (SP),4(SP)
14433 065206 062706 000002      ADD      #2,SP
14434 065212 062716 000002      ADD      #2,(SP)      ;UPDATE RETURN POINTER
14435 065216 000207              RTS      PC      ;RETURN WITH OCTAL LIMIT ON STACK
14436 065220 011637 064776      4$:     MOV      (SP),RCHAR      ;ECHO CHAR ACCEPTED
14437 065224 004737 064760      JSR      PC,ECHAR
14438
14439 065230 021627 000060      CMP      (SP),#60      ;CHAR <0?
14440 065234 002422      BLT      5$      ;BRANCH IF YES
14441 065236 021627 000067      CMP      (SP),#67      ;CHAR>7?
14442 065242 003017      BGT      5$      ;BRANCH IF YES
14443 065244 042726 000060      BIC      #60,(SP)+      ;STRIP OFF ASCII
14444 065250 006316      ASL      (SP)      ;SHIFT PRESENT DATA OVER TO
14445 065252 006316      ASL      (SP)      ; MAKE ROOM FOR NEW DIGIT
14446 065254 006316      ASL      (SP)
14447 065256 056616 177776      BIS      -2(SP),(SP)      ;SET IN NEW DIGIT
14448 065262 105777 114400      10$:    TSTB     @TKS      ;CHAR THERE
14449 055266 100375              BPL 10$      ;NO - WAIT
14450 065270 117746 114416      MOVB     @TKB,-(SP)      ;SAVE CHAR
14451 065274 042726 177600      BIC     #^C177,(SP)+
14452 065300 000720      BR 2$
14453 065302 104400      5$:     TYPE      ;TYPE ?<CR><LF>*
14454 065304 015552      QUES
14455 065306 062706 000002      ADD      #2,SP
14456 065312 000674      BR      1$
14457 065314 011637 064776      6$:     MOV      (SP),RCHAR      ;ECHO '-'
14458 065320 004737 064760      JSR      PC,ECHAR
14459 065324 062766 000002 000004      ADD      #2,4(SP)      ;UPDATE RETURN POINTER
14460 065332 000716      BR      7$
14461
14462      ;SUBROUTINE TO FILL BYTE PRINT BUFFER. RETURNS TO CALL +2 WHEN DISPLAY REQUEST IS
14463      ;COMPLETE (RLL=RUL). NORMAL RETURN TO CALL+4 WITH RLL=RLL +10 & PRINT BUFFER FILLED.
14464 065334 023737 002222 002224  FILLPB:  CMP      RLL,RUL
14465 065342 001421      BEQ      1$      ; NORMAL RETURN TO CALL +4
14466 065344 013701 002222      MOV      RLL,R1      ;SETUP POINTER TO DISPLAY LOCS
14467 065350 010137 002226      MOV      R1,BAD      ;SAVE BUFFER ADDRESS FOR PRINTOUT
14468 065354 012137 003142      MOV      (R1)+,PB0      ;TRANSFER 10 BYTES AT DISPLAY
14469 065360 012137 003144      MOV      (R1)+,PB2      ; LOC ADDRESS TO PRINT
14470 065364 012137 003146      MOV      (R1)+,PB4      ; BUFFER.
14471 065370 012137 003150      MOV      (R1)+,PB6
14472 065374 062737 000010 002222      ADD      #10,RLL      ;UPDATE LOWER LIMIT DISPLAY POINTER
14473 065402 062716 000002      ADD      #2,(SP)      ;UPDATE RETURN POINTER
14474 065406 000207      1$:     RTS      PC
14475
14476      .EVEN
14477
14478      ;SUBROUTINE TO TYPE FORMATED 'PRINTB' STATEMENTS
14479
14480      FPRINT:
14481 065410 010537 002062      MOV      R5,FSAVR5      ;SAVE REGISTERS
14482 065414 010437 002064      MOV      R4,FSAVR4
14483 065420 010337 002066      MOV      R3,FSAVR3
14484 065424 010237 002070      MOV      R2,FSAVR2
14485 065430 010137 002072      MOV      R1,FSAVR1

```

14486	065434	012001			MOV (R0)+,R1	;SETUP R1 AS POINTER INTO FORMAT STATEMENT	
14487	065436	112102		1\$:	MOVB (R1)+,R2	;GET NEXT FORMAT BYTE	
14488	065440	020227	000045		CMP R2,#'X	;IS BYTE = X ?	
14489	065444	001774			BEQ 1\$	;BRANCH IF YES	
14490	065446	020227	000117		CMP R2,#'0	;IS BYTE = 0 ? (OCTAL)	
14491	065452	001426			BEQ 2\$	;BRANCH IF YES	
14492	065454	020227	000101		CMP R2,#'A	;IS BYTE = A ? (ASCII)	
14493	065460	001432			BEQ 3\$	;BRANCH IF YES	
14494	065462	020227	000116		CMP R2,#'N	;IS BYTE = N ? (CRLF)	
14495	065466	001443			BEQ 4\$	;BRANCH IF YES	
14496	065470	020227	000131		CMP R2,#'Y	;IS BYTE = Y ? (BINARY)	
14497	065474	001446			BEQ 5\$	;BRANCH IF YES	
14498	065476	020227	000104		CMP R2,#'D	;IS BYTE = D ? (DECIMAL)	
14499	065502	001452			BEQ 6\$	;BRANCH IF YES	
14500	065504	020227	000123		CMP R2,#'S	;IS BYTE = S ? (SPACE)	
14501	065510	001463			BEQ 7\$	;BRANCH IF YES	
14502	065512	020227	000000		CMP R2,#0	;IS BYTE = 0 ? (END OF FORMAT STATEMENT)	
14503	065516	001512			BEQ 10\$	;BRANCH IF YES	
14504	065520	020227	000102		CMP R2,#'B	;IS BYTE = B ? (BYTE)	
14505	065524	001524			BEQ 11\$	;BRANCH IF YES	
14506	065526	000743			BR 1\$	;BYTE = NONE OF THE ABOVE - IGNORE IT.	
14507							
14508	065530	112102		2\$:	MOVB (R1)+,R2	;SET R2 = COUNT OF # OF DIGITS TO PRINT	
14509	065532	042702	177770		BIC #177770,R2		
14510	065536	012003			MOV (R0)+,R3	;SET R3 = WORD OF DIGITS TO PRINT	
14511	065540	004737	066020		JSR PC,POCT	;CALL ROUTINE TO PRINT OCTAL DIGITS	
14512	065544	000734			BR 1\$		
14513	065546	112102		3\$:	MOVB (R1)+,R2	;SET R2 = NEXT ASCII CHAR TO PRINT	
14514	065550	022702	000045		CMP #'X,R2	;IS CHAR = X	
14515	065554	001730			BEQ 1\$	;BRANCH IF YES	
14516	065556	022702	000000		CMP #0,R2	;END OF FORMAT BYTES?	
14517	065562	001470			BEQ 10\$	;BRANCH IF YES	
14518	065564	110237	066122		MOVB R2,TDIG	;YES - PREPARE TO EXIT ROUTINE	
14519	065570	104400	066122		TYPE ,TDIG	;CALL ROUTINE TO PRINT ASCII BYTE	
14520	065574	000764			BR 3\$		
14521	065576	012737	000200	066122	4\$:	MOV #CRLF,TDIG	;CALL ROUTINE TO PRINT CRLF
14522	065604	104400	066122		TYPE ,TDIG		
14523	065610	000712			BR 1\$		
14524	065612	112102		5\$:	MOVB (R1)+,R2	;SET R2 = COUNT OF # OF DIGITS TO PRINT	
14525	065614	042702	177770		BIC #177770,R2		
14526	065620	012003			MOV (R0)+,R3	;SET R3 = WORD OF DIGITS TO PRINT	
14527	065622	004737	066124		JSR PC,PBIN	;CALL ROUTINE TO PRINT BINARY DIGITS	
14528	065626	000703			BR 1\$		
14529	065630	112102		6\$:	MOVB (R1)+,R2	;SET R2 = COUNT OF DIGITS TO PRINT	
14530	065632	042702	177770		BIC #177770,R2		
14531	065636	020227	000006		CMP R2,#6	;IF REQUEST IS TO PRINT MORE THAN 5 DIGITS	
14532	065642	103402			BLO 61\$	; PRINT 5 INSTEAD	
14533	065644	012702	000005		MOV #5,R2		
14534	065650	012003		61\$:	MOV (R0)+,R3	;SET R3 = WORD OF DIGITS TO PRINT	
14535	065652	004737	066216		JSR PC,PDEC	;CALL ROUTINE TO CONVERT (R3) TO DECIMAL	
14536	065656	000667			BR 1\$	; AND PRINT DECIMAL DIGITS	
14537	065660	112102		7\$:	MOVB (R1)+,R2	;GET MOST SIGN DIGIT OF 1 OR 2 DIGIT	
14538						; COUNT OF # OF SPACES TO PRINT	
14539	065662	042702	177770		BIC #177770,R2		

14540	065666	121127	000045		CMPB (R1),#'X	;IS NEXT BYTE = X ?
14541	065672	001415			BEQ 12\$	
14542	065674	121127	000000		CMPB (R1),#0	
14543	065700	001412			BEQ 12\$	
14544	065702	006302			ASL R2	
14545	065704	006302			ASL R2	
14546	065706	006302			ASL R2	
14547	065710	112137	066120		MOVB (R1)+,OCNT	
14548	065714	142737	000370	066120	BICB #370,OCNT	
14549	065722	153702	066120		BISB OCNT,R2	;GET LEAST SIGN DIGIT INTO R2
14550	065726	012737	000040	066122	12\$: MOV #' ,TDIG	;PRINT A SPACE
14551	065734	104400	066122		TYPE ,TDIG	
14552	065740	077206			SOB R2,12\$	
14553	065742	000635			BR 1\$	
14554	065744	011640			10\$: MOV (SP),-(R0)	
14555	065746	010006			MOV R0,SP	
14556	065750	013705	002062		MOV FSAVR5,R5	
14557	065754	013704	002064		MOV FSAVR4,R4	
14558	065760	013703	002066		MOV FSAVR3,R3	
14559	065764	013702	002070		MOV FSAVR2,R2	
14560	065770	013701	002072		MOV FSAVR1,R1	
14561	065774	000207			RTS PC	
14562						
14563	065776	112102			11\$: MOVB (R1)+,R2	;SET R2 = COUNT OF # OF DIGITS TO PRINT
14564	066000	042702	177770		BIC #177770,R2	
14565	066004	012003			MOV (R0)+,R3	;SET R3 = BYTE TO PRINT
14566	066006	042703	177400		BIC #177400,R3	
14567	066012	004737	066020		JSR PC,POCT	
14568	066016	000607			BR 1\$	
14569						
14570						
14571						
14572						
14573						
14574	066020	112737	000005	066120	POCT: MOVB #5,OCNT	;SET THE ITERATION COUNT
14575	066026	005402			NEG R2	
14576	066030	062702	000006		ADD #6,R2	;SUBTRACT # OF DIGITS TO TYPE FROM MAX ALLOWED
14577	066034	110237	066121		MOVB R2,OMODE	;SAVE IT FOR USE
14578	066040	005004			CLR R4	
14579	066042	006103			1\$: ROL R3	;ROTATE MSB INTO 'C'
14580	066044	000404			BR 3\$	
14581	066046	006103			2\$: ROL R3	;FORM THIS DIGIT
14582	066050	006103			ROL R3	
14583	066052	006103			ROL R3	
14584	066054	010304			MOV R3,R4	
14585	066056	006104			3\$: ROL R4	;GET LSB OF THIS DIGIT
14586	066060	105337	066121		DECB OMODE	;TYPE THIS DIGIT
14587	066064	100010			BPL 7\$	;BRANCH IF NO
14588	066066	042704	177770		BIC #177770,R4	;GET RID OF JUNK
14589	066072	052704	000060		BIS #'0,R4	;MAKE THIS DIGIT ASCII
14590	066076	110437	066122		MOVB R4,TDIG	;SAVE FOR TYPING
14591	066102	104400	066122		TYPE ,TDIG	;TYPE THIS DIGIT
14592	066106	105337	066120		7\$: DECB OCNT	;COUNT BY 1
14593	066112	002401			BLT 6\$	;BRANCH IF DONE

```

:SUBROUTINE TO CONVERT A BINARY # TO OCTAL (ASCII) AN TYPE IT
:ENTER WITH R2 = # OF OCTAL DIGITS TO TYPE
:          R3 = BINARY #
:
:

```

```

14594 066114 000754          BR 2$          ;BRANCH IF MORE TO DO
14595 066116 000207          6$:          RTS PC
14596
14597 066120      000          OCNT:        .BYTE 0
14598 066121      000          OMODE:       .BYTE 0
14599 066122      000          TDIG:        .BYTE 0
14600 066123      000          .           .BYTE 0
14601
14602          ;SUBROUTINE TO CHANGE A BINARY # TO ASCII AND TYPE IT
14603          ;ENTER WITH R2 = # OF BINARY DIGITS TO TYPE
14604          ;R3 = BINARY #
14605
14606 066124 112737 000017 066120 PBIN:  MOVB #17,OCNT      ;SET THE ITERATION COUNT
14607 066132 005402          NEG R2
14608 066134 062702 000020          ADD #20,R2      ;SUBTRACT # OF DIGITS TO TYPE FROM MAX ALLOWED
14609 066140 110237 066121          MOVB R2,OMODE   ;SAVE IT FOR USE
14610 066144 005004          CLR R4          ;CLEAR THE OUTPUT WORD
14611 066146 006103          1$:          ROL R3
14612 066150 012704 000000          MOV #0,R4
14613 066154 006104          ROL R4          ;GET BINARY DIGIT
14614 066156 105337 066121          DECB OMODE      ;TYPE THIS DIGIT?
14615 066162 100010          BPL 7$          ;BRANCH IF NO
14616 066164 042704 177776          BIC #177776,R4 ;GET RID OF JUNK
14617 066170 052704 000060          BIS #'0,R4      ;MAKE THIS BIT ASCII
14618 066174 010437 066122          MOV R4,TDIG     ;SAVE FOR TYPING
14619 066200 104400 066122          TYPE ,TDIG     ;TYPE THIS DIGIT
14620 066204 105337 066120          7$:          DECB OCNT       ;COUNT BY 1
14621 066210 002401          BLT 6$          ;BRANCH IF DONE
14622 066212 000755          BR 1$          ;BRANCH IF MORE TO DO
14623 066214 000207          6$:          RTS PC
14624
14625          ;
14626          ;SUBROUTINE TO CONVERT A BINARY # TO DECIMAL (ASCII) AND TYPE DECIMAL DIGITS
14627          ;ENTER WITH R3 = BINARY #
14628          ;R2 = # OF DECIMAL DIGITS TO TYPE
14629          ;
14630 066216 010146          PDEC:  MOV R1,-(SP)   ;SAVE R1
14631 066220 010046          MOV R0,-(SP)   ;SAVE R0
14632 066222 012700 000005          MOV #5,R0
14633 066226 160200          SUB R2,R0      ;R0 CONTAINS # OF DIGITS TO SKIP BEFORE PRINTING
14634 066230 005004          CLR R4          ;ZERO CONSTANTS TABLE INDEX
14635 066232 012705 066336          MOV #DBLK,R5   ;SETUP THE OUTPUT POINTER
14636 066236 005002          2$:          CLR R2          ;CLEAR THE BCD #
14637 066240 016401 066326          MOV DTBL(R4),R1 ;GET THE CONSTANT
14638 066244 160103          3$:          SUB R1,R3       ;FORM THIS BCD DIGIT
14639 066246 103402          BLO 4$          ;BRANCH IF DONE
14640 066250 005202          INC R2         ;INCREASE THE BCD DIGIT BY 1
14641 066252 000774          BR 3$
14642 066254 060103          4$:          ADD R1,R3       ;ADD BACK THE CONSTANT
14643 066256 052702 000060          6$:          BIS #'0,R2     ;MAKE THE BCD DIGIT ASCII
14644 066262 005700          TST R0         ;PRINT THIS DIGIT?
14645 066264 001402          BEQ 61$        ;BRANCH IF YES
14646 066266 005300          DEC R0         ;DECREMENT SKIP COUNT
14647 066270 000401          BR 62$

```

14648 066272 110225  
 14649 066274 005724  
 14650 066276 020427 000010  
 14651 066302 002755  
 14652 066304 003002  
 14653 066306 010302  
 14654 066310 000762  
 14655 066312 105015  
 14656 066314 104400 066336  
 14657 066320 012600  
 14658 066322 012601  
 14659 066324 000707  
 14660 066326 023420  
 14661 066330 001750  
 14662 066332 000144  
 14663 066334 000012  
 14664 066336 000004

```

61$: MOV B R2,(R5)+ ;PUT THIS CHAR IN THE OUTPUT BUFFER
62$: TST (R4)+ ;JUST INCREMENTING
      CMP R4,#10 ;CHECK THE TABLE INDEX
      BLT 2$ ;GO DO THE NEXT DIGIT
      BGT 8$ ;GO TO EXIT
      MOV R3,R2 ;GET LSD
      BR 6$ ;GO CHANGE TO ASCII
8$: CLR B (R5) ;SET THE TERMINATOR
      TYPE ,DBLK ;NOW TYPE THE #
      MOV (SP)+,R0 ;RESTORE R0
      MOV (SP)+,R1 ;RESTORE R1
      RTS PC ;EXIT
DTBL: 10000.
      1000.
      100.
      10.
DBLK: .BLKW 4
  
```

.....  
 : 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  
 INSRC1 AND INSRC2.

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.

14693 066346 005737 036210  
 14694 066352 001401  
 14695 066354 000207  
 14696 066356 032737 000004 002140  
 14697 066364 001401  
 14698 066366 000207  
 14699 066370 005737 002472  
 14700 066374 001001  
 14701 066376 000207

```

PRNIB: TST EZDF ;IS TEST CONDITION A 'DIVIDE BY ZERO'
        BEQ 1$
        RTS PC ;YES - EXIT WITHOUT DISPLAYING ANY BUFFER STRINGS
1$: BIT #4,SPHAND ;IS SPECIAL HANDLING BIT 2 SET?
        BEQ 2$
        RTS PC ;YES - EXIT WITHOUT DISPLAYING ANY BUFFER STRINGS.
2$: TST PZCODE ;ARE ANY STRINGS TO BE DISPLAYED?
        BNE 3$
        RTS PC ;NO - EXIT WITHOUT DISPLAY.
  
```

14702	066400	005037	026322		3\$:	CLR EPAK	
14703	066404	032737	000001	002472		BIT #1,PZCODE	;PRINT A ZONED SRC STRING?
14704	066412	001402				BEQ 4\$	
14705	066414	004737	066566			JSR PC,SN	;YES
14706	066420	032737	000002	002472	4\$:	BIT #2,PZCODE	;PRINT A ZONED SRC1 STRING?
14707	066426	001402				BEQ 5\$	
14708	066430	004737	066616			JSR PC,S1N	;YES
14709	066434	032737	000004	002472	5\$:	BIT #4,PZCODE	;PRINT A ZONED SRC2 STRING?
14710	066442	001402				BEQ 6\$	
14711	066444	004737	066706			JSR PC,S2N	;YES
14712	066450	012737	177777	026322	6\$:	MOV #177777,EPAK	
14713	066456	032737	000400	002472		BIT #400,PZCODE	;PRINT A PACKED SOURCE STRING?
14714	066464	001402				BEQ 7\$	
14715	066466	004737	066566			JSR PC,SN	;YES
14716	066472	032737	001000	002472	7\$:	BIT #1000,PZCODE	;PRINT A PACKED SRC1 STRING?
14717	066500	001402				BEQ 10\$	
14718	066502	004737	066616			JSR PC,S1N	;YES
14719	066506	032737	002000	002472	10\$:	BIT #7000,PZCODE	;PRINT A PACKED SRC2 STRING?
14720	066514	001402				BEQ 11\$	
14721	066516	004737	066706			JSR PC,S2N	;YES
14722	066522	032737	000070	002472	11\$:	BIT #70,PZCODE	;PRINT A ZONED DEST. STRING?
14723	066530	001404				BEQ 12\$	
14724	066532	005037	026322			CLR EPAK	
14725	066536	004737	066776			JSR PC,DN	;YES
14726	066542	032737	034000	002472	12\$:	BIT #34000,PZCODE	;PRINT A PACKED DEST. STRING?
14727	066550	001405				BEQ 13\$	
14728	066552	012737	177777	026322		MOV #177777,EPAK	
14729	066560	004737	066776			JSR PC,DN	;YES
14730	066564	000207			13\$:	RTS PC	;EXIT DECIMAL DISPLAY SUBROUTINE.
14731							
14732							
14733	066566				SN:	PRINTB #FORM22	;PRINT 'SRC'
(6)	066566	012746	014077			MOV #FORM22,-(SP)	
(3)	066572	010600				MOV SP,R0	
(4)	066574	004737	065410			JSR PC,FPRINT	
14734	066600	013701	002500			MOV INSRC1,R1	;LOAD R1 WITH STRING LEN
14735	066604	013700	002502			MOV INSRC1+2,R0	;LOAD R0 WITH STRING ADD
14736	066610	004737	067164			JSR PC,DECPRT	;PRINT DECIMAL DIGIT STRING
14737	066614	000207				RTS PC	
14738	066616	013701	050012		S1N:	MOV TINST,R1	
14739	066622	042701	177700			BIC #177700,R1	
14740	066626	020127	000052			CMP R1,#52	;IS INST = CMPN?
14741	066632	001403				BEQ 1\$	;BRANCH IF YES
14742	066634	020127	000072			CMP R1,#72	;IS INST = CMPP?
14743	066640	001006				BNE 2\$	;BRANCH IF NO
14744	066642				1\$:	PRINTB #FORM24	;PRINT 'SRC2'
(6)	066642	012746	014132			MOV #FORM24,-(SP)	
(3)	066646	010600				MOV SP,R0	
(4)	066650	004737	065410			JSR PC,FPRINT	
14745	066654	000405				BR 3\$	
14746	066656				2\$:	PRINTB #FORM23	;PRINT 'SRC1'
(6)	066656	012746	014114			MOV #FORM23,-(SP)	
(3)	066662	010600				MOV SP,R0	
(4)	066664	004737	065410			JSR PC,FPRINT	



14747	066670	013701	002500		3\$:	MOV INSR1,R1	:LOAD R1 WITH STRING LEN
14748	066674	013700	002502			MOV INSR1+2,R0	:LOAD R0 WITH STRING ADD
14749	066700	004737	067164			JSR PC,DECPRT	:PRINT DECIMAL DIGIT STRING
14750	066704	000207				RTS PC	
14751	066706	013701	050012		S2N:	MOV TINST,R1	
14752	066712	042701	177700			BIC #177700,R1	
14753	066716	020127	000052			CMP R1,#52	:IS INST = CMPN?
14754	066722	001403				BEQ 1\$	:BRANCH IF YES
14755	066724	020127	000072			CMP R1,#72	:IS INST = CMPP?
14756	066730	001006				BNE 2\$	:BRANCH IF NO
14757	066732				1\$:	PRINTB #FORM23	:PRINT 'SRC1'
(6)	066732	012746	014114			MOV #FORM23,-(SP)	
(3)	066736	010600				MOV SP,R0	
(4)	066740	004737	065410			JSR PC,FPRINT	
14758	066744	000405				BR 3\$	
14759	066746				2\$:	PRINTB #FORM24	:PRINT 'SRC2'
(6)	066746	012746	014132			MOV #FORM24,-(SP)	
(3)	066752	010600				MOV SP,R0	
(4)	066754	004737	065410			JSR PC,FPRINT	
14760	066760	013701	002504		3\$:	MOV INSR2,R1	:LOAD R1 WITH STRING LENGTH
14761	066764	013700	002506			MOV INSR2+2,R0	:LOAD R0 WITH STRING ADD
14762	066770	004737	067164			JSR PC,DECPRT	:PRINT DECIMAL DIGIT STRING
14763	066774	000207				RTS PC	
14764	066776				DN:	PRINTB #FORM25	:PRINT 'EM RESULT'
(6)	066776	012746	014150			MOV #FORM25,-(SP)	
(3)	067002	010600				MOV SP,R0	
(4)	067004	004737	065410			JSR PC,FPRINT	
14765	067010	032737	020040	002472		BIT #20040,PZCODE	:LOAD R1 WITH STRING LEN
14766	067016	001405				BEQ 1\$	:LOAD R0 WITH STRING ADDRESS
14767	067020	013701	003670			MOV ER0,R1	
14768	067024	013700	003672			MOV ER1,R0	
14769	067030	000415				BR 4\$	
14770	067032	032737	010020	002472	1\$:	BIT #10020,PZCODE	
14771	067040	001405				BEQ 2\$	
14772	067042	013701	003674			MOV ER2,R1	
14773	067046	013700	003676			MOV ER3,R0	
14774	067052	000404				BR 4\$	
14775	067054	013701	003700		2\$:	MOV ER4,R1	
14776	067060	013700	003702			MOV ER5,R0	
14777	067064	004737	067164		4\$:	JSR PC,DECPRT	:PRINT DECIMAL DIGIT STRING
14778	067070					PRINTB #FORM26	:PRINT 'ACT RESULT'
(6)	067070	012746	014173			MOV #FORM26,-(SP)	
(3)	067074	010600				MOV SP,R0	
(4)	067076	004737	065410			JSR PC,FPRINT	
14779	067102	032737	020040	002472		BIT #20040,PZCODE	:LOAD R1 WITH STRING LEN
14780	067110	001405				BEQ 11\$	:LOAD R0 WITH STRING ADDRESS
14781	067112	013701	002234			MOV TTR0,R1	
14782	067116	013700	002236			MOV TTR1,R0	
14783	067122	000415				BR 44\$	
14784	067124	032737	010020	002472	11\$:	BIT #10020,PZCODE	
14785	067132	001405				BEQ 22\$	
14786	067134	013701	002240			MOV TTR2,R1	
14787	067140	013700	002242			MOV TTR3,R0	
14788	067144	000404				BR 44\$	

```

14789 067146 013701 002244      22$:  MOV TTR4,R1
14790 067152 013700 002246      MOV TTR5,R0
14791 067156 004737 067164      44$:  JSR PC,DECPRT      ;PRINT DECIMAL DIGIT STRING
14792 067162 000207
14793
14794
14795      ;SUBROUTINE TO PRINT A DECIMAL STRING OF DIGITS; MSD FIRST ....
14796      ;LEAST SIGNIFICANT DIGIT, SIGN.
14797      ;INPUT:  EPAK=0 FOR ZONED STRING;177777 FOR PACKED
14798      ;R0=STRING ADR
14799      ;R1=STRING LEN
14800
14801      ;NOTE: ROUTINE PRINTS '0 +' FOR ZONED STRINGS OF
14802      ;ZERO LENGTH (EXCEPT SEPARATE TYPE).
14803
14804 067164 012737 177777 002304 DECPRT: MOV #177777,PRTSGN      ;SET PRINTING IN PROGRESS FLAG
14805 067172 010003      MOV R0,R3      ;SAVE R0 IN R3
14806 067174 005037 025112      CLR EODD
14807 067200 032701 000001      BIT #1,R1      ;IS STRING ODD IN LENGTH
14808 067204 001403      BEQ 1$
14809 067206 012737 177777 025112      MOV #177777,EODD      ;SET ODD INDICATOR
14810 067214 010137 025130      1$:  MOV R1,ELSD
14811 067220 110137 002100      MOV B R1,NBLKS      ;DETERMINE # OF BLANK DIGITS TO PRINT
14812 067224 012702 000037      MOV #37,R2
14813 067230 163702 002100      SUB NBLKS,R2
14814 067234 001002      BNE 11$
14815 067236 012702 000001      MOV #1,R2
14816 067242      11$: PRINTB #FORM27      ;PRINT THE BLANKS
      (6) 067242 012746 014216      MOV #FORM27,-(SP)
      (3) 067246 010600      MOV SP,R0
      (4) 067250 004737 065410      JSR PC,FPRINT
14817 067254 005302      DEC R2
14818 067256 001371      BNE 11$
14819 067260 105701      3$:  TSTB R1      ;STRING LENGTH = 0?
14820 067262 001017      BNE 4$      ;BRANCH IF NO
14821 067264 005737 026322      TST EPAK      ;STRING PACKED?
14822 067270 001407      BEQ 5$      ;BRANCH IF NO
14823 067272 005201      INC R1      ;YES - SET LEN = 1
14824 067274 005237 025130      INC ELSD
14825 067300 012737 177777 025112      MOV #177777,EODD
14826 067306 000405      BR 4$
14827 067310 005002      5$:  CLR R2      ;ZONED - ZERO LENGTH
14828 067312 004537 067506      JSR R5,CONVN      ;PRINT 0
14829 067316 067542      DIGTBL
14830 067320 000413      BR 12$      ;EXIT
14831 067322 010300      4$:  MOV R3,R0      ;RESTORE R0
14832 067324 105001      CLRB R1
14833 067326 004737 021372      7$:  JSR PC,ESNK      ;GET NEXT DIGIT
14834 067332 004537 067506      JSR R5,CONVN      ;CONVERT NIBBLE & PRINT HEX DIGIT
14835 067336 067542      DIGTBL
14836 067340 105201      INCB R1
14837 067342 120137 025130      CMPB R1,ELSD
14838 067346 001367      BNE 7$
14839 067350 004737 021372      12$: JSR PC,ESNK      ;CALL ROUTINE TO FIND SIGN

```

```

14840                                     ;SIGN RETURNED IN ERSNEG (0=+ ,/=0=-)
14841                                     ;SIGN BYTE RETURNED IN 'SGNBYT'
14842 067354 005737 025062               TST ERSNEG
14843 067360 001006                       BNE 2$
14844 067362                               PRINTB #FORM34
      (6) 067362 012746 014302           MOV #FORM34,-(SP)
      (3) 067366 010600                   MOV SP,R0
      (4) 067370 004737 065410           JSR PC,FPRINT
14845 067374 000405                       BR 33$
14846 067376                               2$: PRINTB #FORM35
      (6) 067376 012746 014307           MOV #FORM35,-(SP)
      (3) 067402 010600                   MOV SP,R0
      (4) 067404 004737 065410           JSR PC,FPRINT
14847 067410                               33$: PRINTB #FORM32
      (6) 067410 012746 014256           MOV #FORM32,-(SP)
      (3) 067414 010600                   MOV SP,R0
      (4) 067416 004737 065410           JSR PC,FPRINT
14848 067422 013702 025166               MOV SGNBYT,R2
14849 067426 006202                       ASR R2
14850 067430 006202                       ASR R2
14851 067432 006202                       ASR R2
14852 067434 006202                       ASP R2
14853 067436 042702 177760               BIC #177760,R2
14854 067442 004537 067506               JSR R5,CONVN
14855 067446 067542                       DIGTBL
14856 067450 113702 025166               MOV SGNBYT,R2
14857 067454 042702 177760               BIC #177760,R2
14858 067460 004537 067506               JSR R5,CONVN
14859 067464 067542                       DIGTBL
14860 067466                               PRINTB #FORM33
      (6) 067466 012746 014276           MOV #FORM33,-(SP)
      (3) 067472 010600                   MOV SP,R0
      (4) 067474 004737 065410           JSR PC,FPRINT
14861 067500 005037 002304               6$: CLR PRISGN
14862 067504 000207                       RTS PC
14863
14864
14865
14866
14867
14868
14869
14870
14871
14872
14873
14874
14875
14876
14877
14878

```

```

: SUBROUTINE TO CONVERT NIBBLE (IN R2) TO A PRINTABLE CHARACTER
: AND PRINT CHARACTER.
: INPUT PARAMETER FOLLOWS CALL - CONVERSION TABLE ADDRESS

```

```

CONVN:
MOV R0,-(SP)
MOV (R5)+,R0
ADD R2,R0
MOVB (R0),ANIB+2
PRINTB #ANIB
MOV #ANIB,-(SP)
MOV SP,R0
JSR PC,FPRINT
MOV (SP)+,R0
RTS R5

```

```

;SAVE R0
;GET ADDRESS OF CONVERSION TABLE
;INDEX INTO TABLE
;TRANSFER PRINT CHAR FROM TABLE TO ASCII PRINT STREAM.
;PRINT CONVERTED NIBBLE

;RESTORE R0
;EXIT

```

14879	067536	045		ANIB:	.BYTE	045		:X
14880	067537	101			.BYTE	101		:A
14881	067540	000			.BYTE	000		:PRINT CHAR
14882	067541	000			.BYTE	000		:ZERO BYTE
14883								
14884								
14885	067542	060		DIGTBL:	.BYTE	060		:0
14886	067543	061			.BYTE	061		:1
14887	067544	062			.BYTE	062		:2
14888	067545	063			.BYTE	063		:3
14889	067546	064			.BYTE	064		:4
14890	067547	065			.BYTE	065		:5
14891	067550	066			.BYTE	066		:6
14892	067551	067			.BYTE	067		:7
14893	067552	070			.BYTE	070		:8
14894	067553	071			.BYTE	071		:9
14895	067554	101			.BYTE	101		:A
14896	067555	102			.BYTE	102		:B
14897	067556	103			.BYTE	103		:C
14898	067557	104			.BYTE	104		:D
14899	067560	105			.BYTE	105		:E
14900	067561	106			.BYTE	106		:F
14901								
14902								
14903								
14904								
14905	067562	012701	004450	SFCI:	MOV #ASZINS,R1			
14906	067566	005711		1\$:	TST (R1)			:REACHED END OF TABLED ASCII LIST?
14907	067570	001434			BEQ NOMTCH			:BRANCH IF YES
14908	067572	021137	002462		CMP (R1),ACINST			:DO 1ST TWO CHARS MATCH TABLED INST?
14909	067576	001403			BEQ 2\$			:BRANCH IF YES
14910	067600	062701	000010	11\$:	ADD #10,R1			:UPDATE TO NEXT TABLED INST
14911	067604	000770			BR 1\$			:RETURN TO CONTINUE SEARCH
14912	067606	026137	000002	002464	2\$:	CMP 2(R1),ACINST+2		:DO 2ND GROUP OF 2 CHARS MATCH
14913	067614	001401			BEQ 3\$			:BRANCH IF YES
14914	067616	000770			BR 11\$			
14915	067620	026137	000004	002466	3\$:	CMP 4(R1),ACINST+4		:DO 3RD GROUP OF 2 CHARS MATCH
14916	067626	001364			BNE 11\$			:BRANCH IF NO
14917	067630	005737	001760		TST RANDOM			:RANDOM EXERCISE MODE?
14918	067634	001405			BEQ 4\$			:BRANCH IF NO
14919	067636	016100	000006		MOV 6(R1),R0			
14920	067642	011037	072144		MOV (R0),IDUM			:LOAD OCTAL CODING FOR CIS INST INTO DUMMY INPUT TABLE
14921	067646	000403			BR MTCH			
14922	067650	016137	000006	072220	4\$:	MOV 6(R1),INPTBL		:MATCH FOUND - RETURN TO CALL + 4
14923								:SAVE DESIRED INST INPUT TABLE ADDRESS
14924								:IN INPTBL.
14925	067656	062716	000002		MTCH: ADD #2,(SP)			
14926	067662	000207			NOMTCH: RTS PC			:NO MATCH - RETURN TO CALL + 2
14927								
14928								
14929								
14930								
14931								
14932	067664	005037	002462		ACASZ: CLR ACINST			

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

: SUBROUTINE TO ACCEPT ASCII CHARS (6MAX,LESS+CR) FROM TTY.  
: STORES ASCII CHARS 2 PER WORD IN ACINST,ACINST+2, AND ACINST+4.

14933	067670	005037	002464		CLR ACINST+2	
14934	067674	005037	002466		CLR ACINST+4	:CLEAR OUT STORAGE AREA
14935	067700	012701	002462		MOV #ACINST,R1	:SETUP REG POINTER TO STORAGE AREA
14936	067704	105777	111756		TSTB @TKS	:WAIT FOR A CHAR
14937	067710	100375		1\$:	BPL 1\$	
14938	067712	117737	111774	064776	MOVB @TKB,RCHAR	:READ AND SAVE CHAR
14939	067720	032737	000100	064776	BIT #100,RCHAR	:CONVERT LOWER CASE TO UPPER
14940	067726	001403			BEQ 3\$	: CASE LETTERS
14941	067730	042737	000040	064776	BIC #40,RCHAR	
14942	067736	042737	177600	064776	BIC #^C177,RCHAR	:GET RID OF JUNK IF ANY
14943	067744	123727	064776	000015	CMPB RCHAR,#15	:IS CHAR A CR?
14944	067752	001412			BEQ 2\$	:BRANCH IF YES
14945	067754	113721	064776		MOVB RCHAR,(R1)+	:SAVE CHAR
14946	067760	004737	064760		JSR PC,ECHAR	:ECHO 6TH CHAR
14947	067764	022701	002470		CMP #ACINST+6,R1	:6 CHARS ENTERED?
14948	067770	001345			BNE 1\$	:BRANCH IF NO TC LISTEN FOR NEXT CHAR
14949	067772	112737	000015	064776	MOVB #15,RCHAR	:ECHO A CR
14950	070000	004737	064760		JSR PC,ECHAR	:ECHO CR
14951	070004	000207			RTS PC	
14952						
14953						:SUBROUTINE TO RECORD WHICH CONDITION CODE STATES ARE EXERCISED
14954						:FOR EACH INSTRUCTION.
14955						:
14956	070006				RECCC:	
14957	070006	013701	002276		MOV OCTIC,R1	:FORM POINTER INTO TABLE OF COND. CODE USAGE
14958	070012	006301			ASL R1	
14959	070014	062701	004016		ADD #CCREC,R1	
14960	070020	113737	003724	002035	MOVB ECCR,ZCCR+1	
14961	070026	153711	003724		BISB ECCR,(R1)	:LOG CC '1' STATES EXERCISED
14962	070032	005137	002034		COM ZCCR	
14963	070036	042737	170377	002034	BIC #170377,ZCCR	
14964	070044	053711	002034		BIS ZCCR,(R1)	:LOG CC '0' STATES EXERCISED
14965	070050	000207			RTS PC	

```

14967          ;POINTERS TO CIS INSTRUCTION ERROR MESSAGEE HEADER ROUTINE
14968          ;
14969 070052      INEM:
14970 070052 000000      .WORD 0
14971 070054 070142      .WORD YMOV C
14972 070056 070176      .WORD YMOVRC
14973 070060 070232      .WORD YMOVIC
14974 070062 070266      .WORD YLOCC
14975 070064 070322      .WORD YSKPC
14976 070066 070356      .WORD YSCANC
14977 070070 070412      .WORD YSPAN C
14978 070072 070446      .WORD YCMPC
14979 070074 070502      .WORD YMATCHC
14980 070076 070536      .WORD YADDN
14981 070100 070600      .WORD YSUBN
14982 070102 070642      .WORD YCMPN
14983 070104 070704      .WORD YCVTNL
14984 070106 070746      .WORD YCVTPN
14985 070110 071010      .WORD YCVTNP
14986 070112 071052      .WORD YASHN
14987 070114 071114      .WORD YCVTLN
14988 070116 071156      .WORD YADDP
14989 070120 071220      .WORD YSUBP
14990 070122 071262      .WORD YCMP P
14991 070124 071324      .WORD YCVTPL
14992 070126 071366      .WORD YMULP
14993 070130 071430      .WORD YDIVP
14994 070132 071472      .WORD YASHP
14995 070134 071534      .WORD YCVTLP
14996 070136 071576      .WORD YL2D
14997 070140 071636      .WORD YL3D
14998          ;
  
```

15000  
15001  
15002  
15003 070142  
15004 070142  
    (6) 070142 012746 007440  
    (3) 070146 010600  
    (4) 070150 004737 065410  
15005 070154 004737 071676  
15006 070160 000405  
15007 070162  
    (6) 070162 012746 010006  
    (3) 070166 010600  
    (4) 070170 004737 065410  
15008 070174 000207  
15009 070176  
15010 070176  
    (6) 070176 012746 007450  
    (3) 070202 010600  
    (4) 070204 004737 065410  
15011 070210 004737 071676  
15012 070214 000405  
15013 070216  
    (6) 070216 012746 010006  
    (3) 070222 010600  
    (4) 070224 004737 065410  
15014 070230 000207  
15015 070232  
15016 070232  
    (6) 070232 012746 007461  
    (3) 070236 010600  
    (4) 070240 004737 065410  
15017 070244 004737 071676  
15018 070250 000405  
15019 070252  
    (6) 070252 012746 010066  
    (3) 070256 010600  
    (4) 070260 004737 065410  
15020 070264 000207  
15021 070266  
15022 070266  
    (6) 070266 012746 007472  
    (3) 070272 010600  
    (4) 070274 004737 065410  
15023 070300 004737 071676  
15024 070304 000405  
15025 070306  
    (6) 070306 012746 010154  
    (3) 070312 010600  
    (4) 070314 004737 065410  
15026 070320 000207  
15027 070322  
15028 070322  
    (6) 070322 012746 007502

.SBTTL ERROR MESSAGE HEADERS  
:ERROR MESSAGE HEADERS  
:YMOVC:  
PRINTB #AMOV  
MOV #AMOV,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
JSR PC,PRNTIQ ;PRINT TEST #  
BR 1\$ ;CNTL-T RETURN  
PRINTB #FORM1 NORMAL RETURN  
MOV #FORM1,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
RTS PC  
1\$:  
YMOVRC:  
PRINTB #AMOVRC  
MOV #AMOVRC,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
JSR PC,PRNTIQ ;PRINT TEST #  
BR 1\$  
PRINTB #FORM1  
MOV #FORM1,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
RTS PC  
1\$:  
YMOVTC:  
PRINTB #AMOVTC  
MOV #AMOVTC,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
JSR PC,PRNTIQ ;PRINT TEST #  
BR 1\$  
PRINTB #FORM2  
MOV #FORM2,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
RTS PC  
1\$:  
YLOCC:  
PRINTB #ALOCC  
MOV #ALOCC,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
JSR PC,PRNTIQ ;PRINT TEST #  
BR 1\$  
PRINTB #FORM3  
MOV #FORM3,-(SP)  
MOV SP,RO  
JSR PC,FPRINT  
RTS PC  
1\$:  
YSKPC:  
PRINTB #ASKPC  
MOV #ASKPC,-(SP)

(3)	070326	010600		MOV	SP,R0	
(4)	070330	004737	065410	JSR	PC,FPRINT	
15029	070334	004737	071676	JSR	PC,PRNTIQ	;PRINT TEST #
15030	070340	000405		BR	1\$	
15031	070342			PRINTB	#FORM3	
(6)	070342	012746	010154	MOV	#FORM3,-(SP)	
(3)	070346	010600		MOV	SP,R0	
(4)	070350	004737	065410	JSR	PC,FPRINT	
15032	070354	000207		RTS	PC	
15033	070356					
15034	070356			PRINTB	#ASCANC	
(6)	070356	012746	007512	MOV	#ASCANC,-(SP)	
(3)	070362	010600		MOV	SP,R0	
(4)	070364	004737	065410	JSR	PC,FPRINT	
15035	070370	004737	071676	JSR	PC,PRNTIQ	;PRINT TEST #
15036	070374	000405		BR	1\$	
15037	070376			PRINTB	#FORM4	
(6)	070376	012746	010222	MOV	#FORM4,-(SP)	
(3)	070402	010600		MOV	SP,R0	
(4)	070404	004737	065410	JSR	PC,FPRINT	
15038	070410	000207		RTS	PC	
15039	070412					
15040	070412			PRINTB	#ASPANC	
(6)	070412	012746	007523	MOV	#ASPANC,-(SP)	
(3)	070416	010600		MOV	SP,R0	
(4)	070420	004737	065410	JSR	PC,FPRINT	
15041	070424	004737	071676	JSR	PC,PRNTIQ	;PRINT TEST #
15042	070430	000405		BR	1\$	
15043	070432			PRINTB	#FORM4	
(6)	070432	012746	010222	MOV	#FORM4,-(SP)	
(3)	070436	010600		MOV	SP,R0	
(4)	070440	004737	065410	JSR	PC,FPRINT	
15044	070444	000207		RTS	PC	
15045	070446					
15046	070446			PRINTB	#ACMPC	
(6)	070446	012746	007534	MOV	#ACMPC,-(SP)	
(3)	070452	010600		MOV	SP,R0	
(4)	070454	004737	065410	JSR	PC,FPRINT	
15047	070460	004737	071676	JSR	PC,PRNTIQ	;PRINT TEST #
15048	070464	000405		BR	1\$	
15049	070466			PRINTB	#FORM5	
(6)	070466	012746	010276	MOV	#FORM5,-(SP)	
(3)	070472	010600		MOV	SP,R0	
(4)	070474	004737	065410	JSR	PC,FPRINT	
15050	070500	000207		RTS	PC	
15051	070502					
15052	070502			PRINTB	#AMATCHC	
(6)	070502	012746	007544	MOV	#AMATCHC,-(SP)	
(3)	070506	010600		MOV	SP,R0	
(4)	070510	004737	065410	JSR	PC,FPRINT	
15053	070514	004737	071676	JSR	PC,PRNTIQ	;PRINT TEST #
15054	070520	000405		BR	1\$	
15055	070522			PRINTB	#FORM6	
(6)	070522	012746	010362	MOV	#FORM6,-(SP)	



Line	Address	Offset	Code	Label	Comment
(3)	070526	010600	MOV	SP,R0	
(4)	070530	004737	JSR	PC,FPRINT	
15056	070534	000207	RTS	PC	
15057	070536				
15058	070536				
(6)	070536	012746	PRINTB	#AADDN	
(3)	070542	010600	MOV	#AADDN,-(SP)	
(4)	070544	004737	MOV	SP,R0	
15059	070550	004737	JSR	PC,FPRINT	
15060	070554	000410	JSR	PC,PRNTIQ	;PRINT TEST #
15061	070556		BR	1\$	
(6)	070556	012746	PRINTB	#FORM7	
(3)	070562	010600	MOV	#FORM7,-(SP)	
(4)	070564	004737	MOV	SP,R0	
15062	070570	012737	JSR	PC,FPRINT	
15063	070576	000207	MOV	#000016,PZCODE	;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2,DST
15064	070600		RTS	PC	
15065	070600				
(6)	070600	012746	PRINTB	#ASUBN	
(3)	070604	010600	MOV	#ASUBN,-(SP)	
(4)	070606	004737	MOV	SP,R0	
15066	070612	004737	JSR	PC,FPRINT	
15067	070616	000410	JSR	PC,PRNTIQ	;PRINT TEST #
15068	070620		BR	1\$	
(6)	070620	012746	PRINTB	#FORM7	
(3)	070624	010600	MOV	#FORM7,-(SP)	
(4)	070626	004737	MOV	SP,R0	
15069	070632	012737	JSR	PC,FPRINT	
15070	070640	000207	MOV	#000016,PZCODE	;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2,DST.
15071	070642		RTS	PC	
15072	070642				
(6)	070642	012746	PRINTB	#ACMPN	
(5)	070646	010600	MOV	#ACMPN,-(SP)	
(4)	070650	004737	MOV	SP,R0	
15073	070654	004737	JSR	PC,FPRINT	
15074	070660	000410	JSR	PC,PRNTIQ	;PRINT TEST #
15075	070662		BR	1\$	
(6)	070662	012746	PRINTB	#FORM8	
(3)	070666	010600	MOV	#FORM8,-(SP)	
(4)	070670	004737	MOV	SP,R0	
15076	070674	012737	JSR	PC,FPRINT	
15077	070702	000207	MOV	#000006,PZCODE	;SET PRINTCODE TO DISPLAY ZONED SRC1,SRC2.
15078	070704		RTS	PC	
15079	070704				
(6)	070704	012746	PRINTB	#ACVTNL	
(3)	070710	010500	MOV	#ACVTNL,-(SP)	
(4)	070712	004737	MOV	SP,R0	
15080	070716	004737	JSR	PC,FPRINT	
15081	070722	000410	JSR	PC,PRNTIQ	;PRINT TEST #
15082	070724		BR	1\$	
(6)	070724	012746	PRINTB	#FORM9	
(3)	070730	010600	MOV	#FORM9,-(SP)	
(4)	070732	004737	MOV	SP,R0	
15083	070736	012737	JSR	PC,FPRINT	
		000001	MOV	#000001,PZCODE	;SET PRINTCODE TO DISPLAY ZONED SRC.

Address	Offset	PC	Code	Label	Comment
15084	070744	000207	1\$: RTS PC		
15085	070746		YCVTPN:		
15086	070746		PRINTB #ACVTPN		
(6)	070746	012746	MOV #ACVTPN,-(SP)		
(3)	070752	010600	MOV SP,RO		
(4)	070754	004737	JSR PC,FPRINT		
15087	070760	004737	JSR PC,PRNTIQ		;PRINT TEST #
15088	070764	000410	BR 1\$		
15089	070766		PRINTB #FORM10		
(6)	070766	012746	MOV #FORM10,-(SP)		
(3)	070772	010600	MOV SP,RO		
(4)	070774	004737	JSR PC,FPRINT		
15090	071000	012737	MOV #000420,PZCODE		;SET PRINTCODE TO DISPLAY PACKED SRC AND ZONED DST.
15091	071006	000207	1\$: RTS PC		
15092	071010		YCVTNP:		
15093	071010		PRINTB #ACVTNP		
(6)	071010	012746	MOV #ACVTNP,-(SP)		
(3)	071014	010600	MOV SP,RO		
(4)	071016	004737	JSR PC,FPRINT		
15094	071022	004737	JSR PC,PRNTIQ		;PRINT TEST #
15095	071026	000410	BR 1\$		
15096	071030		PRINTB #FORM10		
(6)	071030	012746	MOV #FORM10,-(SP)		
(3)	071034	010600	MOV SP,RO		
(4)	071036	004737	JSR PC,FPRINT		
15097	071042	012737	MOV #010001,PZCODE		;SET PRINTCODE TO DSIPLAY ZONED SRC AND PACKED DST.
15098	071050	000207	1\$: RTS PC		
15099	071052		YASHN:		
15100	071052		PRINTB #AASHN		
(6)	071052	012746	MOV #AASHN,-(SP)		
(3)	071056	010600	MOV SP,RO		
(4)	071060	004737	JSR PC,FPRINT		
15101	071064	004737	JSR PC,PRNTIQ		;PRINT TEST #
15102	071070	000410	BR 1\$		
15103	071072		PRINTB #FORM11		
(6)	071072	012746	MOV #FORM11,-(SP)		
(3)	071076	010600	MOV SP,RO		
(4)	071100	004737	JSR PC,FPRINT		
15104	071104	012737	MOV #000021,PZCODE		;SET PRINTCODE TO DISPLAY ZONED SRC,DST.
15105	071112	000207	1\$: RTS PC		
15106	071114		YCVTLN:		
15107	071114		PRINTB #ACVTLN		
(6)	071114	012746	MOV #ACVTLN,-(SP)		
(3)	071120	010600	MOV SP,RO		
(4)	071122	004737	JSR PC,FPRINT		
15108	071126	004737	JSR PC,PRNTIQ		;PRINT TEST #
15109	071132	000410	BR 1\$		
15110	071134		PRINTB #FGRM12		
(6)	071134	012746	MOV #FORM12,-(SP)		
(3)	071140	010600	MOV SP,RO		
(4)	071142	004737	JSR PC,FPRINT		
15111	071146	012737	MOV #000040,PZCODE		;SET PRINTCODE TO DISPLAY ZONED DST.
15112	071154	000207	1\$: RTS PC		
15113	071156		YADDP:		

15114	071156				PRINTB #AADDP	
(6)	071156	012746	007660		MOV #AADDP,-(SP)	
(3)	071162	010600			MOV SP,RO	
(4)	071164	004737	065410		JSR PC,FPRINT	
15115	071170	004737	071676		JSR PC,PRNTIQ	;PRINT TEST #
15116	071174	000410			BR 1\$	
15117	071176				PRINTB #FORM7	
(6)	071176	012746	010434		MOV #FORM7,-(SP)	
(3)	071202	010600			MOV SP,RO	
(4)	071204	004737	065410		JSR PC,FPRINT	
15118	071210	012737	007000	002472	MOV #007000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST.
15119	071216	000707			RTS PC	
15120	071220				1\$: YSUBP:	
15121	071220				PRINTB #ASUBP	
(6)	071220	012746	007670		MOV #ASUBP,-(SP)	
(3)	071224	010600			MOV SP,RO	
(4)	071226	004737	065410		JSR PC,FPRINT	
15122	071232	004737	071676		JSR PC,PRNTIQ	;PRINT TEST #
15123	071236	000410			BR 1\$	
15124	071240				PRINTB #FORM7	
(6)	071240	012746	010434		MOV #FORM7,-(SP)	
(3)	071244	010600			MOV SP,RO	
(4)	071246	004737	065410		JSR PC,FPRINT	
15125	071252	012737	007000	002472	MOV #007000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST.
15126	071260	000207			RTS PC	
15127	071262				1\$: YCMPP:	
15128	071262				PRINTB #ACMPP	
(6)	071262	012746	007700		MOV #ACMPP,-(SP)	
(3)	071266	010600			MOV SP,RO	
(4)	071270	004737	065410		JSR PC,FPRINT	
15129	071274	004737	071676		JSR PC,PRNTIQ	;PRINT TEST #
15130	071300	000410			BR 1\$	
15131	071302				PRINTB #FORM8	
(6)	071302	012746	010530		MOV #FORM8,-(SP)	
(3)	071306	010600			MOV SP,RO	
(4)	071310	004737	065410		JSR PC,FPRINT	
15132	071314	012737	003000	002472	MOV #003000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC1, AND SRC2.
15133	071322	000207			RTS PC	
15134	071324				1\$: YCVTPL:	
15135	071324				PRINTB #ACVTPL	
(6)	071324	012746	007710		MOV #ACVTPL,-(SP)	
(3)	071330	010600			MOV SP,RO	
(4)	071332	004737	065410		JSR PC,FPRINT	
15136	071336	004737	071676		JSR PC,PRNTIQ	;PRINT TEST #
15137	071342	000410			BR 1\$	
15138	071344				PRINTB #FORM9	
(6)	071344	012746	010606		MOV #FORM9,-(SP)	
(3)	071350	010600			MOV SP,RO	
(4)	071352	004737	065410		JSR PC,FPRINT	
15139	071356	012737	000400	002472	MOV #000400,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC.
15140	071364	000207			RTS PC	
15141	071366				1\$: YMULP:	
15142	071366				PRINTB #AMULP	
(6)	071366	012746	007721		MOV #AMULP,-(SP)	

(3)	071372	010600				MOV	SP,R0	
(4)	071374	004737	065410			JSR	PC,FPRINT	
15143	071400	004737	071676			JSR	PC,PRNTIQ	;PRINT TEST #
15144	071404	000410				BR	1\$	
15145	071406					PRINTB	#FORM7	
(6)	071406	012746	010434			MOV	#FORM7,-(SP)	
(3)	071412	010600				MOV	SP,R0	
(4)	071414	004737	065410			JSR	PC,FPRINT	
15146	071420	012737	007000	002472		MOV	#007000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST.
15147	071426	000207			1\$:	RTS	PC	
15148	071430				YDIVP:			
15149	071430					PRINTB	#ADIVP	
(6)	071430	012746	007731			MOV	#ADIVP,-(SP)	
(3)	071434	010600				MOV	SP,R0	
(4)	071436	004737	065410			JSR	PC,FPRINT	
15150	071442	004737	071676			JSR	PC,PRNTIQ	;PRINT TEST #
15151	071446	000410				BR	1\$	
15152	071450					PRINTB	#FORM7	
(6)	071450	012746	010434			MOV	#FORM7,-(SP)	
(3)	071454	010600				MOV	SP,R0	
(4)	071456	004737	065410			JSR	PC,FPRINT	
15153	071462	012737	007000	002472		MOV	#007000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC1, SRC2, AND DST.
15154	071470	000207			1\$:	RTS	PC	
15155	071472				YASHP:			
15156	071472					PRINTB	#AASHP	
(6)	071472	012746	007741			MOV	#AASHP,-(SP)	
(3)	071476	010600				MOV	SP,R0	
(4)	071500	004737	065410			JSR	PC,FPRINT	
15157	071504	004737	071676			JSR	PC,PRNTIQ	;PRINT TEST #
15158	071510	000410				BR	1\$	
15159	071512					PRINTB	#FORM11	
(6)	071512	012746	010734			MOV	#FORM11,-(SP)	
(3)	071516	010600				MOV	SP,R0	
(4)	071520	004737	065410			JSR	PC,FPRINT	
15160	071524	012737	010400	002472		MOV	#010400,PZCODE	;SET PRINTCODE TO DISPLAY PACKED SRC,DST.
15161	071532	000207			1\$:	RTS	PC	
15162	071534				YCVTLP:			
15163	071534					PRINTB	#ACVTLP	
(6)	071534	012746	007751			MOV	#ACVTLP,-(SP)	
(3)	071540	010600				MOV	SP,R0	
(4)	071542	004737	065410			JSR	PC,FPRINT	
15164	071546	004737	071676			JSR	PC,PRNTIQ	;PRINT TEST #
15165	071552	000410				BR	1\$	
15166	071554					PRINTB	#FORM12	
(6)	071554	012746	011016			MOV	#FORM12,-(SP)	
(3)	071560	010600				MOV	SP,R0	
(4)	071562	004737	065410			JSR	PC,FPRINT	
15167	071566	012737	020000	002472		MOV	#020000,PZCODE	;SET PRINTCODE TO DISPLAY PACKED DST.
15168	071574	000207			1\$:	RTS	PC	
15169	071576				YL2D:			
15170	071576					PRINTB	#AL2D,TINST	
(7)	071576	013746	050012			MOV	TINST,-(SP)	
(6)	071602	012746	007762			MOV	#AL2D,-(SP)	
(3)	071606	010600				MOV	SP,R0	

(4)	071610	004737	065410			JSR PC,FPRINT	
15171	071614	004737	071676			JSR PC,PRNTIQ	
15172	071620	000405				BR 1\$	
15173	071622					PRINTB #FORM41	
(6)	071622	012746	014647			MOV #FORM41,-(SP)	
(3)	071626	010600				MOV SP,R0	
(4)	071630	004737	065410			JSR PC,FPRINT	
15174	071634	000207			1\$:	RTS PC	
15175	071636				YL3D:		
15176	071636					PRINTB #AL3D,TINST	
(7)	071636	013746	050012			MOV TINST,-(SP)	
(6)	071642	012746	007774			MOV #AL3D,-(SP)	
(3)	071646	010600				MOV SP,R0	
(4)	071650	004737	065410			JSR PC,FPRINT	
15177	071654	004737	071676			JSR PC,PRNTIQ	
15178	071660	000405				BR 1\$	
15179	071662					PRINTB #FORM41	
(6)	071662	012746	014647			MOV #FORM41,-(SP)	
(3)	071666	010600				MOV SP,R0	
(4)	071670	004737	065410			JSR PC,FPRINT	
15180	071674	000207			1\$:	RTS PC	
15181					:		
15182					:		
15183					:		
15184					:		
15185	071676	032737	000100	050012		PRNTIQ: BIT #100,TINST	;SUBROUTINE TO PRINT I ON END OF IN-LINE OPCODE
15186							; NEUMONIC AND TEST NUMBER.
15187							; IS OPCODE UNDER TEST AN IN-LINE TYPE?
15188	071704	001405				BEQ 1\$	;BRANCH IF NO
15189	071706					PRINTB #FORM31	;PRINT I
(6)	071706	012746	014252			MOV #FORM31,-(SP)	
(3)	071712	010600				MOV SP,R0	
(4)	071714	004737	065410			JSR PC,FPRINT	
15190	071720				1\$:	PRINTB #FORM30,TOTTCH,TOTTC	;PRINT TEST #
(8)	071720	013746	001420			MOV TOTTC,-(SP)	
(7)	071724	013746	001416			MOV TOTTCH,-(SP)	
(6)	071730	012746	014222			MOV #FORM30,-(SP)	
(3)	071734	010600				MOV SP,R0	
(4)	071736	004737	065410			JSR PC,FPRINT	
15191	071742	005737	001762			TST CTACT	;CNTL-T REQUEST ACTIVE?
15192	071746	001066				BNE 3\$	;BRANCH IF YES
15193	071750					PRINTB #FORM40,INTCT,REGSET	;PRINT INTERRUPT COUNT AND REG SET
(8)	071750	013746	002052			MOV REGSET,-(SP)	
(7)	071754	013746	002546			MOV INTCT,-(SP)	
(6)	071760	012746	014567			MOV #FORM40,-(SP)	
(3)	071764	010600				MOV SP,R0	
(4)	071766	004737	065410			JSR PC,FPRINT	
15194	071772	062716	000002			ADD #2,(SP)	
15195	071776	005737	002160			TST MODE	;FORM ASCII MODE CHARACTER
15196	072002	001004				BNE 4\$	
15197	072004	112737	000113	066122		MOVB #113,TDIG	;KERNEL MODF (K)
15198	072012	000413				BR 10\$	
15199	072014	022737	000001	002160	4\$:	CMP #1,MODE	
15200	072022	001004				BNE 5\$	

15201	072024	112737	000123	066122		MOVB #123,TDIG	:SUPERVISOR MODE (S)
15202	072032	000403				BR 10\$	
15203	072034	112737	000125	066122	5\$:	MOVB #125,TDIG	:USER MODE (U)
15204	072042	104400			10\$:	TYPE	
15205	072044	066122				TDIG	
15206	072046					PRINTB #FRM40A	:PRINT D EN
(6)	072046	012746	014635			MOV #FRM40A,-(SP)	
(3)	072052	010600				MOV SP,R0	
(4)	072054	004737	065410			JSR PC,FPRINT	
15207	072060	005737	002162			TST DEN	:PRINT Y OR N
15208	072064	001004				BNE 6\$	
15209	072066	112737	000116	066122		MOVB #116,TDIG	
15210	072074	000403				BR 11\$	
15211	072076	112737	000131	066122	6\$:	MOVB #131,TDIG	
15212	072104	104400			11\$:	TYPE	
15213	072106	066122				TDIG	
15214	072110					PRINTB #FORM21	:PRINT CRLF
(6)	072110	012746	014074			MOV #FORM21,-(SP)	
(3)	072114	010600				MOV SP,R0	
(4)	072116	004737	065410			JSR PC,FPRINT	
15215	072122	000405				BR 2\$	
15216	072124				3\$:	PRINTB #FORM21	:PRINT CRLF
(6)	072124	012746	014074			MOV #FORM21,-(SP)	
(3)	072130	010600				MOV SP,R0	
(4)	072132	004737	065410			JSR PC,FPRINT	
15217	072136	005037	001762		2\$:	CLR CTACT	:CLEAR CNTL-T FLAG
15218	072142	000207				RTS PC	
15219							
15220							
15221						.EVEN	
15222							
15223							

15226  
 15227  
 15228  
 15229  
 15230 072144 000000  
 15231 072146 000000  
 15232 072150 000000  
 15233 072152 000000  
 15234 072154 000000  
 15235 072156 000000  
 15236 072160 000000  
 15237 072162 000000  
 15238 072164 000000  
 15239 072166 000000  
 15240 072170 000000  
 15241 072172 000000  
 15242 072174 000000  
 15243 072176 000000  
 15244 072200 000000  
 15245 072202 000000  
 15246 072204 000000  
 15247 072206 000000  
 15248 072210 000000  
 15249 072212 000000  
 15250 072214 000000  
 15251 072216 000000  
 15252

```
.SBTTL DUMMY INPUT TABLE (FOR RANDOM MODE)
;DUMMY INPUT TABLE - USED ONLY IN RANDOM EXERCISE MODE
;
;IDUM: .WORD 0 ;INST
; .WORD 0 ;TYPE = 0
; .WORD 0 ;IP1
; .WORD 0 ;IP2
; .WORD 0 ;IP3
; .WORD 0 ;IP4
; .WORD 0 ;IP5
; .WORD 0 ;IP6
; .WORD 0 ;IP7
; .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
;
;IDUME: .WORD 0
```

15254  
15255  
15256  
15257  
15258 072220 000000  
15259

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



15261  
 15262  
 15263  
 15264  
 15265 072222 000032  
 15266 072224 000003  
 15267 072226 107616  
 15268 072230 107626  
 15269 072232 107636  
 15270 072234 107646  
 15271 072236 107656  
 15272 072240 107666  
 15273 072242 107676  
 15274 072244 000000  
 15275 072246 000000  
 15276 072250 000000  
 15277 072252 000000  
 15278 072254 000000  
 15279 072256 000000  
 15280 072260 000000  
 15281 072262 000000  
 15282 072264 000000  
 15283 072266 000000  
 15284 072270 000000  
 15285 072272 000000  
 15286 072274 000000  
 15287  
 15288

```

.SBTTL          L2D TABLES
:ENTRY 0.1 - INSTRUCTION UNDER TEST = L2DR
:IL2D:          .WORD 32          ;INST=L2DR; 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
  
```

15290  
 15291  
 15292  
 15293  
 15294 072276 000033  
 15295 072300 000003  
 15296 072302 107616  
 15297 072304 107626  
 15298 072306 107636  
 15299 072310 107646  
 15300 072312 107656  
 15301 072314 107666  
 15302 072316 107676  
 15303 072320 000000  
 15304 072322 000000  
 15305 072324 000000  
 15306 072326 000000  
 15307 072330 000000  
 15308 072332 000000  
 15309 072334 000000  
 15310 072336 000000  
 15311 072340 000000  
 15312 072342 000000  
 15313 072344 000000  
 15314 072346 000000  
 15315 072350 000000  
 15316  
 15317

```

.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
  
```

15320  
 15321  
 15322  
 15323 072352 000001  
 15324 072354 000003  
 15325 072356 103074  
 15326 072360 103134  
 15327 072362 103114  
 15328 072364 103170  
 15329 072366 104026  
 15330 072370 104056  
 15331 072372 104110  
 15332 072374 104126  
 15333 072376 104142  
 15334 072400 104160  
 15335 072402 104176  
 15336 072404 104212  
 15337 072406 000000  
 15338 072410 102560  
 15339 072412 000000  
 15340 072414 000000  
 15341 072416 000000  
 15342 072420 000000  
 15343 072422 000000  
 15344 072424 000000  
 15345  
 15346  
 15347  
 15348  
 15349  
 15350  
 15351  
 15352  
 15353  
 15354  
 15355  
 15356  
 15357  
 15358  
 15359  
 15360  
 15361  
 15362

```

.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 MOV C)
            .WORD T0          ;IP16 - SPECIAL HANDLING
            .WORD 0           ;IP17
            .WORD C           ;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
:
  
```

15364  
15365  
15366 072426 000001  
15367 072430 000003  
15368 072432 102604  
15369 072434 103152  
15370 072436 102604  
15371 072440 103432  
15372 072442 104026  
15373 072444 104100  
15374 072446 102560  
15375 072450 102560  
15376 072452 102560  
15377 072454 102560  
15378 072456 102560  
15379 072460 104212  
15380 072462 000000  
15381 072464 103030  
15382 072466 000000  
15383 072470 000000  
15384 072472 000000  
15385 072474 000000  
15386 072476 000000  
15387 072500 000000  
15388  
15389  
15390  
15391  
15392  
15393  
15394  
15395  
15396  
15397  
15398  
15399  
15400

:ENTRY 2 - INST UNDER TEST = MOV C

```
MOV 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

15402  
15403  
15404 072502 000001  
15405 072504 000000  
15406 072506 100010  
15407 072510 000201  
15408 072512 000011  
15409 072514 000224  
15410 072516 000377  
15411 072520 104060  
15412 072522 000240  
15413 072524 000000  
15414 072526 000252  
15415 072530 000360  
15416 072532 000000  
15417 072534 000000  
15418 072536 000000  
15419 072540 000003  
15420 072542 000000  
15421 072544 000000  
15422 072546 000000  
15423 072550 000000  
15424 072552 000000  
15425 072554 000000  
15426  
15427  
15428  
15429  
15430  
15431  
15432  
15433  
15434  
15435  
15436  
15437

:ENTRY 3 - INST UNDER TEST = MOV C

:MOV 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 16+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

15439  
15440  
15441  
15442  
15443 072556 000004  
15444 072560 000003  
15445 072562 103074  
15446 072564 103134  
15447 072566 104236  
15448 072570 104056  
15449 072572 104110  
15450 072574 104126  
15451 072576 102560  
15452 072600 000000  
15453 072602 000000  
15454 072604 000000  
15455 072606 000000  
15456 072610 000000  
15457 072612 000000  
15458 072614 000000  
15459 072616 000000  
15460 072620 000000  
15461 072622 000000  
15462 072624 000000  
15463 072626 000000  
15464 072630 000000  
15465  
15466  
15467  
15468  
15469  
15470  
15471  
15472  
15473  
15474  
15475

```
.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  TLIC       ;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
:
```

15477  
15478  
15479  
15480 072632 000004  
15481 072634 000003  
15482 072636 102604  
15483 072640 103152  
15484 072642 104256  
15485 072644 104100  
15486 072646 102560  
15487 072650 102560  
15488 072652 103030  
15489 072654 000000  
15490 072656 000000  
15491 072660 000000  
15492 072662 000000  
15493 072664 000000  
15494 072666 000000  
15495 072670 000000  
15496 072672 000000  
15497 072674 000000  
15498 072676 000000  
15499 072700 000000  
15500 072702 000000  
15501 072704 000000  
15502  
15503  
15504  
15505  
15506  
15507  
15508  
15509  
15510  
15511  
15512

: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
```

15514  
15515  
15516  
15517 072706 000004  
15518 072710 000000  
15519 072712 100011  
15520 072714 000201  
15521 072716 000002  
15522 072720 104060  
15523 072722 000000  
15524 072724 000000  
15525 072726 000003  
15526 072730 000000  
15527 072732 000000  
15528 072734 000000  
15529 072736 000000  
15530 072740 000000  
15531 072742 000000  
15532 072744 000000  
15533 072746 000000  
15534 072750 000000  
15535 072752 000000  
15536 072754 000000  
15537 072756 000000  
15538 072760 000000  
15539  
15540  
15541  
15542  
15543  
15544  
15545  
15546  
15547  
15548  
15549

: 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  
:



15551  
15552  
15553  
15554  
15555 072762 000010  
15556 072764 000003  
15557 072766 103074  
15558 072770 103134  
15559 072772 103114  
15560 072774 104000  
15561 072776 104040  
15562 073000 105172  
15563 073002 102560  
15564 073004 104126  
15565 073006 105172  
15566 073010 104160  
15567 073012 104176  
15568 073014 104212  
15569 073016 102560  
15570 073020 000000  
15571 073022 000000  
15572 073024 000000  
15573 073026 000000  
15574 073030 000000  
15575 073032 000000  
15576 073034 000000  
15577  
15578  
15579  
15580  
15581  
15582  
15583  
15584  
15585  
15586  
15587  
15588  
15589  
15590  
15591

.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  
:

15593  
 15594  
 15595  
 15596 073036 000010  
 15597 073040 000001  
 15598 073042 105262  
 15599 073044 103134  
 15600 073046 105300  
 15601 073050 103170  
 15602 073052 104026  
 15603 073054 104056  
 15604 073056 104110  
 15605 073060 104126  
 15606 073062 104056  
 15607 073064 104160  
 15608 073066 104176  
 15609 073070 104212  
 15610 073072 102560  
 15611 073074 000000  
 15612 073076 000000  
 15613 073100 000000  
 15614 073102 000000  
 15615 073104 000000  
 15616 073106 000000  
 15617 073110 000000  
 15618  
 15619  
 15620  
 15621  
 15622  
 15623  
 15624  
 15625  
 15626  
 15627  
 15628  
 15629  
 15630  
 15631  
 15632  
 15633  
 15634  
 15635  
 15636  
 15637

;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
:
  
```

15639  
15640  
15641  
15642 073112 000010  
15643 073114 000003  
15644 073116 102604  
15645 073120 103152  
15646 073122 102604  
15647 073124 103432  
15648 073126 104026  
15649 073130 104100  
15650 073132 102560  
15651 073134 102560  
15652 073136 104100  
15653 073140 102560  
15654 073142 102560  
15655 073144 104212  
15656 073146 103030  
15657 073150 000000  
15658 073152 000000  
15659 073154 000000  
15660 073156 000000  
15661 073160 000000  
15662 073162 000000  
15663 073164 000000  
15664  
15665  
15666  
15667  
15668  
15669  
15670  
15671  
15672  
15673  
15674  
15675  
15676  
15677  
15678

:ENTRY 9 - INSTRUCTION UNDER TEST = CMPC

```
:CMPC2: .WORD 10          ;INST = CMPC
        .WORD 3          ;TYPE = 1(BIT 0);11/44 OV 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
```

15680  
 15681  
 15682  
 15683  
 15684 073166 000002  
 15685 073170 000003  
 15686 073172 103074  
 15687 073174 103134  
 15688 073176 103114  
 15689 073200 103170  
 15690 073202 104026  
 15691 073204 104056  
 15692 073206 104110  
 15693 073210 104126  
 15694 073212 104142  
 15695 073214 104160  
 15696 073216 104176  
 15697 073220 104212  
 15698 073222 000000  
 15699 073224 102560  
 15700 073226 000000  
 15701 073230 000000  
 15702 073232 000000  
 15703 073234 000000  
 15704 073236 000000  
 15705 073240 000000  
 15706  
 15707  
 15708  
 15709  
 15710  
 15711  
 15712  
 15713  
 15714  
 15715  
 15716  
 15717  
 15718  
 15719  
 15720  
 15721  
 15722  
 15723

```

.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
:
  
```

15725  
 15726  
 15727  
 15728 073242 000002  
 15729 073244 000001  
 15730 073246 102604  
 15731 073250 103152  
 15732 073252 102604  
 15733 073254 103432  
 15734 073256 104026  
 15735 073260 104100  
 15736 073262 102560  
 15737 073264 102560  
 15738 073266 102560  
 15739 073270 102560  
 15740 073272 102560  
 15741 073274 104212  
 15742 073276 000000  
 15743 073300 103030  
 15744 073302 000000  
 15745 073304 000000  
 15746 073306 000000  
 15747 073310 000000  
 15748 073312 000000  
 15749 073314 000000  
 15750  
 15751  
 15752  
 15753  
 15754  
 15755  
 15756  
 15757  
 15758  
 15759  
 15760  
 15761  
 15762

```

:ENTRY 11 - INSTRUCTION UNDER TEST = MOVRC
:MOVRC: .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
  
```

```

: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
:
  
```

15764  
15765  
15766  
15767 073316 000002  
15768 073320 000000  
15769 073322 100100  
15770 073324 100000  
15771 073326 000111  
15772 073330 000100  
15773 073332 000376  
15774 073334 104060  
15775 073336 000240  
15776 073340 000000  
15777 073342 000255  
15778 073344 000366  
15779 073346 000000  
15780 073350 000000  
15781 073352 000000  
15782 073354 000037  
15783 073356 000000  
15784 073360 000000  
15785 073362 000000  
15786 073364 000000  
15787 073366 000000  
15788 073370 000000  
15789  
15790  
15791  
15792  
15793  
15794  
15795  
15796  
15797  
15798  
15799  
15800

ENTRY 12 - INSTRUCTION UNDER TEST = MOVRC

```
MOVRC: .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 16+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

15802  
15803  
15804  
15805 073372 000003  
15806 073374 000001  
15807 073376 103074  
15808 073400 103134  
15809 073402 103114  
15810 073404 103170  
15811 073406 104026  
15812 073410 104056  
15813 073412 104110  
15814 073414 104126  
15815 073416 104142  
15816 073420 104160  
15817 073422 104176  
15818 073424 104212  
15819 073426 105356  
15820 073430 102560  
15821 073432 000000  
15822 073434 000000  
15823 073436 000000  
15824 073440 000000  
15825 073442 000000  
15826 073444 000000  
15827  
15828  
15829  
15830  
15831  
15832  
15833  
15834  
15835  
15836  
15837  
15838  
15839  
15840  
15841  
15842  
15843  
15844  
15845

```

.SBTTL      MOVTC TABLES
:ENTRY 13 - INSTRUCTION UNDER TEST = MOVTC
:
:IMOVTC:   .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

```

15847  
15848  
15849 073446 000003  
15850 073450 000003  
15851 073452 102604  
15852 073454 103152  
15853 073456 102604  
15854 073460 103432  
15855 073462 104026  
15856 073464 104100  
15857 073466 102560  
15858 073470 102560  
15859 073472 102560  
15860 073474 102560  
15861 073476 102560  
15862 073500 104212  
15863 073502 105356  
15864 073504 103030  
15865 073506 000000  
15866 073510 000000  
15867 073512 000000  
15868 073514 000000  
15869 073516 000000  
15870 073520 000000  
15871  
15872  
15873  
15874  
15875  
15876  
15877  
15878  
15879  
15880  
15881  
15882  
15883  
15884  
15885

```
;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
;
```



15887  
15888  
15889 073522 000003  
15890 073524 000000  
15891 073526 100010  
15892 073530 000201  
15893 073532 000011  
15894 073534 000225  
15895 073536 000375  
15896 073540 104060  
15897 073542 000240  
15898 073544 000000  
15899 073546 000254  
15900 073550 000355  
15901 073552 000000  
15902 073554 000000  
15903 073556 001734  
15904 073560 000003  
15905 073562 000000  
15906 073564 000000  
15907 073566 000000  
15908 073570 000000  
15909 073572 000000  
15910 073574 000000  
15911  
15912  
15913  
15914  
15915  
15916  
15917  
15918  
15919  
15920  
15921  
15922  
15923

:ENTRY 15 - INSTRUCTION UNDER TEST = MOVTC

```
:MOV12: .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
```

: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 8 TEST THE MOVTC V-BIT OPERATION  
:TOTAL # OF TESTS = (1 REG. + 1 INLINE)= 2  
:

15925  
15926  
15927  
15928 073576 000005  
15929 073600 000001  
15930 073602 103074  
15931 073604 103134  
15932 073606 105372  
15933 073610 105416  
15934 073612 104110  
15935 073614 104126  
15936 073616 102560  
15937 073620 000000  
15938 073622 000000  
15939 073624 000000  
15940 073626 000000  
15941 073630 000000  
15942 073632 000000  
15943 073634 000000  
15944 073636 000000  
15945 073640 000000  
15946 073642 000000  
15947 073644 000000  
15948 073646 000000  
15949 073650 000000  
15950  
15951  
15952  
15953  
15954  
15955  
15956  
15957  
15958  
15959  
15960

.SBTTL SKPC TABLES  
:ENTRY 16 - INSTRUCTION UNDER TEST = SKPC  
:ISKPC:  
.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.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)  
: 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

15962  
15963  
15964 073652 000005  
15965 073654 000001  
15966 073656 102604  
15967 073660 103152  
15968 073662 105410  
15969 073664 104100  
15970 073666 102560  
15971 073670 102560  
15972 073672 103030  
15973 073674 000000  
15974 073676 000000  
15975 073700 000000  
15976 073702 000000  
15977 073704 000000  
15978 073706 000000  
15979 073710 000000  
15980 073712 000000  
15981 073714 000000  
15982 073716 000000  
15983 073720 000000  
15984 073722 000000  
15985 073724 000000  
15986  
15987  
15988  
15989  
15990  
15991  
15992  
15993  
15994  
15995

:ENTRY 17 - INSTRUCTION UNDER TEST = SKPC

```
ISKPC1: .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
:
```

15997  
15998  
15999  
16000 073726 000005  
16001 073730 000000  
16002 073732 100111  
16003 073734 000203  
16004 073736 000003  
16005 073740 104060  
16006 073742 000000  
16007 073744 000000  
16008 073746 000003  
16009 073750 000000  
16010 073752 000000  
16011 073754 000000  
16012 073756 000000  
16013 073760 000000  
16014 073762 000000  
16015 073764 000000  
16016 073766 000000  
16017 073770 000000  
16018 073772 000000  
16019 073774 000000  
16020 073776 000000  
16021 074000 000000  
16022  
16023  
16024  
16025  
16026  
16027  
16028  
16029  
16030  
16031  
16032

```
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 8 TEST THE SKPC V-BIT OPERATION.  
: TOTAL # OF TESTS = (1 REG. + 1 INLINE) = 2

16034  
16035  
16036  
16037  
16038 074002 000011  
16039 074004 000003  
16040 074006 103074  
16041 074010 103134  
16042 074012 103114  
16043 074014 103510  
16044 074016 104026  
16045 074020 105172  
16046 074022 102560  
16047 074024 104126  
16048 074026 105172  
16049 074030 104160  
16050 074032 104176  
16051 074034 104212  
16052 074036 102560  
16053 074040 000000  
16054 074042 000000  
16055 074044 000000  
16056 074046 000000  
16057 074050 000000  
16058 074052 000000  
16059 074054 000000  
16060  
16061  
16062  
16063  
16064  
16065  
16066  
16067  
16068  
16069  
16070  
16071  
16072  
16073  
16074  
16075  
16076  
16077  
16078

```
.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 =
```

16080  
16081  
16082  
16083  
16084 074056 000011  
16085 074060 000003  
16086 074062 102604  
16087 074064 103152  
16088 074066 102604  
16089 074070 103510  
16090 074072 102560  
16091 074074 104100  
16092 074076 102560  
16093 074100 102560  
16094 074102 104056  
16095 074104 102560  
16096 074106 102560  
16097 074110 104212  
16098 074112 103030  
16099 074114 000000  
16100 074116 000000  
16101 074120 000000  
16102 074122 000000  
16103 074124 000000  
16104 074126 000000  
16105 074130 000000  
16106  
16107  
16108  
16109  
16110  
16111  
16112  
16113  
16114  
16115  
16116  
16117  
16118  
16119  
16120  
16121  
16122  
16123  
16124

ENTRY 21 - INSTRUCTION UNDER TEST = MATCHC  
MATCHC: .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

16126  
16127  
16128  
16129 074132 000011  
16130 074134 000000  
16131 074136 000020  
16132 074140 000100  
16133 074142 100020  
16134 074144 000201  
16135 074146 000000  
16136 074150 104060  
16137 074152 000000  
16138 074154 000000  
16139 074156 104060  
16140 074160 000000  
16141 074162 000000  
16142 074164 000000  
16143 074166 000003  
16144 074170 000000  
16145 074172 000000  
16146 074174 000000  
16147 074176 000000  
16148 074200 000000  
16149 074202 000000  
16150 074204 000000  
16151  
16152  
16153  
16154  
16155  
16156  
16157  
16158  
16159  
16160  
16161  
16162  
16163

: 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

16165  
16166  
16167  
16168  
16169  
16170 074206 000006  
16171 074210 000003  
16172 074212 103074  
16173 074214 103134  
16174 074216 105434  
16175 074220 105450  
16176 074222 105524  
16177 074224 104056  
16178 074226 104160  
16179 074230 104176  
16180 074232 105564  
16181 074234 104110  
16182 074236 104126  
16183 074240 104212  
16184 074242 102560  
16185 074244 000000  
16186 074246 000000  
16187 074250 000000  
16188 074252 000000  
16189 074254 000000  
16190 074256 000000  
16191 074260 000000  
16192  
16193  
16194  
16195  
16196  
16197  
16198  
16199  
16200  
16201  
16202  
16203  
16204  
16205  
16206  
16207  
16208

.SBTTL SCANC TABLES

:ENTRY 23 - INSTRUCTION UNDER TEST = SCANC

:SCAN: .WORD 6 ;INST = SCANC  
          :WORD 3 ;TYPE = 1(BIT 0);11/44 OV 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



16210  
16211  
16212  
16213 074262 000006  
16214 074264 000003  
16215 074266 102604  
16216 074270 103152  
16217 074272 105434  
16218 074274 105470  
16219 074276 105476  
16220 074300 104100  
16221 074302 102560  
16222 074304 102560  
16223 074306 104100  
16224 074310 102560  
16225 074312 102560  
16226 074314 104212  
16227 074316 103030  
16228 074320 000000  
16229 074322 000000  
16230 074324 000000  
16231 074326 000000  
16232 074330 000000  
16233 074332 000000  
16234 074334 000000  
16235  
16236  
16237  
16238  
16239  
16240  
16241  
16242  
16243  
16244  
16245  
16246

:  
:ENTRY 24 - INSTRUCTION UNDER TEST = SCANC  
:

:SCAN1: .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 C ;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  
:

16248  
 16249  
 16250  
 16251 074336 000006  
 16252 074340 000000  
 16253 074342 100040  
 16254 074344 000110  
 16255 074346 000256  
 16256 074350 000377  
 16257 074352 000200  
 16258 074354 104060  
 16259 074356 000000  
 16260 074360 000000  
 16261 074362 105572  
 16262 074364 000000  
 16263 074366 000000  
 16264 074370 000000  
 16265 074372 000003  
 16266 074374 000000  
 16267 074376 000000  
 16268 074400 000000  
 16269 074402 000000  
 16270 074404 000000  
 16271 074406 000000  
 16272 074410 000000  
 16273  
 16274  
 16275  
 16276  
 16277  
 16278  
 16279  
 16280  
 16281  
 16282  
 16283  
 16284  
 16285

```

:ENTRY 25 - INSTRUCTION UNDER TEST = SCANC
:
:ISCAN2: .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

16287  
16288  
16289  
16290  
16291  
16292 074412 000007  
16293 074414 000001  
16294 074416 103074  
16295 074420 103134  
16296 074422 105434  
16297 074424 105450  
16298 074426 105524  
16299 074430 104056  
16300 074432 104160  
16301 074434 104176  
16302 074436 105564  
16303 074440 104110  
16304 074442 104126  
16305 074444 104212  
16306 074446 102560  
16307 074450 000000  
16308 074452 000000  
16309 074454 000000  
16310 074456 000000  
16311 074460 000000  
16312 074462 000000  
16313 074464 000000  
16314  
16315  
16316  
16317  
16318  
16319  
16320  
16321  
16322  
16323  
16324  
16325  
16326  
16327  
16328  
16329  
16330

.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

16332  
16333  
16334  
16335 074466 000007  
16336 074470 000001  
16337 074472 102604  
16338 074474 103152  
16339 074476 105434  
16340 074500 105470  
16341 074502 105476  
16342 074504 104100  
16343 074506 102560  
16344 074510 102560  
16345 074512 104100  
16346 074514 102560  
16347 074516 102560  
16348 074520 104212  
16349 074522 103030  
16350 074524 000000  
16351 074526 000000  
16352 074530 000000  
16353 074532 000000  
16354 074534 000000  
16355 074536 000000  
16356 074540 000000  
16357  
16358  
16359  
16360  
16361  
16362  
16363  
16364  
16365  
16366  
16367  
16368

: 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 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)  
: 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  
:

16370  
16371  
16372  
16373 074542 000007  
16374 074544 000000  
16375 074546 100040  
16376 074550 000110  
16377 074552 000256  
16378 074554 000000  
16379 074556 000200  
16380 074560 104060  
16381 074562 000000  
16382 074564 000000  
16383 074566 105572  
16384 074570 000000  
16385 074572 000000  
16386 074574 000000  
16387 074576 000003  
16388 074600 000000  
16389 074602 000000  
16390 074604 000000  
16391 074606 000000  
16392 074610 000000  
16393 074612 000000  
16394 074614 000000  
16395  
16396  
16397  
16398  
16399  
16400  
16401  
16402  
16403  
16404  
16405  
16406  
16407

: ENTRY 28 - INSTRUCTION UNDER TEST = SPANC  
: ISPAN2: .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 8 TEST THE SPANC N-BIT OPERATION.  
: TOTAL # OF TESTS - (1 REG. + 1 INLINE)=2

16409  
16410  
16411  
16412  
16413  
16414 074616 000016  
16415 074620 000001  
16416 074622 105650  
16417 074624 103134  
16418 074626 105670  
16419 074630 105756  
16420 074632 106034  
16421 074634 104110  
16422 074636 104126  
16423 074640 104142  
16424 074642 104160  
16425 074644 104176  
16426 074646 104212  
16427 074650 102560  
16428 074652 000000  
16429 074654 000000  
16430 074656 000000  
16431 074660 000000  
16432 074662 000000  
16433 074664 000000  
16434 074666 000000  
16435 074670 000000  
16436  
16437  
16438  
16439  
16440  
16441  
16442  
16443  
16444  
16445  
16446  
16447  
16448  
16449  
16450

.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

16452  
16453  
16454 074672 000016  
16455 074674 000001  
16456 074676 102604  
16457 074700 103144  
16458 074702 102604  
16459 074704 105724  
16460 074706 106122  
16461 074710 102560  
16462 074712 102560  
16463 074714 102560  
16464 074716 102560  
16465 074720 102560  
16466 074722 104212  
16467 074724 103030  
16468 074726 000000  
16469 074730 000000  
16470 074732 000000  
16471 074734 000000  
16472 074736 000000  
16473 074740 000000  
16474 074742 000000  
16475 074744 000000

:ENTRY 30 - INSTRUCTION UNDER TEST = CVTPN

ILPZ1: .WORD 16 ;INST=CVTPN  
.WORD 1 ;TYPE = 1  
.WORD T1A ;IP1 - SRC.LEN  
.WORD T2A ;IP2 - SPC.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 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 REG.)64 = 64

16476  
16477  
16478  
16479  
16480  
16481  
16482  
16483  
16484  
16485  
16486  
16487  
16488

16490  
16491  
16492  
16493 074746 000017  
16494 074750 000003  
16495 074752 105650  
16496 074754 103134  
16497 074756 105670  
16498 074760 106312  
16499 074762 106374  
16500 074764 104110  
16501 074766 104126  
16502 074770 104142  
16503 074772 104160  
16504 074774 104176  
16505 074776 104212  
16506 075000 102560  
16507 075002 000000  
16508 075004 000000  
16509 075006 000000  
16510 075010 000000  
16511 075012 000000  
16512 075014 000000  
16513 075016 000000  
16514 075020 000000  
16515  
16516  
16517  
16518  
16519  
16520  
16521  
16522  
16523  
16524  
16525  
16526  
16527  
16528  
16529

```
.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



16531  
16532  
16533 075022 000017  
16534 075024 000001  
16535 075026 102604  
16536 075030 103144  
16537 075032 102604  
16538 075034 106264  
16539 075036 106462  
16540 075040 102560  
16541 075042 102560  
16542 075044 102560  
16543 075046 102560  
16544 075050 102560  
16545 075052 104212  
16546 075054 103030  
16547 075056 000000  
16548 075060 000000  
16549 075062 000000  
16550 075064 000000  
16551 075066 000000  
16552 075070 000000  
16553 075072 000000  
16554 075074 000000

: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 TO ;IP6 - SRC SURR DATA  
          .WORD TO ;IP7 - SRC SURR LEN  
          .WORD TO ;IP10 - DST DATA  
          .WORD TO ;IP11 - DST SURR DATA  
          .WORD TO ;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

16555  
16556  
16557  
16558  
16559  
16560  
16561  
16562  
16563  
16564  
16565  
16566

16568  
16569  
16570 075076 000017  
16571 075100 000001  
16572 075102 102630  
16573 075104 103152  
16574 075106 102630  
16575 075110 106264  
16576 075112 106462  
16577 075114 102560  
16578 075116 102560  
16579 075120 102560  
16580 075122 102560  
16581 075124 102560  
16582 075126 104212  
16583 075130 102560  
16584 075132 000000  
16585 075134 000000  
16586 075136 000000  
16587 075140 000000  
16588 075142 000000  
16589 075144 000000  
16590 075146 000000  
16591 075150 000000

: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 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 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

16592  
16593  
16594  
16595  
16596  
16597  
16598  
16599  
16600  
16601  
16602

16604  
16605  
16606  
16607 075152 000031  
16608 075154 000003  
16609 075156 106540  
16610 075160 106620  
16611 075162 105614  
16612 075164 103134  
16613 075166 104142  
16614 075170 104160  
16615 075172 104176  
16616 075174 102560  
16617 075176 000000  
16618 075200 000000  
16619 075202 000000  
16620 075204 000000  
16621 075206 000000  
16622 075210 000000  
16623 075212 000000  
16624 075214 000000  
16625 075216 000000  
16626 075220 000000  
16627 075222 000000  
16628 075224 000000  
16629  
16630  
16631  
16632  
16633  
16634  
16635  
16636  
16637  
16638  
16639  
16640

```
.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
:
```

16642  
16643  
16644 075226 000031  
16645 075230 000003  
16646 075232 106566  
16647 075234 106604  
16648 075236 102604  
16649 075240 103144  
16650 075242 102560  
16651 075244 102560  
16652 075246 102560  
16653 075250 103030  
16654 075252 000000  
16655 075254 000000  
16656 075256 000000  
16657 075260 000000  
16658 075262 000000  
16659 075264 000000  
16660 075266 000000  
16661 075270 000000  
16662 075272 000000  
16663 075274 000000  
16664 075276 000000  
16665 075300 000000  
16666  
16667  
16668  
16669  
16670  
16671  
16672  
16673  
16674  
16675  
16676

: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 140A ;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  
:

16678  
16679  
16680 075302 000031  
16681 075304 000003  
16682 075306 106572  
16683 075310 106640  
16684 075312 104212  
16685 075314 103134  
16686 075316 102560  
16687 075320 102560  
16688 075322 102560  
16689 075324 102560  
16690 075326 000000  
16691 075330 000000  
16692 075332 000000  
16693 075334 000000  
16694 075336 000000  
16695 075340 000000  
16696 075342 000000  
16697 075344 000000  
16698 075346 000000  
16699 075350 000000  
16700 075352 000000  
16701 075354 000000  
16702  
16703  
16704  
16705  
16706  
16707  
16708  
16709  
16710  
16711  
16712

: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 140B :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  
: .

16714  
16715  
16716  
16717 075356 000021  
16718 075360 000001  
16719 075362 106540  
16720 075364 106620  
16721 075366 105614  
16722 075370 103134  
16723 075372 104142  
16724 075374 104160  
16725 075376 104176  
16726 075400 102560  
16727 075402 000000  
16728 075404 000000  
16729 075406 000000  
16730 075410 000000  
16731 075412 000000  
16732 075414 000000  
16733 075416 000000  
16734 075420 000000  
16735 075422 000000  
16736 075424 000000  
16737 075426 000000  
16738 075430 000000  
16739  
16740  
16741  
16742  
16743  
16744  
16745  
16746  
16747  
16748  
16749

.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  
:

16751  
16752  
16753 075432 000021  
16754 075434 000001  
16755 075436 106610  
16756 075440 106614  
16757 075442 102604  
16758 075444 103144  
16759 075446 102560  
16760 075450 102560  
16761 075452 102560  
16762 075454 103030  
16763 075456 000000  
16764 075460 000000  
16765 075462 000000  
16766 075464 000000  
16767 075466 000000  
16768 075470 000000  
16769 075472 000000  
16770 075474 000000  
16771 075476 000000  
16772 075500 000000  
16773 075502 000000  
16774 075504 000000  
16775  
16776  
16777  
16778  
16779  
16780  
16781  
16782  
16783  
16784  
16785

:ENTRY 36 - INSTRUCTION UNDER TEST = CVTLN  
:ICLZ1: .WORD 21 ;INST=CVTLN  
          .WORD 1 ;TYPE = 1  
          .WORD 140AA ;IP1 - SRC.HIGH (R2)  
          .WORD T41AA ;IP2 - SRC.LOW (R3)  
          .WORD T1A ;IP3 - DST.LEN (R4)  
          .WORD T2A ;IP4 - DST.ADR (R5)  
          .WORD TO ;IP5 - DST DATA  
          .WORD TO ;IP6 - DST SURR DATA  
          .WORD TO ;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  
:

16787  
16788  
16789  
16790 075506 000025  
16791 075510 000003  
16792 075512 105614  
16793 075514 103134  
16794 075516 106034  
16795 075520 104110  
16796 075522 104126  
16797 075524 104026  
16798 075526 102560  
16799 075530 000000  
16800 075532 000000  
16801 075534 000000  
16802 075536 000000  
16803 075540 000000  
16804 075542 000000  
16805 075544 000000  
16806 075546 000000  
16807 075550 000000  
16808 075552 000000  
16809 075554 000000  
16810 075556 000000  
16811 075560 000000  
16812  
16813  
16814  
16815  
16816  
16817  
16818  
16819  
16820  
16821  
16822  
16823

```
.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
:
```



16825  
 16826  
 16827  
 16828  
 16829 075562 000005  
 16830 075564 000003  
 16831 075566 105634  
 16832 075570 103134  
 16833 075572 107522  
 16834 075574 104110  
 16835 075576 104126  
 16836 075600 104026  
 16837 075602 102560  
 16838 075604 000000  
 16839 075606 000000  
 16840 075610 000000  
 16841 075612 000000  
 16842 075614 000000  
 16843 075616 000000  
 16844 075620 000000  
 16845 075622 000000  
 16846 075624 000000  
 16847 075626 000000  
 16848 075630 000000  
 16849 075632 000000  
 16850 075634 000000  
 16851  
 16852  
 16853  
 16854  
 16855  
 16856  
 16857  
 16858  
 16859  
 16860  
 16861  
 16862  
 16863  
 16864  
 16865

: 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

16867  
 16868  
 16869 075636 000025  
 16870 075640 000003  
 16871 075642 102604  
 16872 075644 103174  
 16873 075646 106122  
 16874 075650 102560  
 16875 075652 102560  
 16876 075654 104026  
 16877 075656 103030  
 16878 075660 000000  
 16879 075662 000000  
 16880 075664 000000  
 16881 075666 000000  
 16882 075670 000000  
 16883 075672 000000  
 16884 075674 000000  
 16885 075676 000000  
 16886 075700 000000  
 16887 075702 000000  
 16888 075704 000000  
 16889 075706 000000  
 16890 075710 000000  
 16891  
 16892  
 16893  
 16894  
 16895  
 16896  
 16897  
 16898  
 16899  
 16900

```

: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 11A       :IP1 - SRC.LEN (R0)
        .WORD T2A       :IP2 - SRC.ADR (R1)
        .WORD TP19      :IP3 - SRC DATA
        .WORD T0        :IP4 - SRL SURR DATA
        .WORD T0        :IP5 - SRC SURR LEN
        .WORD T5        :IP6- UNUSED PORTION OF REGISTER 4
        .WORD 'SPA      :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
:

```

16902  
 16903  
 16904 075712 000025  
 16905 075714 000003  
 16906 075716 105634  
 16907 075720 103134  
 16908 075722 106132  
 16909 075724 102560  
 16910 075726 102560  
 16911 075730 104026  
 16912 075732 102560  
 16913 075734 000000  
 16914 075736 000000  
 16915 075740 000000  
 16916 075742 000000  
 16917 075744 000000  
 16918 075746 000000  
 16919 075750 000000  
 16920 075752 000000  
 16921 075754 000000  
 16922 075756 000000  
 16923 075760 000000  
 16924 075762 000000  
 16925 075764 000000  
 16926  
 16927  
 16928  
 16929  
 16930  
 16931  
 16932  
 16933  
 16934  
 16935  
 16936

: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 133A       :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
  
```

16938  
16939  
16940  
16941 075766 000015  
16942 075770 000001  
16943 075772 105614  
16944 075774 103134  
16945 075776 106374  
16946 076000 104110  
16947 076002 104126  
16948 076004 104026  
16949 076006 102560  
16950 076010 000000  
16951 076012 000000  
16952 076014 000000  
16953 076016 000000  
16954 076020 000000  
16955 076022 000000  
16956 076024 000000  
16957 076026 000000  
16958 076030 000000  
16959 076032 000000  
16960 076034 000000  
16961 076036 000000  
16962 076040 000000  
16963  
16964  
16965  
16966  
16967  
16968  
16969  
16970  
16971  
16972  
16973  
16974

```
.SBTTL          CVTNL TABLES
: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

16976  
16977  
16978  
16979  
16980 076042 000015  
16981 076044 000001  
16982 076046 105634  
16983 076050 103134  
16984 076052 107564  
16985 076054 104110  
16986 076056 104126  
16987 076060 104026  
16988 076062 102560  
16989 076064 000000  
16990 076066 000000  
16991 076070 000000  
16992 076072 000000  
16993 076074 000000  
16994 076076 000000  
16995 076100 000000  
16996 076102 000000  
16997 076104 000000  
16998 076106 000000  
16999 076110 000000  
17000 076112 000000  
17001 076114 000000  
17002  
17003  
17004  
17005  
17006  
17007  
17008  
17009  
17010  
17011  
17012  
17013  
17014

: 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

17016  
 17017  
 17018 076116 000015  
 17019 076120 000001  
 17020 076122 102604  
 17021 076124 103144  
 17022 076126 106462  
 17023 076130 102560  
 17024 076132 102560  
 17025 076134 104026  
 17026 076136 103030  
 17027 076140 000000  
 17028 076142 000000  
 17029 076144 000000  
 17030 076146 000000  
 17031 076150 000000  
 17032 076152 000000  
 17033 076154 000000  
 17034 076156 000000  
 17035 076160 000000  
 17036 076162 000000  
 17037 076164 000000  
 17038 076166 000000  
 17039 076170 000000  
 17040  
 17041  
 17042  
 17043  
 17044  
 17045  
 17046  
 17047  
 17048  
 17049

```

:ENTRY 44 - INSTRUCTION UNDER TEST = CVTNL
:ICZL2: .WORD 15          ;INST = CVTNL
        .WORD 1          ;TYPE = 1
        .WORD 11A       ;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
:
  
```

17051  
17052  
17053  
17054 076172 000022  
17055 076174 000003  
17056 076176 102560  
17057 076200 103134  
17058 076202 102560  
17059 076204 104402  
17060 076206 102564  
17061 076210 102560  
17062 076212 105320  
17063 076214 104110  
17064 076216 104126  
17065 076220 105320  
17066 076222 105172  
17067 076224 102560  
17068  
17069 076226 104142  
17070 076230 104160  
17071 076232 104176  
17072 076234 104212  
17073 076236 102560  
17074 076240 000000  
17075 076242 000000  
17076 076244 000000  
17077  
17078  
17079  
17080  
17081  
17082  
17083  
17084  
17085  
17086  
17087  
17088  
17089  
17090  
17091

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

17093  
17094  
17095 076246 000022  
17096 076250 000001  
17097 076252 102560  
17098 076254 103134  
17099 076256 105210  
17100 076260 104402  
17101 076262 102564  
17102 076264 102560  
17103 076266 105320  
17104 076270 104110  
17105 076272 104126  
17106 076274 105116  
17107 076276 105172  
17108 076300 102560  
17109  
17110 076302 104142  
17111 076304 104160  
17112 076306 104176  
17113 076310 104212  
17114 076312 102560  
17115 076314 000000  
17116 076316 000000  
17117 076320 000000  
17118  
17119  
17120  
17121  
17122  
17123  
17124  
17125  
17126  
17127  
17128  
17129  
17130  
17131  
17132  
17133  
17134  
17135

: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
        .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



17137  
17138  
17139 076322 000022  
17140 076324 000001  
17141 076326 105210  
17142 076330 103134  
17143 076332 102560  
17144 076334 104402  
17145 076336 102564  
17146 076340 102560  
17147 076342 105116  
17148 076344 104110  
17149 076346 104126  
17150 076350 105320  
17151 076352 105172  
17152 076354 102560  
17153  
17154 076356 104142  
17155 076360 104160  
17156 076362 104176  
17157 076364 104212  
17158 076366 102560  
17159 076370 000000  
17160 076372 000000  
17161 076374 000000  
17162  
17163  
17164  
17165  
17166  
17167  
17168  
17169  
17170  
17171  
17172  
17173  
17174  
17175  
17176  
17177  
17178  
17179

: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 = 1234567891234567891234000991233; 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
:
```

17181  
17182  
17183 076376 000022  
17184 076400 000001  
17185 076402 105224  
17186 076404 103134  
17187 076406 105242  
17188 076410 104402  
17189 076412 102764  
17190 076414 102560  
17191 076416 105116  
17192 076420 104110  
17193 076422 104126  
17194 076424 105116  
17195 076426 105172  
17196 076430 102560  
17197  
17198 076432 104142  
17199 076434 104160  
17200 076436 104176  
17201 076440 104224  
17202 076442 102560  
17203 076444 000000  
17204 076446 000000  
17205 076450 000000  
17206  
17207  
17208  
17209  
17210  
17211  
17212  
17213  
17214  
17215  
17216  
17217  
17218  
17219  
17220  
17221  
17222  
17223  
17224  
17225  
17226

: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 = (? DATA TYPES + 1 INLINE)384 = 1152
:
```

17228  
17229  
17230 076452 000022  
17231 076454 000003  
17232 076456 102604  
17233 076460 103144  
17234 076462 102604  
17235 076464 104264  
17236 076466 102604  
17237 076470 102560  
17238 076472 106122  
17239 076474 102560  
17240 076476 102560  
17241 076500 106122  
17242 076502 102560  
17243 076504 102560  
17244  
17245 076506 102560  
17246 076510 102560  
17247 076512 102560  
17248 076514 104212  
17249 076516 103030  
17250 076520 000000  
17251 076522 000000  
17252 076524 000000  
17253  
17254  
17255  
17256  
17257  
17258  
17259  
17260  
17261  
17262  
17263  
17264  
17265  
17266

:ENTRY 49 - INSTRUCTION UNDER TEST = ADDP

```

:ADDP4: .WORD 22          ;INST=ADDP
        .WORD 3          ;TYPE = 1(BIT 0):11/44 OV 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
                        ; 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
:

```

17268  
17269  
17270  
17271 076526 000012  
17272 076530 000001  
17273 076532 102560  
17274 076534 103134  
17275 076536 102560  
17276 076540 104754  
17277 076542 102564  
17278 076544 102560  
17279 076546 106652  
17280 076550 104110  
17281 076552 104126  
17282 076554 106652  
17283 076556 105172  
17284 076560 102560  
17285  
17286 076562 104142  
17287 076564 102560  
17288 076566 102560  
17289 076570 106670  
17290 076572 102560  
17291 076574 000000  
17292 076576 000000  
17293 076600 000000  
17294  
17295  
17296  
17297  
17298  
17299  
17300  
17301  
17302  
17303  
17304  
17305  
17306  
17307  
17308

```
.SBTTL      ADDN TABLES
:ENTRY 50 - INSTRUCTION UNDER TEST = ADDN
:ADDN:      .WORD 12      ;INST=ADDN
            .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
:
```

17310  
17311  
17312 076602 000012  
17313 076604 000001  
17314 076606 102560  
17315 076610 103134  
17316 076612 105210  
17317 076614 104754  
17318 076616 102564  
17319 076620 102560  
17320 076622 106652  
17321 076624 104110  
17322 076626 104126  
17323 076630 106374  
17324 076632 105172  
17325 076634 102560  
17326  
17327 076636 104142  
17328 076640 104160  
17329 076642 104176  
17330 076644 104212  
17331 076646 102560  
17332 076650 000000  
17333 076652 000000  
17334 076654 000000  
17335  
17336  
17337  
17338  
17339  
17340  
17341  
17342  
17343  
17344  
17345  
17346  
17347  
17348  
17349  
17350  
17351

: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
.....
```

```

17353
17354
17355 076656 000012
17356 076660 000001
17357 076662 105210
17358 076664 103134
17359 076666 102560
17360 076670 104754
17361 076672 102564
17362 076674 102560
17363 076676 106374
17364 076700 104110
17365 076702 104126
17366 076704 106652
17367 076706 105172
17368 076710 102560
17369
17370 076712 104142
17371 076714 104160
17372 076716 104176
17373 076720 104212
17374 076722 102560
17375 076724 000000
17376 076726 000000
17377 076730 000000
17378
17379
17380
17381
17382
17383
17384
17385
17386
17387
17388
17389
17390
17391
17392
17393
17394

```

```

: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
:

```

17396  
17397  
17398 076732 000012  
17399 076734 000001  
17400 076736 105224  
17401 076740 103134  
17402 076742 105242  
17403 076744 104754  
17404 076746 102764  
17405 076750 102560  
17406 076752 106424  
17407 076754 104110  
17408 076756 104126  
17409 076760 106424  
17410 076762 105172  
17411 076764 102560  
17412  
17413 076766 104142  
17414 076770 102560  
17415 076772 102560  
17416 076774 106704  
17417 076776 102560  
17418 077000 000000  
17419 077002 000000  
17420 077004 000000  
17421  
17422  
17423  
17424  
17425  
17426  
17427  
17428  
17429  
17430  
17431  
17432  
17433  
17434  
17435  
17436  
17437  
17438  
17439  
17440  
17441

```

: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 T0        ;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
:
    
```

17443  
17444  
17445 077006 000012  
17446 077010 000003  
17447 077012 102604  
17448 077014 103144  
17449 077016 102604  
17450 077020 104706  
17451 077022 102604  
17452 077024 102560  
17453 077026 106462  
17454 077030 102560  
17455 077032 102560  
17456 077034 106462  
17457 077036 102560  
17458 077040 102560  
17459  
17460 077042 102560  
17461 077044 102560  
17462 077046 102560  
17463 077050 104212  
17464 077052 103030  
17465 077054 000000  
17466 077056 000000  
17467 077060 000000  
17468  
17469  
17470  
17471  
17472  
17473  
17474  
17475  
17476  
17477  
17478  
17479  
17480  
17481

: 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
         .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
```



17483  
17484  
17485 077062 000012  
17486 077064 000001  
17487 077066 102642  
17488 077070 103152  
17489 077072 102642  
17490 077074 104706  
17491 077076 104212  
17492 077100 102560  
17493 077102 106462  
17494 077104 102560  
17495 077106 102560  
17496 077110 106462  
17497 077112 102560  
17498 077114 102560  
17499  
17500 077116 102560  
17501 077120 102560  
17502 077122 102560  
17503 077124 104212  
17504 077126 102560  
17505 077130 000000  
17506 077132 000000  
17507 077134 000000  
17508  
17509  
17510  
17511  
17512  
17513  
17514  
17515  
17516  
17517  
17518  
17519  
17520  
17521

```
:ENTRY 54A - INSTRUCTION UNDER TEST = ADDN
:ADDNS: .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
:
```

17523  
17524  
17525  
17526 077136 000023  
17527 077140 000001  
17528 077142 102560  
17529 077144 103134  
17530 077146 102560  
17531 077150 104402  
17532 077152 102564  
17533 077154 102560  
17534 077156 105320  
17535 077160 104110  
17536 077162 104126  
17537 077164 105320  
17538 077166 105172  
17539 077170 102560  
17540  
17541 077172 104142  
17542 077174 104160  
17543 077176 104176  
17544 077200 104212  
17545 077202 102560  
17546 077204 000000  
17547 077206 000000  
17548 077210 000000  
17549  
17550  
17551  
17552  
17553  
17554  
17555  
17556  
17557  
17558  
17559  
17560  
17561  
17562  
17563

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

17565  
17566  
17567  
17568  
17569 077212 000023  
17570 077214 000001  
17571 077216 102560  
17572 077220 103134  
17573 077222 105210  
17574 077224 104402  
17575 077226 102564  
17576 077230 102560  
17577 077232 105320  
17578 077234 104110  
17579 077236 104126  
17580 077240 105116  
17581 077242 105172  
17582 077244 102560  
17583  
17584 077246 104142  
17585 077250 104160  
17586 077252 104176  
17587 077254 104212  
17588 077256 102560  
17589 077260 000000  
17590 077262 000000  
17591 077264 000000

: 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

17592  
17593  
17594  
17595  
17596  
17597  
17598  
17599  
17600  
17601  
17602  
17603  
17604  
17605  
17606  
17607  
17608  
17609

17611  
17612  
17613 077266 000023  
17614 077270 000001  
17615 077272 105210  
17616 077274 103134  
17617 077276 102560  
17618 077300 104402  
17619 077302 102564  
17620 077304 102560  
17621 077306 105116  
17622 077310 104110  
17623 077312 104126  
17624 077314 105320  
17625 077316 105172  
17626 077320 102560  
17627  
17628 077322 104142  
17629 077324 104160  
17630 077326 104176  
17631 077330 104212  
17632 077332 102560  
17633 077334 000000  
17634 077336 000000  
17635 077340 000000  
17636  
17637  
17638  
17639  
17640  
17641  
17642  
17643  
17644  
17645  
17646  
17647  
17648  
17649  
17650  
17651  
17652  
17653

;ENTRY 57 - INSTRUCTION UNDER TEST = SUBP

```
ISUBP2: .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 T10       ;IP10 - SRC1 SURR DATA
         .WORD T22       ;IP11 - SRC1 SURR LEN
         .WORD T20       ;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 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
.....
```

17655  
 17656  
 17657 077342 000023  
 17658 077344 000001  
 17659 077346 105224  
 17660 077350 103134  
 17661 077352 105242  
 17662 077354 104402  
 17663 077356 102764  
 17664 077360 102560  
 17665 077362 105116  
 17666 077364 104110  
 17667 077366 104126  
 17668 077370 105116  
 17669 077372 105172  
 17670 077374 102560  
 17671  
 17672 077376 104142  
 17673 077400 104160  
 17674 077402 104176  
 17675 077404 104224  
 17676 077406 102560  
 17677 077410 000000  
 17678 077412 000000  
 17679 077414 000000  
 17680  
 17681  
 17682  
 17683  
 17684  
 17685  
 17686  
 17687  
 17688  
 17689  
 17690  
 17691  
 17692  
 17693  
 17694  
 17695  
 17696  
 17697  
 17698  
 17699  
 17700

```

: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
  
```

17702  
17703  
17704 077416 000023  
17705 077420 000001  
17706 077422 102604  
17707 077424 103144  
17708 077426 102604  
17709 077430 104264  
17710 077432 102604  
17711 077434 102560  
17712 077436 106122  
17713 077440 102560  
17714 077442 102560  
17715 077444 106122  
17716 077446 102560  
17717 077450 102560  
17718  
17719 077452 102560  
17720 077454 102560  
17721 077456 102560  
17722 077460 104212  
17723 077462 103030  
17724 077464 000000  
17725 077466 000000  
17726 077470 000000  
17727  
17728  
17729  
17730  
17731  
17732  
17733  
17734  
17735  
17736  
17737  
17738  
17739  
17740

: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
:

```

17742  
17743  
17744  
17745 077472 000013  
17746 077474 000001  
17747 077476 102560  
17748 077500 103134  
17749 077502 102560  
17750 077504 104754  
17751 077506 102564  
17752 077510 102560  
17753 077512 106652  
17754 077514 104110  
17755 077516 104126  
17756 077520 106652  
17757 077522 105172  
17758 077524 102560  
17759  
17760 077526 104142  
17761 077530 102560  
17762 077532 102560  
17763 077534 106670  
17764 077536 102560  
17765 077540 000000  
17766 077542 000000  
17767 077544 000000  
17768  
17769  
17770  
17771  
17772  
17773  
17774  
17775  
17776  
17777  
17778  
17779  
17780  
17781  
17782

.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  
:

17784  
17785  
17786 077546 000013  
17787 077550 000001  
17788 077552 102560  
17789 077554 103134  
17790 077556 105210  
17791 077560 104754  
17792 077562 102564  
17793 077564 102560  
17794 077566 106652  
17795 077570 104110  
17796 077572 104126  
17797 077574 106374  
17798 077576 105172  
17799 077600 102560  
17800  
17801 077602 104142  
17802 077604 104160  
17803 077606 104176  
17804 077610 104212  
17805 077612 102560  
17806 077614 000000  
17807 077616 000000  
17808 077620 000000  
17809  
17810  
17811  
17812  
17813  
17814  
17815  
17816  
17817  
17818  
17819  
17820  
17821  
17822  
17823  
17824  
17825

: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



17827  
17828  
17829  
17830  
17831 077622 000013  
17832 077624 000001  
17833 077626 105210  
17834 077630 103134  
17835 077632 102560  
17836 077634 104754  
17837 077636 102564  
17838 077640 102560  
17839 077642 106374  
17840 077644 104110  
17841 077646 104126  
17842 077650 106652  
17843 077652 105172  
17844 077654 102560  
17845  
17846 077656 104142  
17847 077660 104160  
17848 077662 104176  
17849 077664 104212  
17850 077666 102560  
17851 077670 000000  
17852 077672 000000  
17853 077674 000000  
17854  
17855  
17856  
17857  
17858  
17859  
17860  
17861  
17862  
17863  
17864  
17865  
17866  
17867  
17868  
17869  
17870

: ENTRY 62 - INSTRUCTION UNDER TEST = SUBN  
: ISUBN2: .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  
:

17872  
17873  
17874 077676 000013  
17875 077700 000001  
17876 077702 105224  
17877 077704 103134  
17878 077706 105242  
17879 077710 104754  
17880 077712 102764  
17881 077714 102560  
17882 077716 106424  
17883 077720 104110  
17884 077722 104126  
17885 077724 106424  
17886 077726 105172  
17887 077730 102560  
17888  
17889 077732 104142  
17890 077734 102560  
17891 077736 102560  
17892 077740 106704  
17893 077742 102560  
17894 077744 000000  
17895 077746 000000  
17896 077750 000000  
17897  
17898  
17899  
17900  
17901  
17902  
17903  
17904  
17905  
17906  
17907  
17908  
17909  
17910  
17911  
17912  
17913  
17914  
17915  
17916  
17917

```

: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 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
:
    
```

17919  
17920  
17921 077752 000013  
17922 077754 000001  
17923 077756 102604  
17924 077760 103144  
17925 077762 102604  
17926 077764 104706  
17927 077766 102604  
17928 077770 102560  
17929 077772 106462  
17930 077774 102560  
17931 077776 102560  
17932 100000 106462  
17933 100002 102560  
17934 100004 102560  
17935  
17936 100006 102560  
17937 100010 102560  
17938 100012 102560  
17939 100014 104212  
17940 100016 103030  
17941 100020 000000  
17942 100022 000000  
17943 100024 000000  
17944  
17945  
17946  
17947  
17948  
17949  
17950  
17951  
17952  
17953  
17954  
17955  
17956  
17957

: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)<sup>12</sup> = 512  
.....

```

17959
17960
17961
17962 100026 000024
17963 100030 000003
17964 100032 102560
17965 100034 103134
17966 100036 102560
17967 100040 106752
17968 100042 104026
17969 100044 105320
17970 100046 105172
17971 100050 102560
17972 100052 105320
17973 100054 104110
17974 100056 104126
17975 100060 106670
17976 100062 102560
17977 100064 000000
17978 100066 000000
17979 100070 000000
17980 100072 000000
17981 100074 000000
17982 100076 000000
17983 100100 000000
17984
17985
17986
17987
17988
17989
17990
17991
17992
17993
17994
17995
17996
17997

```

```

.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 DAT' - 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
;

```

17999  
18000  
18001  
18002 100102 000024  
18003 100104 000001  
18004 100106 102560  
18005 100110 103134  
18006 100112 105210  
18007 100114 106752  
18008 100116 104026  
18009 100120 105320  
18010 100122 105172  
18011 100124 102560  
18012 100126 107072  
18013 100130 104110  
18014 100132 104126  
18015 100134 106670  
18016 100136 102560  
18017 100140 000000  
18018 100142 000000  
18019 100144 000000  
18020 100146 000000  
18021 100150 000000  
18022 100152 000000  
18023 100154 000000  
18024  
18025  
18026  
18027  
18028  
18029  
18030  
18031  
18032  
18033  
18034  
18035  
18036  
18037  
18038  
18039

:ENTRY 66 - INSTRUCTION UNDER TEST = CMPP

```
CMPP1: .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

18041  
18042  
18043  
18044 100156 000024  
18045 100160 000001  
18046 100162 105210  
18047 100164 103134  
18048 100166 102560  
18049 100170 106752  
18050 100172 104026  
18051 100174 107072  
18052 100176 105172  
18053 100200 102560  
18054 100202 105320  
18055 100204 104110  
18056 100206 104126  
18057 100210 106670  
18058 100212 102560  
18059 100214 000000  
18060 100216 000000  
18061 100220 000000  
18062 100222 000000  
18063 100224 000000  
18064 100226 000000  
18065 100230 000000  
18066  
18067  
18068  
18069  
18070  
18071  
18072  
18073  
18074  
18075  
18076  
18077  
18078  
18079  
18080  
18081

: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
```

18083  
18084  
18085  
18086 100232 000024  
18087 100234 000001  
18088 100236 105224  
18089 100240 103134  
18090 100242 105242  
18091 100244 106752  
18092 100246 104026  
18093 100250 107072  
18094 100252 105172  
18095 100254 102560  
18096 100256 107072  
18097 100260 104110  
18098 100262 104126  
18099 100264 106670  
18100 100266 102560  
18101 100270 000000  
18102 100272 000000  
18103 100274 000000  
18104 100276 000000  
18105 100300 000000  
18106 100302 000000  
18107 100304 000000  
18108  
18109  
18110  
18111  
18112  
18113  
18114  
18115  
18116  
18117  
18118  
18119  
18120  
18121  
18122  
18123  
18124  
18125  
18126

: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

18128  
18129  
18130 100306 000024  
18131 100310 000003  
18132 100312 102604  
18133 100314 103144  
18134 100316 102604  
18135 100320 106720  
18136 100322 104026  
18137 100324 106122  
18138 100326 102560  
18139 100330 102560  
18140 100332 106122  
18141 100334 102560  
18142 100336 102560  
18143 100340 104212  
18144 100342 103030  
18145 100344 000000  
18146 100346 000000  
18147 100350 000000  
18148 100352 000000  
18149 100354 000000  
18150 100356 000000  
18151 100360 000000  
18152  
18153  
18154  
18155  
18156  
18157  
18158  
18159  
18160  
18161  
18162  
18163  
18164

;ENTRY 69 - INSTRUCTION UNDER TEST = CMPP  
;CMPP4: .WORD 24 ;INST = CMPP  
; .WGRD 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  
:



18166  
18167  
18168  
18169 100362 000014  
18170 100364 000001  
18171 100366 102560  
18172 100370 103134  
18173 100372 102560  
18174 100374 107154  
18175 100376 104026  
18176 100400 106652  
18177 100402 105172  
18178 100404 102560  
18179 100406 106652  
18180 100410 104110  
18181 100412 104126  
18182 100414 104212  
18183 100416 102560  
18184 100420 000000  
18185 100422 000000  
18186 100424 000000  
18187 100426 000000  
18188 100430 000000  
18189 100432 000000  
18190 100434 000000  
18191  
18192  
18193  
18194  
18195  
18196  
18197  
18198  
18199  
18200  
18201  
18202  
18203  
18204  
18205  
18206

```

.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

```

```

: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) -
:

```

18208  
 18209  
 18210  
 18211 100436 000014  
 18212 100440 000001  
 18213 100442 102560  
 18214 100444 103134  
 18215 100446 105210  
 18216 100450 107154  
 18217 100452 104026  
 18218 100454 106652  
 18219 100456 105172  
 18220 100460 102560  
 18221 100462 107376  
 18222 100464 104110  
 18223 100466 104126  
 18224 100470 104212  
 18225 100472 102560  
 18226 100474 000000  
 18227 100476 000000  
 18228 100500 000000  
 18229 100502 000000  
 18230 100504 000000  
 18231 100506 000000  
 18232 100510 000000  
 18233  
 18234  
 18235  
 18236  
 18237  
 18238  
 18239  
 18240  
 18241  
 18242  
 18243  
 18244  
 18245  
 18246  
 18247  
 18248  
 18249  
 18250  
 18251

: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)
:
  
```

18253  
 18254  
 18255  
 18256 100512 000014  
 18257 100514 000001  
 18258 100516 105210  
 18259 100520 103134  
 18260 100522 102560  
 18261 100524 107154  
 18262 100526 104026  
 18263 100530 107376  
 18264 100532 105172  
 18265 100534 102560  
 18266 100536 106652  
 18267 100540 104110  
 18268 100542 104126  
 18269 100544 104212  
 18270 100546 102560  
 18271 100550 000000  
 18272 100552 000000  
 18273 100554 000000  
 18274 100556 000000  
 18275 100560 000000  
 18276 100562 000000  
 18277 100564 000000  
 18278  
 18279  
 18280  
 18281  
 18282  
 18283  
 18284  
 18285  
 18286  
 18287  
 18288  
 18289  
 18290  
 18291  
 18292  
 18293  
 18294  
 18295  
 18296

;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) =
:

```

18298  
18299  
18300  
18301 100566 000014  
18302 100570 000001  
18303 100572 105224  
18304 100574 103134  
18305 100576 105242  
18306 100600 107154  
18307 100602 104026  
18308 100604 107376  
18309 100606 105172  
18310 100610 102560  
18311 100612 107376  
18312 100614 104110  
18313 100616 104126  
18314 100620 104212  
18315 100622 102560  
18316 100624 000000  
18317 100626 000000  
18318 100630 000000  
18319 100632 000000  
18320 100634 000000  
18321 100636 000000  
18322 100640 000000  
18323  
18324  
18325  
18326  
18327  
18328  
18329  
18330  
18331  
18332  
18333  
18334  
18335  
18336  
18337  
18338  
18339  
18340  
18341  
18342  
18343  
18344

:ENTRY 73 - INSTRUCTION UNDER TEST = CMPN

:CMPN3: .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

18346  
18347  
18348 100642 000014  
18349 100644 000003  
18350 100646 102604  
18351 100650 103144  
18352 100652 102604  
18353 100654 107126  
18354 100656 104026  
18355 100660 106462  
18356 100662 102560  
18357 100664 102560  
18358 100666 106462  
18359 100670 102560  
18360 100672 102560  
18361 100674 104212  
18362 100676 103030  
18363 100700 000000  
18364 100702 000000  
18365 100704 000000  
18366 100706 000000  
18367 100710 000000  
18368 100712 000000  
18369 100714 000000  
18370  
18371  
18372  
18373  
18374  
18375  
18376  
18377  
18378  
18379  
18380  
18381  
18382

:ENTRY 74 - INSTRUCTION UNDER TEST = CMPN

```

:ICMPN4: .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

```

: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 - 20i (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
:

```

18384  
18385  
18386  
18387 100716 000030  
18388 100720 000001  
18389 100722 103034  
18390 100724 103134  
18391 100726 107432  
18392 100730 103054  
18393 100732 105756  
18394 100734 106214  
18395 100736 105172  
18396 100740 102560  
18397 100742 104142  
18398 100744 104160  
18399 100746 104176  
18400 100750 106670  
18401 100752 102560  
18402 100754 000000  
18403 100756 000000  
18404 100760 000000  
18405 100762 000000  
18406 100764 000000  
18407 100766 000000  
18408 100770 000000  
18409  
18410  
18411  
18412  
18413  
18414  
18415  
18416  
18417  
18418  
18419  
18420  
18421  
18422  
18423  
18424  
18425  
18426  
18427  
18428  
18429  
18430  
18431

.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

```

18433
18434
18435 100772 000030
18436 100774 000001
18437 100776 102604
18438 101000 103144
18439 101002 107476
18440 101004 102604
18441 101006 105724
18442 101010 106122
18443 101012 102560
18444 101014 102560
18445 101016 102560
18446 101020 102560
18447 101022 102560
18448 101024 104212
18449 101026 103030
18450 101030 000000
18451 101032 000000
18452 101034 000000
18453 101036 000000
18454 101040 000000
18455 101042 000000
18456 101044 000000
18457
18458
18459
18460
18461
18462
18463
18464
18465
18466
18467
18468
18469
18470

```

;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 - DIGIIS FROM STRING = 1234567891234567891234000891233; SIGN +
: ROUND DIGIT, SHIFT COUNT - 7,-1
:                               7,-2
:                               1, 0
:                               8, 3

```

;TOTAL # OF TESTS = 256

18472  
 18473  
 18474 101046 000030  
 18475 101050 000003  
 18476 101052 102672  
 18477 101054 103144  
 18478 101056 107512  
 18479 101060 102730  
 18480 101062 105724  
 18481 101064 106206  
 18482 101066 102560  
 18483 101070 102560  
 18484 101072 102560  
 18485 101074 102560  
 18486 101076 102560  
 18487 101100 104212  
 18488 101102 102560  
 18489 101104 000000  
 18490 101106 000000  
 18491 101110 000000  
 18492 101112 000000  
 18493 101114 000000  
 18494 101116 000000  
 18495 101120 000000  
 18496  
 18497  
 18498  
 18499  
 18500  
 18501  
 18502  
 18503  
 18504  
 18505  
 18506  
 18507  
 18508

;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 TO        ;IP7 - SRC SURR DATA
        .WORD TO        ;IP10 - SRC SURR LEN
        .WORD TO        ;IP11 - DST DATA
        .WORD TO        ;IP12 - DST SURR DATA
        .WORD TO        ;IP13 - DST SURR LEN
        .WORD T14       ;IP14 - SEP CONST
        .WORD TO        ;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
  
```



18510  
 18511  
 18512  
 18513 101122 000020  
 18514 101124 000001  
 18515 101126 103034  
 18516 101130 103134  
 18517 101132 107432  
 18518 101134 103054  
 18519 101136 106312  
 18520 101140 106472  
 18521 101142 105172  
 18522 101144 102560  
 18523 101146 104142  
 18524 101150 104160  
 18525 101152 104176  
 18526 101154 106670  
 18527 101156 102560  
 18528 101160 000000  
 18529 101162 000000  
 18530 101164 000000  
 18531 101166 000000  
 18532 101170 000000  
 18533 101172 000000  
 18534 101174 000000  
 18535  
 18536  
 18537  
 18538  
 18539  
 18540  
 18541  
 18542  
 18543  
 18544  
 18545  
 18546  
 18547  
 18548  
 18549  
 18550  
 18551  
 18552  
 18553  
 18554  
 18555  
 18556  
 18557

```

.SBTTL          ASHN TABLES
:ENTRY 77 - INSTRUCTION UNDER TEST = ASHN
:ASHN: .WORD 20          ;INST=ASHN
        .WORD 1          ;TYPE = 1
        .WORD T111       ;IP1 - SRC.LIN
        .WORD T2         ;IP2 - SRC., R
        .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
  
```

: 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

18559		
18560		
18561	101176	000020
18562	101200	000003
18563	101202	102604
18564	101204	103144
18565	101206	107462
18566	101210	102604
18567	101212	106264
18568	101214	106462
18569	101216	102560
18570	101220	102560
18571	101222	102560
18572	101224	102560
18573	101226	102560
18574	101230	104212
18575	101232	103030
18576	101234	000000
18577	101236	000000
18578	101240	000000
18579	101242	000000
18580	101244	000000
18581	101246	000000
18582	101250	000000

:ENTRY 78 - INSTRUCTION UNDER TEST = ASHN

```

:ASHN1: .WORD 20          ;INST=ASHN
        .WORD 3          ;TYPE = 1(BIT 0);11/44 OV 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 T219      ;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

18583  
18584  
18585  
18586  
18587  
18588  
18589  
18590  
18591  
18592  
18593

18595  
 18596  
 18597 101252 000020  
 18598 101254 000003  
 18599 101256 102654  
 18600 101260 103144  
 18601 101262 107466  
 18602 101264 102664  
 18603 101266 106264  
 18604 101270 106462  
 18605 101272 102560  
 18606 101274 102560  
 18607 101276 102560  
 18608 101300 102560  
 18609 101302 102560  
 18610 101304 104212  
 18611 101306 102560  
 18612 101310 000000  
 18613 101312 000000  
 18614 101314 000000  
 18615 101316 000000  
 18616 101320 000000  
 18617 101322 000000  
 18618 101324 000000  
 18619  
 18620  
 18621  
 18622  
 18623  
 18624  
 18625  
 18626  
 18627  
 18628  
 18629  
 18630  
 18631

```

:ENTRY 78A - INSTRUCTION UNDER TEST = ASHN
:ASHN2: .WORD 20          ;INST=ASHN
        .WORD 3          ;TYPE = 1(BIT 0);11/44 OV 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
  
```

```

: 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
  
```

18633  
 18634  
 18635  
 18636 101326 000026  
 18637 101330 000003  
 18638 101332 102560  
 18639 101334 103134  
 18640 101336 102560  
 18641 101340 104564  
 18642 101342 102564  
 18643 101344 102560  
 18644 101346 105320  
 18645 101350 104110  
 18646 101352 104126  
 18647 101354 105320  
 18648 101356 105172  
 18649 101360 102560  
 18650  
 18651 101362 104142  
 18652 101364 104160  
 18653 101366 104176  
 18654 101370 104212  
 18655 101372 102560  
 18656 101374 000000  
 18657 101376 000000  
 18658 101400 000000  
 18659  
 18660  
 18661  
 18662  
 18663  
 18664  
 18665  
 18666  
 18667  
 18668  
 18669  
 18670  
 18671  
 18672  
 18673

```

.SBTTL          MULP TABLES
:ENTRY 79 - INSTRUCTION UNDER TEST = MULP
:
:IMULP:         .WORD 26          ;INST=MULP
                .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
:
  
```

18675  
18676  
18677  
18678  
18679 101402 000026  
18680 101404 000001  
18681 101406 102560  
18682 101410 103134  
18683 101412 105210  
18684 101414 104564  
18685 101416 102564  
18686 101420 102560  
18687 101422 105320  
18688 101424 104110  
18689 101426 104126  
18690 101430 105116  
18691 101432 105172  
18692 101434 102560  
18693  
18694 101436 104142  
18695 101440 104160  
18696 101442 104176  
18697 101444 104212  
18698 101446 102560  
18699 101450 000000  
18700 101452 000000  
18701 101454 000000  
18702  
18703  
18704  
18705  
18706  
18707  
18708  
18709  
18710  
18711  
18712  
18713  
18714  
18715  
18716  
18717  
18718  
18719

: ENTRY 80 - INSTRUCTION UNDER TEST = MULP

```

:IMULP1: .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

```

18721  
18722  
18723 101456 000026  
18724 101460 000001  
18725 101462 105210  
18726 101464 103134  
18727 101466 102560  
18728 101470 104564  
18729 101472 102564  
18730 101474 102560  
18731 101476 105116  
18732 101500 104110  
18733 101502 104126  
18734 101504 105320  
18735 101506 105172  
18736 101510 102560  
18737  
18738 101512 104142  
18739 101514 104160  
18740 101516 104176  
18741 101520 104212  
18742 101522 102560  
18743 101524 000000  
18744 101526 000000  
18745 101530 000000  
18746  
18747  
18748  
18749  
18750  
18751  
18752  
18753  
18754  
18755  
18756  
18757  
18758  
18759  
18760  
18761  
18762  
18763

: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 T0      ;IP4 - SRC2.ADR
        .WORD T161    ;IP5 - DST.LEN
        .WORD XT1     ;IP6 - DST.ADR - SPECIFIED BY T16 USED FOR IP4
        .WORD T0      ;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 CF 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

18765  
 18766  
 18767 101532 000026  
 18768 101534 000001  
 18769 101536 105224  
 18770 101540 103134  
 18771 101542 105242  
 18772 101544 104564  
 18773 101546 102764  
 18774 101550 102560  
 18775 101552 105116  
 18776 101554 104110  
 18777 101556 104126  
 18778 101560 105116  
 18779 101562 105172  
 18780 101564 102560  
 18781  
 18782 101566 104142  
 18783 101570 104160  
 18784 101572 104176  
 18785 101574 104212  
 18786 101576 102560  
 18787 101600 000000  
 18788 101602 000000  
 18789 101604 000000  
 18790  
 18791  
 18792  
 18793  
 18794  
 18795  
 18796  
 18797  
 18798  
 18799  
 18800  
 18801  
 18802  
 18803  
 18804  
 18805  
 18806  
 18807  
 18808  
 18809  
 18810

: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
:
  
```

18812  
18813  
18814 101606 000026  
18815 101610 000001  
18816 101612 102604  
18817 101614 103144  
18818 101616 102604  
18819 101620 104264  
18820 101622 102604  
18821 101624 102560  
18822 101626 106122  
18823 101630 102560  
18824 101632 102560  
18825 101634 106122  
18826 101636 102560  
18827 101640 102560  
18828  
18829 101642 102560  
18830 101644 102560  
18831 101646 102560  
18832 101650 104212  
18833 101652 103030  
18834 101654 000000  
18835 101656 000000  
18836 101660 000000  
18837  
18838  
18839  
18840  
18841  
18842  
18843  
18844  
18845  
18846  
18847  
18848  
18849  
18850

:ENTRY 83 - INSTRUCTION UNDER TEST = MULP

:MULP4: .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



18852  
18853  
18854 101662 000026  
18855 101664 000003  
18856 101666 102714  
18857 101670 103134  
18858 101672 102714  
18859 101674 104264  
18860 101676 102742  
18861 101700 102560  
18862 101702 106122  
18863 101704 102560  
18864 101706 102560  
18865 101710 106122  
18866 101712 102560  
18867 101714 102560  
18868  
18869 101716 102560  
18870 101720 102560  
18871 101722 102560  
18872 101724 104212  
18873 101726 103030  
18874 101730 000000  
18875 101732 000000  
18876 101734 000000  
18877  
18878  
18879  
18880  
18881  
18882  
18883  
18884  
18885  
18886  
18887  
18888  
18889  
18890

:ENTRY 83A - INSTRUCTION UNDER TEST = MULP  
:MULP5: .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 SU  
                  ; 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

18892  
 18893  
 18894 101736 000026  
 18895 101740 000003  
 18896 101742 102756  
 18897 101744 103134  
 18898 101746 103004  
 18899 101750 104264  
 18900 101752 102742  
 18901 101754 102560  
 18902 101756 106160  
 18903 101760 102560  
 18904 101762 102560  
 18905 101764 106172  
 18906 101766 102560  
 18907 101770 102560  
 18908  
 18909 101772 102560  
 18910 101774 102560  
 18911 101776 102560  
 18912 102000 104212  
 18913 102002 103030  
 18914 102004 000000  
 18915 102006 000000  
 18916 102010 000000  
 18917  
 18918  
 18919  
 18920  
 18921  
 18922  
 18923  
 18924  
 18925  
 18926  
 18927  
 18928  
 18929  
 18930

```

:ENTRY 83B - INSTRUCTION UNDER TEST = MULP
:MULP6: .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
  
```

18932  
 18933  
 18934  
 18935 102012 000026  
 18936 102014 000001  
 18937 102016 102750  
 18938 102020 103134  
 18939 102022 102750  
 18940 102024 104342  
 18941 102026 104176  
 18942 102030 102560  
 18943 102032 106150  
 18944 102034 102560  
 18945 102036 102560  
 18946 102040 106150  
 18947 102042 102560  
 18948 102044 102560  
 18949  
 18950 102046 102560  
 18951 102050 102560  
 18952 102052 102560  
 18953 102054 102560  
 18954 102056 102560  
 18955 102060 000000  
 18956 102062 000000  
 18957 102064 000000  
 18958  
 18959  
 18960  
 18961  
 18962  
 18963  
 18964  
 18965  
 18966  
 18967  
 18968  
 18969  
 18970  
 18971

: 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 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 - 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
  
```

18973  
 18974  
 18975  
 18976 102066 000027  
 18977 102070 000003  
 18978 102072 102560  
 18979 102074 103134  
 18980 102076 102560  
 18981 102100 104564  
 18982 102102 102564  
 18983 102104 102560  
 18984 102106 105320  
 18985 102110 104110  
 18986 102112 104126  
 18987 102114 105320  
 18988 102116 105172  
 18989 102120 102560  
 18990  
 18991 102122 104142  
 18992 102124 104160  
 18993 102126 104176  
 18994 102130 104212  
 18995 102132 102560  
 18996 102134 000000  
 18997 102136 000000  
 18998 102140 000000  
 18999  
 19000  
 19001  
 19002  
 19003  
 19004  
 19005  
 19006  
 19007  
 19008  
 19009  
 19010  
 19011  
 19012  
 19013

```

.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
:
  
```

19015  
 19016  
 19017 102142 000027  
 19018 102144 000001  
 19019 102146 102560  
 19020 102150 103134  
 19021 102152 105210  
 19022 102154 104564  
 19023 102156 102564  
 19024 102160 102560  
 19025 102162 105320  
 19026 102164 104110  
 19027 102166 104126  
 19028 102170 105116  
 19029 102172 105172  
 19030 102174 102560  
 19031  
 19032 102176 104142  
 19033 102200 104160  
 19034 102202 104176  
 19035 102204 104212  
 19036 102206 102560  
 19037 102210 000000  
 19038 102212 000000  
 19039 102214 000000  
 19040  
 19041  
 19042  
 19043  
 19044  
 19045  
 19046  
 19047  
 19048  
 19049  
 19050  
 19051  
 19052  
 19053  
 19054  
 19055  
 19056  
 19057

: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
:
  
```

19059  
 19060  
 19061  
 19062 102216 000027  
 19063 102220 000001  
 19064 102222 105210  
 19065 102224 103134  
 19066 102226 102560  
 19067 102230 104564  
 19068 102232 102564  
 19069 102234 102560  
 19070 102236 105116  
 19071 102240 104110  
 19072 102242 104126  
 19073 102244 105320  
 19074 102246 105172  
 19075 102250 102560  
 19076  
 19077 102252 104142  
 19078 102254 104160  
 19079 102256 104176  
 19080 102260 104212  
 19081 102262 102560  
 19082 102264 000000  
 19083 102266 000000  
 19084 102270 000000  
 19085  
 19086  
 19087  
 19088  
 19089  
 19090  
 19091  
 19092  
 19093  
 19094  
 19095  
 19096  
 19097  
 19098  
 19099  
 19100  
 19101  
 19102

: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 INLINE)48 = 144
:
  
```

19104  
19105  
19106 102272 000027  
19107 102274 000001  
19108 102276 105224  
19109 102300 103134  
19110 102302 105242  
19111 102304 104564  
19112 102306 102764  
19113 102310 102560  
19114 102312 105116  
19115 102314 104110  
19116 102316 104126  
19117 102320 105116  
19118 102322 105172  
19119 102324 102560  
19120  
19121 102326 104142  
19122 102330 104160  
19123 102332 104176  
19124 102334 104212  
19125 102336 102560  
19126 102340 000000  
19127 102342 000000  
19128 102344 000000  
19129  
19130  
19131  
19132  
19133  
19134  
19135  
19136  
19137  
19138  
19139  
19140  
19141  
19142  
19143  
19144  
19145  
19146  
19147  
19148  
19149

: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
```

19151  
 19152  
 19153 102346 000027  
 19154 102350 000001  
 19155 102352 102604  
 19156 102354 103144  
 19157 102356 102604  
 19158 102360 104264  
 19159 102362 102604  
 19160 102364 102560  
 19161 102366 106122  
 19162 102370 102560  
 19163 102372 102560  
 19164 102374 106122  
 19165 102376 102560  
 19166 102400 102560  
 19167  
 19168 102402 102560  
 19169 102404 102560  
 19170 102406 102560  
 19171 102410 104212  
 19172 102412 103030  
 19173 102414 000000  
 19174 102416 000000  
 19175 102420 000000  
 19176  
 19177  
 19178  
 19179  
 19180  
 19181  
 19182  
 19183  
 19184  
 19185  
 19186  
 19187  
 19188  
 19189

```

:ENTRY 88 - INSTRUCTION UNDER TEST = DIVP
:DIVP4: .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
:
  
```



19191  
19192  
19193 102422 000027  
19194 102424 000003  
19195 102426 102706  
19196 102430 103134  
19197 102432 102706  
19198 102434 104264  
19199 102436 102706  
19200 102440 102560  
19201 102442 105156  
19202 102444 102560  
19203 102446 102560  
19204 102450 105156  
19205 102452 102560  
19206 102454 102560  
19207  
19208 102456 102560  
19209 102460 102560  
19210 102462 102560  
19211 102464 104212  
19212 102466 103030  
19213 102470 000000  
19214 102472 000000  
19215 102474 000000  
19216  
19217  
19218  
19219  
19220  
19221  
19222  
19223  
19224  
19225  
19226  
19227  
19228  
19229

: 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  
:

19231  
 19232  
 19233 102476 000027  
 19234 102500 000003  
 19235 102502 103014  
 19236 102504 103134  
 19237 102506 103022  
 19238 102510 104264  
 19239 102512 103014  
 19240 102514 102560  
 19241 102516 106200  
 19242 102520 102560  
 19243 102522 102560  
 19244 102524 106200  
 19245 102526 102560  
 19246 102530 102560  
 19247  
 19248 102532 102560  
 19249 102534 102560  
 19250 102536 102560  
 19251 102540 104212  
 19252 102542 103030  
 19253 102544 000000  
 19254 102546 000000  
 19255 102550 000000  
 19256  
 19257  
 19258  
 19259  
 19260  
 19261  
 19262  
 19263  
 19264  
 19265  
 19266  
 19267  
 19268  
 19269

:ENTRY 888 - 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
        ; 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
:
  
```

19273			:ENTRY 89 -
19274			:
19275	102552	000000	.WORD 0
19276	102554	000000	.WORD 0
19277	102556	000000	.WORD 0

19279  
19280  
19281  
19282  
19284 102560  
19285  
19286  
19287 102560 000401  
19288 102562 000000  
19289  
19290 102564  
19291 102564 000403  
19292 102566 000000  
19293 102570 000001  
19294 102572 000005  
19295 102574 000000  
19296 102576 000000  
19297 102600 000000  
19298 102602 000000  
19299  
19300 102604 000410  
19301 102606 000000  
19302 102610 000001  
19303 102612 000002  
19304 102614 000003  
19305 102616 000004  
19306 102620 000005  
19307 102622 000011  
19308 102624 000020  
19309 102626 000000  
19310  
19311 102630 000403  
19312 102632 000001  
19313 102634 000002  
19314 102636 000003  
19315 102640 000000  
19316  
19317 102642 000403  
19318 102644 000002  
19319 102646 000004  
19320 102650 000006  
19321 102652 000000  
19322  
19323 102654 000402  
19324 102656 000001  
19325 102660 000002  
19326 102662 000000  
19327  
19328 102664 000401  
19329 102666 000002  
19330 102670 000000  
19331  
19332 102672 000404  
19333 102674 000020

.SBTTL PARAMETER TABLES (LENGTHS,ADDRESSES,ETC)

:PARAMETER TABLES

T0:

:DUMMY TABLE - USED WHEN AN INPUT PARAMETER  
: IS ALREADY SPECIFIED BY A PRECEDING INPUT PARAMETER T  
:ALSO PROVIDES A SINGLE ENTRY = 0 TABLE

.WORD 401  
.WORD 0

XT1:

:FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES

.WORD 403  
.WORD 0  
.WORD 1  
.WORD 5  
.WORD 0  
.WORD 0  
.WORD 0  
.WORD 0

T1A:

.WORD 410  
.WORD 0  
.WORD 1  
.WORD 2  
.WORD 3  
.WORD 4  
.WORD 5  
.WORD 11  
.WORD 20  
.WORD 0

T1B:

.WORD 403  
.WORD 1  
.WORD 2  
.WORD 3  
.WORD 0

T1C:

.WORD 403  
.WORD 2  
.WORD 4  
.WORD 6  
.WORD 0

T1D:

.WORD 402  
.WORD 1  
.WORD 2  
.WORD 0

T1E:

.WORD 401  
.WORD 2  
.WORD 0

T1F:

.WORD 404  
.WORD 20

19334	102676	000016		.WORD 16
19335	102700	000017		.WORD 17
19336	102702	000003		.WORD 3
19337	102704	000000		.WORD 0
19338				
19339	102706	000401	T1G:	.WORD 401
19340	102710	000011		.WORD 11
19341	102712	000000		.WORD 0
19342				
19343	102714	000404	T1H:	.WORD 404
19344	102716	000012		.WORD 12
19345	102720	000013		.WORD 13
19346	102722	000014		.WORD 14
19347	102724	000015		.WORD 15
19348	102726	000000		.WORD 0
19349				
19350	102730	000404	T1K:	.WORD 404
19351	102732	000020		.WORD 20
19352	102734	000016		.WORD 16
19353	102736	000017		.WORD 17
19354	102740	000001		.WORD 1
19355				
19356	102742	000401	T1L:	.WORD 401
19357	102744	000035		.WORD 35
19358	102746	000000		.WORD 0
19359				
19360	102750	000401	T1M:	.WORD 401
19361	102752	000017		.WORD 17
19362	102754	000000		.WORD 0
19363				
19364	102756	000401	T1N:	.WORD 401
19365	102760	000003		.WORD 3
19366	102762	000000		.WORD 0
19367				
19368	102764	000403	T1P:	.WORD 403
19369	102766	000000		.WORD 0
19370	102770	000001		.WORD 1
19371	102772	000037		.WORD 37
19372	102774	000000		.WORD 0
19373	102776	000000		.WORD 0
19374	103000	000000		.WORD 0
19375	103002	000000		.WORD 0
19376				
19377	103004	000402	T1Q:	.WORD 402
19378	103006	000000		.WORD 0
19379	103010	000003		.WORD 3
19380	103012	000000		.WORD 0
19381				
19382	103014	000401	T1R:	.WORD 401
19383	103016	000011		.WORD 11
19384	103020	000000		.WORD 0
19385				
19386	103022	000401	T1S:	.WORD 401
19387	103024	000021		.WORD 21

19388	103026	000000		.WORD 0	
19389					
19390	103030		TSPA:		
19391	103030	000401		.WORD 401	
19392	103032	140000		.WORD 140000	
19393	103034		T111:		
19394	103034	000403		.WORD 403	;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES
19395	103036	000000		.WORD 0	
19396	103040	000001		.WORD 1	
19397	103042	000037		.WORD 37	
19398	103044	000000		.WORD 0	
19399	103046	000000		.WORD 0	
19400	103050	000000		.WORD 0	
19401	103052	000000		.WORD 0	
19402					
19403	103054		T112:		
19404	103054	000403		.WORD 403	;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES
19405	103056	000000		.WORD 0	
19406	103060	000001		.WORD 1	
19407	103062	000037		.WORD 37	
19408	103064	000000		.WORD 0	
19409	103066	000000		.WORD 0	
19410	103070	000000		.WORD 0	
19411	103072	000000		.WORD 0	
19412					
19413	103074		TL1C:		
19414	103074	000403		.WORD 403	;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES
19415	103076	000000		.WORD 0	
19416	103100	000001		.WORD 1	
19417	103102	000300		.WORD 300	
19418	103104	000000		.WORD 0	
19419	103106	000000		.WORD 0	
19420	103110	000000		.WORD 0	
19421	103112	000000		.WORD 0	
19422					
19423	103114		TL2C:		
19424	103114	000403		.WORD 403	;FIXED LENGTH ENTRIES;1 WORD/ENTRY;3 ENTRIES
19425	103116	000000		.WORD 0	
19426	103120	000001		.WORD 1	
19427	103122	000005		.WORD 5	
19428	103124	000000		.WORD 0	
19429	103126	000000		.WORD 0	
19430	103130	000000		.WORD 0	
19431	103132	000000		.WORD 0	
19432					
19433	103134		T2:		
19434	103134	000401		.WORD 401	
19435	103136	000200		.WORD 200	
19436	103140	000000		.WORD 0	
19437	103142	000000		.WORD 0	
19438					
19439	103144		T2A:		
19440	103144	000401		.WORD 401	
19441	103146	000201		.WORD 201	

19442	103150	000000		.WORD 0	
19443					
19444	103152	000402	T2AA:	.WORD 402	
19445	103154	000200		.WORD 200	
19446	103156	000201		.WORD 201	
19447	103160	000000		.WORD 0	
19448					
19449	103162	000401	T2B:	.WORD 401	
19450	103164	000230		.WORD 230	
19451	103166	000000		.WORD 0	
19452					
19453	103170		T4:		
19454	103170	040007		.WORD 40007	:VARIABLE LENGTH ENTRIES:7 ENTRIES
19455	103172	013701	003632	MOV TR1,R1	:ENTRY 1; NO OVERLAP; TR3=TR1+TR0+SEP. CONSTANT
19456	103176	063701	003630	ADD TR0,R1	
19457	103202	067701	076410	ADD @PTP14,R1	
19458	103206	010137	003636	MOV R1,TR3	
19459	103212	000207		RTS PC	
19460	103214	000000		0	
19461	103216	013701	003632	MOV TR1,R1	:ENTRY 2; NO OVERLAP; TR3=TR1-TR2-SEP CONSTANT
19462	103222	163701	003634	SUB TR2,R1	
19463	103226	167701	076364	SUB @PTP14,R1	
19464	103232	010137	003636	MOV R1,TR3	
19465	103236	000207		RTS PC	
19466	103240	000000		0	
19467	103242	005737	003630	TST TR0	
19468	103246	001465		BEQ 1\$	
19469	103250	013701	003632	MOV TR1,R1	:ENTRY 3; ADJACENT; TR3=TR1+TR0
19470	103254	063701	003630	ADD TR0,R1	:REDUNDANT WITH ENTRY 7 IF TR0=0
19471	103260	010137	003636	MOV R1,TR3	
19472	103264	000207		RTS PC	
19473	103266	000000		0	
19474	103270	005737	003634	TST TR2	
19475	103274	001452		BEQ 1\$	
19476	103276	013701	003632	MOV TR1,R1	:ENTRY 4; ADJACENT; TR3=TR1-TR2
19477	103302	163701	003634	SUB TR2,R1	:REDUNDANT WITH ENTRY 7 IF TR2=0
19478	103306	010137	003636	MOV R1,TR3	
19479	103312	000207		RTS PC	
19480	103314	000000		0	
19481	103316	013701	003630	MOV TR0,R1	:ENTRY 5; PARTIAL OVERLAP; TR3=TR1+TR0-(TR2/2)
19482	103322	013702	003634	MOV TR2,R2	:REDUNDANT WITH ENTRY 7 IF TR0-(TR2/2)=0
19483	103326	006202		ASR R2	
19484	103330	160201		SUB R2,R1	
19485	103332	005701		TST R1	
19486	103334	001432		BEQ 1\$	
19487	103336	063701	003632	ADD TR1,R1	
19488	103342	010137	003636	MOV R1,TR3	
19489	103346	000207		RTS PC	
19490	103350	000000		0	
19491	103352	005737	003630	TST TR0	
19492	103356	001421		BEQ 1\$	
19493	103360	013701	003632	MOV TR1,R1	:ENTRY 6; PARTIAL OVERLAP; TR3=TR1-(TR2/2)
19494	103364	013702	003634	MOV TR2,R2	:REDUNDANT WITH ENTRY 5 IF TR0=0
19495	103370	006202		ASR R2	:REDUNDANT WITH ENTRY 7 IF TR2/2 =0

19496	103372	005702		TST R2	
19497	103374	001412		BEQ 1\$	
19498	103376	160201		SUB R2,R1	
19499	103400	010137	003636	MOV R1,TR3	
19500	103404	000207		RTS PC	
19501	103406	000000		0	
19502	103410	013737	003632 003636	MOV TR1,TR3	;ENTRY 7; COMPLETE OVERLAP; TR3=TR1
19503	103416	000207		RTS PC	
19504	103420	000000		0	
19505	103422	062706	000002	1\$: ADD #2,SP	;FIXUP STACK POINTER
19506	103426	000137	042334	JMP REDNTC	;SKIP ENTRY TEST CONDITION - REDUNDANT
19507					
19508	103432			T4A:	
19509	103432	040002		.WORD 40002	;VARIABLE LENGTH ENTRIES; 2 ENTRIES
19510	103434	013701	003630	MOV TR0,R1	;ENTRY 1; PARTIAL OVERLAP; TR3=TR1+TR0-(TR2/2)
19511	103440	013702	003634	MOV TR2,R2	
19512	103444	006202		ASR R2	
19513	103446	160201		SUB R2,R1	
19514	103450	063701	003632	ADD TR1,R1	
19515	103454	010137	003636	MOV R1,TR3	
19516	103460	000207		RTS PC	
19517	103462	000000		0	
19518	103464	013701	003632	MOV TR1,R1	;ENTRY 2; PARTIAL OVERLAP; TR3=TR1-(TR2/2)
19519	103470	013702	003634	MOV TR2,R2	
19520	103474	006202		ASR R2	
19521	103476	160201		SUB R2,R1	
19522	103500	010137	003636	MOV R1,TR3	
19523	103504	000207		RTS PC	
19524	103506	000000		0	
19525	103510			T4I:	
19526	103510	040007		.WORD 40007	;VARIABLE LENGTH ENTRIES; 7 ENTRIES
19527	103512	013701	003636	MOV TR3,R1	;ENTRY 1; NO OVERLAP; TR1=TR2+TR3 +SEP. CONSTANT
19528	103516	063701	003634	ADD TR2,R1	
19529	103522	067701	076070	ADD @PTP14,R1	
19530	103526	010137	003632	MOV R1,TR1	
19531	103532	000207		RTS PC	
19532	103534	000000		0	
19533	103536	013701	003636	MOV TR3,R1	;ENTRY 2; NO OVERLAP; TR1=TR3-TR0-SEP CONSTANT
19534	103542	163701	003630	SUB TR0,R1	
19535	103546	167701	076044	SUB @PTP14,R1	
19536	103552	010137	003632	MOV R1,TR1	
19537	103556	000207		RTS PC	
19538	103560	000000		0	
19539	103562	005737	003634	TST TR2	
19540	103566	001465		BEQ 1\$	
19541	103570	013701	003636	MOV TR3,R1	;ENTRY 3; ADJACENT; TR1=TR3+TR2
19542	103574	063701	003634	ADD TR2,R1	;REDUNDANT WITH ENTRY 7 IF TR2=0
19543	103600	010137	003632	MOV R1,TR1	
19544	103604	000207		RTS PC	
19545	103606	000000		0	
19546	103610	005737	003630	TST TR0	
19547	103614	001452		BEQ 1\$	
19548	103616	013701	003636	MOV TR3,R1	;ENTRY 4; ADJACENT; TR1=TR3-TR0
19549	103622	163701	003630	SUB TR0,R1	;REDUNDANT WITH ENTRY 7 IF TR0=0



19550	103626	010137	003632	MOV R1,TR1	
19551	103632	000207		RTS PC	
19552	103634	000000		0	
19553	103636	013701	003634	MOV TR2,R1	;ENTRY 5; PARTIAL OVERLAP; TR1=TR3+TR2-(TRO/2)
19554	103642	013702	003630	MOV TRO,R2	;REDUNDANT WITH ENTRY 7 IF TR2-(TRO/2)=0
19555	103646	006202		ASR R2	
19556	103650	160201		SUB R2,R1	
19557	103652	005701		TST R1	
19558	103654	001432		BEQ 1\$	
19559	103656	063701	003636	ADD TR3,R1	
19560	103662	010137	003632	MOV R1,TR1	
19561	103666	000207		RTS PC	
19562	103670	000000		0	
19563	103672	005737	003634	TST TR2	
19564	103676	001421		BEQ 1\$	
19565	103700	013701	003636	MOV TR3,R1	;ENTRY 6; PARTIAL OVERLAP; TR1=TR3-(TRO/2)
19566	103704	013702	003630	MOV TRO,R2	;REDUNDANT WITH ENTRY 5 IF TR2=0
19567	103710	006202		ASR R2	;REDUNDANT WITH ENTRY 7 IF TRO/2 =0
19568	103712	005702		TST R2	
19569	103714	001412		BEQ 1\$	
19570	103716	160201		SUB R2,R1	
19571	103720	010137	003632	MOV R1,TR1	
19572	103724	000207		RTS PC	
19573	103726	000000		0	
19574	103730	013737	003636 003632	MOV TR3,TR1	;ENTRY 7; COMPLETE OVERLAP; TR1=TR3
19575	103736	000207		RTS PC	
19576	103740	000000		0	
19577	103742	062706	000002	ADD #2,SP	;FIXJP STACK POINTER
19578	103746	000137	042334	JMP REDNTC	;SKIP ENTRY TEST CONDITION - REDUNDANT
19579					
19580	103752				
19581	103752	040001		T41A: .WORD 40001	
19582	103754	013701	003636	MOV TR3,R1	
19583	103760	063701	003634	ADD TR2,R1	
19584	103764	067701	075626	ADD @PTP14,R1	
19585	103770	010137	003632	MOV R1,TR1	
19586	103774	000207		RTS PC	
19587	103776	000000		0	
19588	104000			T411: .WORD 40001	
19589	104000	040001		MOV TR1,R1	;VARIABLE LENGTH ENTRY;1 ENTRY
19590	104002	013701	003632	ADD TRO,R1	;ENTRY 1; NO OVERLAP; TR3=TR1+TRO+SEP. CONSTANT
19591	104006	063701	003630	ADD @PTP14,R1	
19592	104012	067701	075600	MOV R1,TR3	
19593	104016	010137	003636	RTS PC	
19594	104022	000207		0	
19595	104024	000000			
19596	104026			T5: .WORD 401	
19597	104026	000401		.WORD 377	
19598	104030	000377		.WORD 0	
19599	104032	000000		.WORD 0	
19600	104034	000000		.WORD 0	
19601	104036	000000		.WORD 0	
19602					
19603	104040			T511: .WORD 0	

19604	104040	000403	.WORD	403	
19605	104042	000201	.WORD	201	
19606	104044	000377	.WORD	377	
19607	104046	000127	.WORD	127	
19608	104050	000000	.WORD	0	
19609	104052	000000	.WORD	0	
19610	104054	000000	.WORD	0	
19611					
19612	104056				T6:
19613	104056	040001	.WORD	40001	:DATA DESCRIPTOR ENTRIES;# OF ENTRIES=1
19614	104060	020001	.WORD	020001	:ENTRY TYPE =1; STARTING BYTE=1;INCREMENT=1
19615	104062	000001	.WORD	1	
19616	104064	000000	.WORD	0	
19617	104066	000000	.WORD	0	
19618	104070	000000	.WORD	0	
19619	104072	000000	.WORD	0	
19620	104074	000000	.WORD	0	
19621	104076	000000	.WORD	0	
19622					
19623	104100				T6A:
19624	104100	040001	.WORD	40001	:DATA DESCRIPTOR ENTRIES;# OF ENTRIES=1
19625	104102	020001	.WORD	020001	:ENTRY TYPE =1; STARTING BYTE=0;INCREMENT=1
19626	104104	000000	.WORD	0	
19627	104106	000000	.WORD	0	
19628	104110				T7:
19629	104110	040001	.WORD	40001	:DATA DESCRIPTOR ENTRIES;# OF ENTRIES=1
19630	104112	000240	.WORD	240	:ENTRY TYPE=0; ALL BYTES IDENTICAL=240
19631	104114	000000	.WORD	0	
19632	104116	000000	.WORD	0	
19633	104120	000000	.WORD	0	
19634	104122	000000	.WORD	0	
19635	104124	000000	.WORD	0	
19636					
19637	104126				T10:
19638	104126	000401	.WORD	401	
19639	104130	000003	.WORD	3	
19640	104132	000000	.WORD	0	
19641	104134	000000	.WORD	0	
19642	104136	000000	.WORD	0	
19643	104140	000000	.WORD	0	
19644					
19645	104142				T11:
19646	104142	040001	.WORD	40001	:DATA DESCRIPTOR ENTRIES; # OF ENTRIES=1
19647	104144	000252	.WORD	252	:ENTRY TYPE=0; ALL BYTES IDENTICAL =252
19648	104146	000000	.WORD	0	
19649	104150	000000	.WORD	0	
19650	104152	000000	.WORD	0	
19651	104154	000000	.WORD	0	
19652	104156	000000	.WORD	0	
19653					
19654	104160				T12:
19655	104160	040001	.WORD	40001	:DATA DESCRIPTOR ENTRIES; # OF ENTRIES=1
19656	104162	000360	.WORD	360	:ENTRY TYPE=0; ALL BYTES IDENTICAL = 360
19657	104164	000000	.WORD	0	

19658	104166	000000		.WORD	0
19659	104170	000000		.WORD	0
19660	104172	000000		.WORD	0
19661	104174	000000		.WORD	0
19662					
19663	104176		T13:		
19664	104176	000401		.WORD	401
19665	104200	000005		.WORD	5
19666	104202	000000		.WORD	0
19667	104204	000000		.WORD	0
19668	104206	000000		.WORD	0
19669	104210	000000		.WORD	0
19670					
19671	104212		T14:		
19672	104212	000401		.WORD	401
19673	104214	000010		.WORD	10
19674	104216	000000		.WORD	0
19675	104220	000000		.WORD	0
19676	104222	000000		.WORD	0
19677					
19678	104224		T14A:		
19679	104224	000401		.WORD	401
19680	104226	000100		.WORD	100
19681	104230	000000		.WORD	0
19682	104232	000000		.WORD	0
19683	104234	000000		.WORD	0
19684					
19685	104236		T15:		
19686	104236	000403		.WORD	403
19687	104240	000004		.WORD	4
19688	104242	000375		.WORD	375
19689	104244	000240		.WORD	240
19690	104246	000000		.WORD	0
19691	104250	000000		.WORD	0
19692	104252	000000		.WORD	0
19693	104254	000000		.WORD	0
19594					
19695	104256		T15A:		
19696	104256	000401		.WORD	401
19697	104260	000004		.WORD	4
19698	104262	000000		.WORD	0
19699	104264		T16A:		
19700	104264	040001		.WORD	40001
19701	104266	113701	003630	MOVB	TR0,R1
19702	104272	006201		ASR	R1
19703	104274	005201		INC	R1
19704	104276	063701	003632	ADD	TR1,R1
19705	104302	067701	075320	ADD	@PTP20,R1
19706	104306	010137	003636	MOV	R1,TR3
19707	104312	113701	003634	MOVB	TR2,R1
19708	104316	006201		ASR	R1
19709	104320	005201		INC	R1
19710	104322	063701	003636	ADD	TR3,R1
19711	104326	067701	075274	ADD	@PTP20,R1

:FOR USE WITH PACKED STRINGS ONLY  
:ENTRY 1; NO OVERLAP  
:TR3=TR1+[(TR0/2)+1]+SEP. CONST.  
:TR5=TR3+[(TR2/2)+1]+SEP. CONST

19712	104332	010137	003642		MOV R1, TR5	
19713	104336	000207			RTS PC	
19714	104340	000000			0	
19715						
19716	104342	040001		T14B:	.WORD 40001	;FOR USE WITH PACKED STRINGS ONLY
19717	104344	013737	003632	003636	MOV TR1, TR3	;S1A=S2A; S1 = S2
19718	104352	113701	003634		MOV B TR2, R1	;DA = S2A+[(S2L/2)+1]+SEP CONST
19719	104356	006201			ASR R1	
19720	104360	005201			INC R1	
19721	104362	063701	003636		ADD TR3, R1	
19722	104366	067701	075234		ADD @PTP20, R1	
19723	104372	010137	003642		MOV R1, TR5	
19724	104376	000207			RTS PC	
19725	104400	000000			0	
19726						
19727	104402			T16:		
19728	104402	040002			.WORD 40002	;FOR USE WITH PACKED STRINGS ONLY
19729	104404	113701	003630		MOV B TR0, R1	;ENTRY 1; NO OVERLAP
19730	104410	006201			ASR R1	; TR3=TR1+[(TR0/2)+1]+SEP. CONSTANT
19731	104412	005201			INC R1	; TR5=TR3+[(TR2/2)+1]+SEP. CONSTANT
19732	104414	063701	003632		ADD TR1, R1	
19733	104420	067701	075202		ADD @PTP20, R1	
19734	104424	010137	003636		MOV R1, TR3	
19735	104430	113701	003634		MOV B TR2, R1	
19736	104434	006201			ASR R1	
19737	104436	005201			INC R1	
19738	104440	063701	003636		ADD TR3, R1	
19739	104444	067701	075156		ADD @PTP20, R1	
19740	104450	010137	003642		MOV R1, TR5	
19741	104454	000207			RTS PC	
19742	104456	000000			0	
19743	104460	113701	003630		MOV B TR0, R1	;ENTRY 2; ALIGNED SRC2 - DST
19744	104464	006201			ASR R1	; TR3=TR1+[(TR0/2)+1]+SEP. CONSTANT
19745	104466	005201			INC R1	; TR5=TR3+(TR2/2)-(TR4/2)
19746	104470	063701	003632		ADD TR1, R1	
19747	104474	067701	075126		ADD @PTP20, R1	
19748	104500	010137	003636		MOV R1, TR3	
19749	104504	113701	003634		MOV B TR2, R1	
19750	104510	006201			ASR R1	
19751	104512	113702	003640		MOV B TR4, R2	
19752	104516	006202			ASR R2	
19753	104520	160201			SUB R2, R1	
19754	104522	063701	003636		ADD TR3, R1	
19755	104526	010137	003642		MOV R1, TR5	
19756	104532	013701	003636		MOV TR3, R1	;DST MUST NOT OVERLAP SRC1
19757	104536	167701	075064		SUB @PTP20, R1	;FIND END OF SRC1
19758	104542	023701	003642		CMP TR5, R1	;COMPARE END WITH START OF DST
19759	104546	103401			BLO 1\$	;OVERLAP-SKIP TEST CONDITION
19760	104550	000207			RTS PC	
19761	104552	062706	000002	1\$:	ADD #2, SP	;FIXUP STACK POINTER
19762	104556	000137	042620		JMP NXC	;SKIP TEST CONDITION - INVALID
19763	104562	000000			0	
19764						
19765	104564			T161:		

19766	104564	040002		.WORD 40002	:FOR USE WITH PACKED STRINGS ONLY
19767	104566	113701	003630	MOVB TR0,R1	:ENTRY 1: NO OVERLAP
19768	104572	006201		ASR R1	: TR3=TR1+[(TR0/2)+1]+SEP. CONSTANT
19769	104574	005201		INC R1	: TR5=TR3+[(TR2/2)+1]+SEP. CONSTANT
19770	104576	063701	003632	ADD TR1,R1	
19771	104602	067701	075020	ADD @PTP20,R1	
19772	104606	010137	003636	MOV R1,TR3	
19773	104612	113701	003634	MOVB TR2,R1	
19774	104616	006201		ASR R1	
19775	104620	005201		INC R1	
19776	104622	063701	003636	ADD TR3,R1	
19777	104626	067701	074774	ADD @PTP20,R1	
19778	104632	010137	003642	MOV R1,TR5	
19779	104636	000207		RTS PC	
19780	104640	000000		0	
19781	104642	113701	003630	MOVB TR0,R1	:ENTRY 2: ADJACENT
19782	104646	006201		ASR R1	: TR3=TR1+(TR0/2)+1
19783	104650	005201		INC R1	: TR5=TR3+(TR2/2)+1
19784	104652	063701	003632	ADD TR1,R1	
19785	104656	010137	003636	MOV R1,TR3	
19786	104662	113701	003634	MOVB TR2,R1	
19787	104666	006201		ASR R1	
19788	104670	005201		INC R1	
19789	104672	063701	003636	ADD TR3,R1	
19790	104676	010137	003642	MOV R1,TR5	
19791	104702	000207		RTS PC	
19792	104704	000000		0	
19793					
19794	104706				
19795	104706	040001		T162A: .WORD 40001	
19796	104710	113701	003630	MOVB TR0,R1	:ENTRY 1: NO OVERLAP
19797	104714	063701	003632	ADD TR1,R1	: TR3=TR1+TR0+SEP CONST
19798	104720	067701	074702	ADD @PTP20,R1	: TR5=TR3+TR2+SEP CONST
19799	104724	010137	003636	MOV R1,TR3	
19800	104730	113701	003634	MOVB TR2,R1	
19801	104734	063701	003636	ADD TR3,R1	
19802	104740	067701	074662	ADD @PTP20,R1	
19803	104744	010137	003642	MOV R1,TR5	
19804	104750	000207		RTS PC	
19805	104752	000000		0	
19806	104754			T162: .WORD 40002	:FOR USE WITH ZONED STRINGS
19807	104754	040002		MOVB TR0,R1	:ENTRY 1: NO OVERLAP
19808	104756	113701	003630	ADD TR1,R1	: TR3=TR1+TR0+SEP CONST
19809	104762	063701	003632	ADD @PTP20,R1	: TR5=TR3+TR2+SEP CONST
19810	104766	067701	074634	MOV R1,TR3	
19811	104772	010137	003636	MOVB TR2,R1	
19812	104776	113701	003634	ADD TR3,R1	
19813	105002	063701	003636	ADD @PTP20,R1	
19814	105006	067701	074614	MOV R1,TR5	
19815	105012	010137	003642	RTS PC	
19816	105016	000207		0	
19817	105020	000000		MOVB TR0,R1	:ENTRY 2 - ALIGNED SRC2 - DST
19818	105022	113701	003630	ADD TR1,R1	: TR3=TR1+TR0+SEP CONST
19819	105026	063701	003632		

19820	105032	067701	074570	ADD @PTP20,R1	; TR5=TR3+TR2-TR4
19821	105036	010137	003636	MOV R1,TR3	
19822	105042	113701	003634	MOVB TR2,R1	
19823	105046	063701	003636	ADD TR3,R1	
19824	105052	113702	003640	MOVB TR4,R2	
19825	105056	160201		SUB R2,R1	
19826	105060	010137	003642	MOV R1,TR5	
19827	105064	013701	003636	MOV TR3,R1	; DST MUST NOT OVERLAP SRC1
19828	105070	167701	074532	SUB @PTP20,R1	; FIND END OF SRC1
19829	105074	023701	003642	CMP TR5,R1	; COMPARE END WITH START OF DST
19830	105100	103401		BLO 1\$	
19831	105102	000207		RTS PC	
19832	105104	062706	000002	1\$: ADD #2,SP	; FIXUP STACK POINTER
19833	105110	000137	042620	JMP NXC	; SKIP TEST CONDITION - INVALID
19834	105114	000000		0	
19835					
19836	105116			T17:	
19837	105116	040004		.WORD 40004	
19838	105120	100037		.WORD 100037	; DATA DESCRIPTOR TYPE 4 - PACKED STRING,
19839	105122	003364		.WORD SSTG12	; PRE SPECIFIED STRING; SIGN POSITIVE
19840	105124	000000		.WORD 0	
19841	105126	060134		.WORD 060134	; DATA DESCRIPTOR TYPE 3 - PACKED STRING;
19842					; ALL DIGITS IDENTICAL = 5; SIGN POSITIVE
19843	105130	000000		.WORD 0	
19844	105132	060075		.WORD 060075	; DATA DESCRIPTOR TYPE 3 - PACKED STRING;
19845					; ALL DIGITS IDENTICAL = 3; SIGN NEGATIVE
19846	105134	000000		.WORD 0	
19847	105136	060014		.WORD 060014	; DATA DESCRIPTOR TYPE 3 - PACKED STRING;
19848					; ALL DIGITS IDENTICAL = 0; SIGN POSITIVE
19849	105140	000000		.WORD 0	
19850	105142	000000		.WORD 0	
19851	105144	000000		.WORD 0	
19852	105146	000000		.WORD 0	
19853	105150	000000		.WORD 0	
19854	105152	000000		.WORD 0	
19855	105154	000000		.WORD 0	
19856					
19857	105156	040002		T17A: .WORD 40002	
19858	105160	060174		.WORD 060174	; ALL DIGITS = 7+
19859	105162	000000		.WORD 0	
19860	105164	060014		.WORD 060014	; ALL DIGITS = 0+
19861	105166	000000		.WORD 0	
19862	105170	000000		.WORD 0	
19863	105172			T20:	
19864	105172	040001		.WORD 40001	; DATA DESCRIPTOR TYPE ENTRIES; # OF ENTRIES=1
19865	105174	000127		.WORD 127	; ENTRY TYPE=0; ALL BYTES IDENTICAL = 127
19866	105176	000000		.WORD 0	
19867	105200	000000		.WORD 0	
19868	105202	000000		.WORD 0	
19869	105204	000000		.WORD 0	
19870	105206	000000		.WORD 0	
19871					
19872	105210			T21:	
19873	105210	000402		.WORD 402	

19874	105212	000001	.WORD 1
19875	105214	000005	.WORD 5
19876	105216	000000	.WORD 0
19877	105220	000000	.WORD 0
19878	105222	000000	.WORD 0
19879			
19880	105224		T211:
19881	105224	000402	.WORD 402
19882	105226	000001	.WORD 1
19883	105230	000037	.WORD 37
19884	105232	000000	.WORD 0
19885	105234	000000	.WORD 0
19886	105236	000000	.WORD 0
19887	105240	000000	.WORD 0
19888			
19889	105242		T212:
19890	105242	000402	.WORD 402
19891	105244	000001	.WORD 1
19892	105246	000037	.WORD 37
19893	105250	000000	.WORD 0
19894	105252	000000	.WORD 0
19895	105254	000000	.WORD 0
19896	105256	000000	.WORD 0
19897	105260	000000	.WORD 0
19898			
19899	105262		TL21C:
19900	105262	000402	.WORD 402
19901	105264	000001	.WORD 1
19902	105266	000005	.WORD 5
19903	105270	000000	.WORD 0
19904	105272	000000	.WORD 0
19905	105274	000000	.WORD 0
19906	105276	000000	.WORD 0
19907			
19908	105300		TL22C:
19909	105300	000402	.WORD 402
19910	105302	000001	.WORD 1
19911	105304	000005	.WORD 5
19912	105306	000000	.WORD 0
19913	105310	000000	.WORD 0
19914	105312	000000	.WORD 0
19915	105314	000000	.WORD 0
19916	105316	000000	.WORD 0
19917			
19918	105320		T22:
19919	105320	040001	.WORD 40001
19920	105322	060074	.WORD 060074
19921	105324	000000	.WORD 0
19922	105326	000000	.WORD 0
19923	105330	000000	.WORD 0
19924	105332	000000	.WORD 0
19925	105334	000000	.WORD 0
19926	105336	000000	.WORD 0
19927			

;DATA DESCRIPTOR TYPE 3 - PACKED STRING  
; ALL DIGITS IDENTICAL = 3; SIGN POSITIVE

19928	105340		T23:		
19929	105340	040001		.WORD 40001	:DATA DESCRIPTOR TYPE ENTRIES; # OF ENTRIES = 1
19930	105342	000120		.WORD 120	:ENTRY TYPE=0; ALL BYTES IDENTICAL = 120
19931	105344	000000		.WORD 0	
19932	105346	000000		.WORD 0	
19933	105350	000000		.WORD 0	
19934	105352	000000		.WORD 0	
19935	105354	000000		.WORD 0	
19936					
19937	105356		T24:		
19938	105356	000401		.WORD 401	
19939	105360	120606		.WORD XLTL1	
19940	105362	000000		.WORD 0	
19941	105364	000000		.WORD 0	
19942	105366	000000		.WORD 0	
19943	105370	000000		.WORD 0	
19944					
19945	105372		T25:		
19946	105372	000402		.WORD 402	
19947	105374	000001		.WORD 1	
19948	105376	000240		.WORD 240	
19949	105400	000000		.WORD 0	
19950	105402	000000		.WORD 0	
19951	105404	000000		.WORD 0	
19952	105406	000000		.WORD 0	
19953					
19954	105410		T25A:		
19955	105410	000401		.WORD 401	
19956	105412	000007		.WORD 007	
19957	105414	000000		.WORD 0	
19958	105416		T26:		
19959	105416	001001		.WORD 1001	
19960	105420	040003		.WORD 040003	:DATA DESCRIPTOR TYPE 2 - CHAR STRING
19961	105422	003244		.WORD SSTG1	: STRING PRE-SPECIFIED AT SSTG1.
19962	105424	000000		.WORD 0	
19963	105426	000000		.WORD 0	
19964	105430	000000		.WORD 0	
19965	105432	000000		.WORD 0	
19966					
19967	105434		T27:		
19968	105434	000401		.WORD 401	
19969	105436	000256		.WORD 256	:TABLE LENGTH = 256 BYTES
19970	105440	000000		.WORD 0	
19971	105442	000000		.WORD 0	
19972	105444	000000		.WORD 0	
19973	105446	000000		.WORD 0	
19974					
19975	105450		T30:		
19976	105450	000403		.WORD 403	
19977	105452	000000		.WORD 0	
19978	105454	000001		.WORD 1	
19979	105456	000377		.WORD 377	
19980	105460	000000		.WORD 0	
19981	105462	000000		.WORD 0	



19982	105464	000000				.WORD 0	
19983	105466	000000				.WORD 0	
19984	105470				T30A:		
19985	105470	000401				.WORD 401	
19986	105472	000252				.WORD 252	
19987	105474	000000				.WORD 0	
19988	105476				T31A:		
19989	105476	040001				.WORD 40001	
19990	105500	013701	003630			MOV TR0,R1	;ENTRY 1; NO OVERLAP
19991	105504	063701	003632			ADD TR1,R1	;TR5 = TR0+TR1+SEP CONSTANT
19992	105510	067701	074102			ADD @PTP14,R1	
19993	105514	010137	003642			MOV R1,TR5	
19994	105520	000207				RTS PC	
19995	105522	000000				0	
19996	105524				T31:		
19997	105524	040002				.WORD 40002	
19998	105526	013701	003630			MOV TR0,R1	;ENTRY 1; NO OVERLAP
19999	105532	063701	003632			ADD TR1,R1	;TR5 = TR0+TR1+SEP CONSTANT
20000	105536	067701	074054			ADD @PTP14,R1	
20001	105542	010137	003642			MOV R1,TR5	
20002	105546	000207				PTS PC	
20003	105550	000000				0	
20004	105552	013737	003632	003642		MOV TR1,TR5	;ENTRY 2; OVERLAP
20005	105560	000207				RTS PC	;TR5 = TR1
20006	105562	000000				0	
20007							
20008	105564				T32:		
20009	105564	040003				.WORD 40003	;DATA DESCRIPTOR ENTRIES; # OF ENTRIES = 3
20010	105566	001000				.WORD 1000	;ENTRY TYPE 0; ALL BYTES IDENTICAL = 0
20011							; NOTE: 1000 RATHER THAN 0 IS USED HERE BECAUSE
20012							; 0 CAN BE USED ONLY FOR DELIMITING TABLE
20013							; ENTRIES.
20014	105570	000000				.WORD 0	;ENTRY DELIMITER.
20015	105572	000377				.WORD 377	;ENTRY TYPE 0; ALL BYTES IDENTICAL - 377
20016	105574	000000				.WORD 0	
20017	105576	020001				.WORD 20001	
20018	105600	000001				.WORD 1	
20019	105602	000000				.WORD 0	
20020	105604	000000				.WORD 0	
20021	105606	000000				.WORD 0	
20022	105610	000000				.WORD 0	
20023	105612	000000				.WORD 0	
20024							
20025	105614				T33:		
20026	105614	000403				.WORD 403	
20027	105616	000000				.WORD 0	
20028	105620	000001				.WORD 1	
20029	105622	000020				.WORD 20	
20030	105624	000000				.WORD 0	
20031	105626	000000				.WORD 0	
20032	105630	000000				.WORD 0	
20033	105632	000000				.WORD 0	
20034							
20035	105634	000401			T33A:	.WORD 401	

20036	105636	000012		.WORD 12
20037	105640	000000		.WORD 0
20038	105642	000000		.WORD 0
20039	105644	000000		.WORD 0
20040	105646	000000		.WORD 0
20041				
20042	105650		T331:	
20043	105650	000403		.WORD 403
20044	105652	000000		.WORD 0
20045	105654	000001		.WORD 1
20046	105656	000037		.WORD 37
20047	105660	000000		.WORD 0
20048	105662	000000		.WORD 0
20049	105664	000000		.WORD 0
20050	105666	000000		.WORD 0
20051	105670		T332:	
20052	105670	000403		.WORD 403
20053	105672	000000		.WORD 0
20054	105674	000001		.WORD 1
20055	105676	000037		.WORD 37
20056	105700	000000		.WORD 0
20057	105702	000000		.WORD 0
20058	105704	000000		.WORD 0
20059	105706	000000		.WORD 0
20060				
20061	105710	000401	T333:	.WORD 401
20062	105712	000037		.WORD 37
20063	105714	000000		.WORD 0
20064	105716	000000		.WORD 0
20065	105720	000000		.WORD 0
20066	105722	000000		.WORD 0
20067				
20068	105724		T34A:	
20069	105724	040001		.WORD 40001
20070	105726	113701	003630	MOV B TR0, R1
20071	105732	006201		ASR R1
20072	105734	005201		INC R1
20073	105736	063701	003632	ADD TR1, R1
20074	105742	067701	073646	ADD @PTP13, R1
20075	105746	010137	003636	MOV R1, TR3
20076	105752	000207		RTS PC
20077	105754	000000		0
20078	105756		T34:	
20079	105756	040002		.WORD 40002
20080	105760	113701	003630	MOV B TR0, R1
20081	105764	006201		ASR R1
20082	105766	005201		INC R1
20083	105770	063701	003632	ADD TR1, R1
20084	105774	067701	073614	ADD @PTP13, R1
20085	106000	010137	003636	MOV R1, TR3
20086	106004	000207		RTS PC
20087	106006	000000		0
20088	106010	113701	003630	MOV B TR0, R1
20089	106014	006201		ASR R1

:ENTRY 1; NO OVERLAP

:NOTE: SRC IS PACKED, THEREFORE # BYTES = # (DIGITS/2)+  
:TR3 = TR1+[(TR0/2)+1] + SEP CONSTANT

:ENTRY 1; NO OVERLAP

:NOTE: SRC IS PACKED, THEREFORE # BYTES = # (DIGITS/2)+  
:TR3 = TR1+[(TR0/2)+1] + SEP CONSTANT

:ENTRY 2; ADJACENT  
: TR3 = TR1+[(TR0/2)+1]

20090	106016	005201		INC R1	
20091	106020	063701	003632	ADD TR1,R1	
20092	106024	010137	003636	MOV R1,TR3	
20093	106030	000207		RTS PC	
20094	106032	000000		0	
20095					
20096	106034				
20097	106034	040003	T35:	.WORD 40003	
20098	106036	060074		.WORD 060074	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL=3; SIGN POSITIVE
20099					
20100	106040	000000		.WORD 0	
20101	106042	060215		.WORD 060215	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL=8; SIGN NEGATIVE
20102					
20103	106044	000000		.WORD 0	
20104	106046	060015		.WORD 060015	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL = 0;SIGN NEGATIVE
20105					
20106	106050	000000		.WORD 0	
20107	106052	000000		.WORD 0	
20108	106054	000000		.WORD 0	
20109	106056	000000		.WORD 0	
20110	106060	000000		.WORD 0	
20111	106062	000000		.WORD 0	
20112					
20113	106064		T35A:		
20114	106064	040004		.WORD 40004	
20115	106066	060074		.WORD 060074	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL=3; SIGN POSITIVE
20116					
20117	106070	000000		.WORD 0	
20118	106072	060215		.WORD 060215	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL=8; SIGN NEGATIVE
20119					
20120	106074	000000		.WORD 0	
20121	106076	060015		.WORD 060015	;DATA DESCRIPTOR TYPE 3 - PACKED STRING ; ALL DIGITS IDENTICAL = 0;SIGN NEGATIVE
20122					
20123	106100	000000		.WORD 0	
20124	106102	100037		.WORD 100037	;DATA DESCRIPTOR TYPE 4 - PACKED STRING ; PRE SPECIFIED STRING; SIGN POSITIVE
20125	106104	003364		.WORD SSTG12	
20126	106106	000000		.WORD 0	
20127	106110	000000		.WORD 0	
20128	106112	000000		.WORD 0	
20129	106114	000000		.WORD 0	
20130	106116	000000		.WORD 0	
20131	106120	000000		.WORD 0	
20132					
20133	106122	040001	TP19:	.WORD 40001	
20134	106124	100037		.WORD 100037	
20135	106126	003364		.WORD SSTG12	
20136	106130	000000		.WORD 0	
20137					
20138	106132	040002	TP19A:	.WORD 40002	
20139	106134	100013		.WORD 100013	;DATA DESC TYPE 4; FACKED STRING ;PRESPECIFID DATA; SIGN +
20140	106136	003604		.WORD SSTG14	
20141	106140	000000		.WORD 0	
20142	106142	100013		.WORD 100013	
20143	106144	003612		.WORD SSTG15	

20144	106146	000000		.WORD 0	
20145					
20146	106150	040001	TP19B:	.WORD 40001	
20147	106152	100017		.WORD 100017	
20148	106154	003620		.WORD SSTG16	
20149	106156	000000		.WORD 0	
20150					
20151	106160	040002	TP19C:	.WORD 40002	;PACKED STRING
20152	106162	060014		.WORD 060014	; ALL DIGITS IDENTICAL = 0; SIGN +
20153	106164	000000		.WORD 0	
20154	106166	060114		.WORD 060114	; ALL DIGITS IDENTICAL = 4; SIGN +
20155	106170	000000		.WORD 0	
20156					
20157	106172	040001	TP19D:	.WORD 40001	;PACKED STRING
20158	106174	060014		.WORD 060014	; ALL DIGITS IDENTICAL = 0; SIGN +
20159	106176	000000		.WORD 0	
20160					
20161	106200	040001	TP19E:	.WORD 40001	;PACKED STRING
20162	106202	060174		.WORD 060174	; ALL DIGITS IDENTICAL = 7; SIGN +
20163	106204	000000		.WORD 0	
20164					
20165	106206	040001	TP99:	.WORD 40001	;DATA DESC TYPE 3; PACKED STRING
20166	106210	060235		.WORD 060235	;ALL DIGITS IDENTICAL = 9; SIGN -
20167	106212	000000		.WORD 0	
20168					
20169	106214		T35B:		
20170	106214	040005		.WORD 40005	
20171	106216	060215		.WORD 060215	;DATA DESCRIPTOR TYPE 3 - PACKED STRING
20172					; ALL DIGITS IDENTICAL=8; SIGN NEGATIVE
20173	106220	000000		.WORD 0	
20174	106222	060015		.WORD 060015	;DATA DESCRIPTOR TYPE 3 - PACKED STRING
20175					; ALL DIGITS IDENTICAL = 0;SIGN NEGATIVE
20176	106224	000000		.WORD 0	
20177	106226	100037		.WORD 100037	;DATA DESCRIPTOR TYPE 4 - PACKED STRING
20178	106230	003364		.WORD SSTG12	; PRE SPECIFIED STRING; SIGN POSITIVE
20179	106232	000000		.WORD 0	
20180	106234	100037		.WORD 100037	;DATA DESCRIPTOR TYPE 4 - PACKED STRING
20181	106236	003404		.WORD STG12B	; PRE SPECIFIED STRING;SIGN NEG
20182	106240	000000		.WORD 0	
20183	106242	100037		.WORD 100037	
20184	106244	003424		.WORD STG12C	
20185	106246	000000		.WORD 0	
20186	106250	000000		.WORD 0	
20187	106252	000000		.WORD 0	
20188	106254	000000		.WORD 0	
20189	106256	000000		.WORD 0	
20190	106260	000000		.WORD 0	
20191	106262	000000		.WORD 0	
20192					
20193	106264		T36A:		
20194	106264	040001		.WORD 40001	
20195	106266	113701		MOVB TR0,R1	;ENTRY 1; NO OVERLAP
20196	106272	063701		ADD TR1,R1	; NOTE SRC IS ZONED THEREFORE #BYTES=#DIGITS
20197	106276	067701		ADD @PTP13,R1	;TR3 = TR0+TR1+SEP CONSTANT

20198	106302	010137	003636		MOV R1,TR3	
20199	106306	000207			RTS PC	
20200	106310	000000			0	
20201	106312			T36:		
20202	106312	040002			.WORD 40002	
20203	106314	113701	003630		MOVB TR0,R1	:ENTRY 1; NO OVERLAP
20204	106320	063701	003632		ADD TR1,R1	: NOTE SRC IS ZONED THEREFORE #BYTES=#DIGITS
20205	106324	067701	073264		ADD @PTP13,R1	:TR3 = TR0+TR1+SEP CONSTANT
20206	106330	010137	003636		MOV R1,TR3	
20207	106334	000207			RTS PC	
20208	106336	000000			0	
20209	106340	113701	003630		MOVB TR0,R1	
20210	106344	063701	003632		ADD TR1,R1	:ENTRY 2; ADJACENT
20211	106350	010137	003636		MOV R1,TR3	: TR3 = TR0 + TR1
20212	106354	123727	003631	000100	CMPB TR0+1,#100	:IF DATA TYPE = TRAILING SEPARATE
20213	106362	001002			BNE 1\$	: THEN TR3 = TR0 + TR1 + 1
20214	106364	005237	003636		INC TR3	
20215	106370	000207		1\$:	RTS PC	
20216	106372	000000			0	
20217						
20218	106374			T37:		
20219	106374	040003			.WORD 40003	
20220	106376	073463			.WORD 073463	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20221						: ALL DIGITS IDENTICAL =3; SIGN POSITIVE
20222						: HIGH NIBBLE = 7.
20223	106400	000000			.WORD 0	
20224	106402	074207			.WORD 074207	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20225						: ALL DIGITS IDENTICAL =8; SIGN NEGATIVE
20226						: HIGH NIBBLE = 8.
20227	106404	000000			.WORD 0	
20228	106406	070407			.WORD 070407	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20229						: ALL DIGITS IDENTICAL = 0; SIGN NEGATIVE
20230						: HIGH NIBBLE = 1.
20231	106410	000000			.WORD 0	
20232	106412	000000			.WORD 0	
20233	106414	000000			.WORD 0	
20234	106416	000000			.WORD 0	
20235	106420	000000			.WORD 0	
20236	106422	000000			.WORD 0	
20237						
20238	106424			T37A:		
20239	106424	040004			.WORD 40004	
20240	106426	073463			.WORD 073463	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20241						: ALL DIGITS IDENTICAL =3; SIGN POSITIVE
20242						: HIGH NIBBLE = 7.
20243	106430	000000			.WORD 0	
20244	106432	074207			.WORD 074207	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20245						: ALL DIGITS IDENTICAL =8; SIGN NEGATIVE
20246						: HIGH NIBBLE = 8.
20247	106434	000000			.WORD 0	
20248	106436	070407			.WORD 070407	:DATA DESCRIPTOR TYPE 3 - ZONED STRING
20249						: ALL DIGITS IDENTICAL = 0; SIGN NEGATIVE
20250						: HIGH NIBBLE = 1.
20251	106440	000000			.WORD 0	

20252	106442	110037	.WORD 110037	;DATA DESCRIPTOR TYPE 4 - ZONED STRING
20253	106444	003444	.WORD SSTG13	;PRE SPECIFIED STRING; SIGN POSTIVE
20254	106446	000000	.WORD 0	
20255	106450	000000	.WORD 0	
20256	106452	000000	.WORD 0	
20257	106454	000000	.WORD 0	
20258	106456	000000	.WORD 0	
20259	106460	000000	.WORD 0	
20260				
20261	106462		TZ19:	
20262	106462	040001	.WORD 40001	
20263	106464	110037	.WORD 110037	
20264	106466	003444	.WORD SSTG13	
20265	106470	000000	.WORD 0	
20266	106472		T37B:	
20267	106472	040005	.WORD 40005	
20268	106474	074207	.WORD 074207	;DATA DESCRIPTOR TYPE 3 - ZONED STRING ; ALL DIGITS IDENTICAL =8; SIGN NEGATIVE ; HIGH NIBBLE = 8.
20269				
20270				
20271	106476	000000	.WORD 0	
20272	106500	070407	.WORD 070407	;DATA DESCRIPTOR TYPE 3 - ZONED STRING ; ALL DIGITS IDENTICAL = 0; SIGN NEGATIVE ; HIGH NIBBLE = 1.
20273				
20274				
20275	106502	000000	.WORD 0	
20276	106504	110037	.WORD 110037	;DATA DESCRIPTOR TYPE 4 - ZONED STRING ;PRE SPECIFIED STRING; SIGN POSTIVE
20277	106506	003444	.WORD SSTG13	

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
PARAMETER TABLES (LENGTHS,ADDRESSES,ETC)

M 13  
PAGE 176 SEQUENCE 376

20279 106510 000000

.WORD 0

20281	106512	110037	.WORD 110037
20282	106514	003504	.WORD STG13B
20283	106516	000000	.WORD 0
20284	106520	110037	.WORD 110037
20285	106522	003544	.WORD STG13C
20286	106524	000000	.WORD 0
20287	106526	000000	.WORD 0
20288	106530	000000	.WORD 0
20289	106532	000000	.WORD 0
20290	106534	000000	.WORD 0
20291	106536	000000	.WORD 0
20292			
20293	106540		

T40:



20295	106540	000405	.WORD 405	;LONG INTEGER - HIGH WORD
20296	106542	000000	.WORD 0	: 0+
20297	106544	100000	.WORD 100000	: 0-
20298	106546	077777	.WORD 077777	: 77777+
20299	106550	177777	.WORD 177777	: 77777-
20300	106552	000005	.WORD 05	: 5+
20301	106554	000000	.WORD 0	
20302	106556	000000	.WORD 0	
20303	106560	000000	.WORD 0	
20304	106562	000000	.WORD 0	
20305	106564	000000	.WORD 0	
20306				
20307	106566		T40A:	
20308	106566	000401	.WORD 401	
20309	106570	000000	.WORD 0	
20310				
20311	106572	000403	T40B:	
20312	106574	000000	.WORD 403	
20313	106576	000231	.WORD 0	
20314	106600	000252	.WORD 231	
20315	106602	000000	.WORD 252	
20316	106604		.WORD 0	
20317	106604	000401	T41A:	
20318	106606	177777	.WORD 401	
20319	106610		.WORD 177777	
20320	106610	000401	T40AA:	
20321	106612	000005	.WORD 401	
20322	106614		.WORD 5	
20323	106614	000401	T41AA:	
20324	106616	000004	.WORD 401	
20325	106620		.WORD 4	
20326	106620	000403	T41:	
20327	106622	000000	.WORD 403	;LONG INTEGER - LOW WORD
20328	106624	000004	.WORD 0	
20329	106626	177777	.WORD 4	
20330	106630	000000	.WORD 177777	
20331	106632	000000	.WORD 0	
20332	106634	000000	.WORD 0	
20333	106636	000000	.WORD 0	
20334				
20335	106640	000403	T41B:	
20336	106642	120360	.WORD 403	
20337	106644	000000	.WORD 120360	
20338	106646	000125	.WORD 0	
20339	106650	000000	.WORD 125	
20340			.WORD 0	
20341	106652		T42:	
20342	106652	040001	.WORD 40001	
20343	106654	077623	.WORD 077623	;DATA DESCRIPTOR TYPE 3 - ZONED STRING
20344				: ALL DIGITS IDENTICAL = 9;SIGN +
20345				: HIGH NIBBLE = 17
20346	106656	000000	.WORD 0	
20347	106660	000000	.WORD 0	
20348	106662	000000	.WORD 0	

20349	106664	000000	.WORD 0
20350	106666	000000	.WORD 0
20351			
20352	106670		

T43:

20354	106670	000401		.WORD 401	
20355	106672	000011		.WORD 11	
20356	106674	000000		.WORD 0	
20357	106676	000000		.WORD 0	
20358	106700	000000		.WORD 0	
20359	106702	000000		.WORD 0	
20360					
20361	106704		T43A:		
20362	106704	000401		.WORD 401	
20363	106706	000101		.WORD 101	
20364	106710	000000		.WORD 0	
20365	106712	000000		.WORD 0	
20366	106714	000000		.WORD 0	
20367	106716	000000		.WORD 0	
20368					
20369	106720		T44A:		
20370	106720	040001		.WORD 40001	
20371	106722	113701	003630	MOVB TR0,R1	:ENTRY 1; NO OVERLAP
20372	106726	006201		ASR R1	: TR3=TR1+[(TR0/2)+1] + SEP CONST
20373	106730	005201		INC R1	
20374	106732	063701	003632	ADD TR1,R1	
20375	106736	067701	072654	ADD @PTP14,R1	
20376	106742	010137	003636	MOV R1,TR3	
20377	106746	000207		RTS PC	
20378	106750	000000		0	
20379	106752				
20380	106752	040002		.WORD 40002	
20381	106754	113701	003630	MOVB TR0,R1	:ENTRY 1; NO OVERLAP
20382	106760	006201		ASR R1	: TR3=TR1+[(TR0/2)+1] + SEP CONST
20383	106762	005201		INC R1	
20384	106764	063701	003632	ADD TR1,R1	
20385	106770	067701	072622	ADD @PTP14,R1	
20386	106774	010137	003636	MOV R1,TR3	
20387	107000	000207		RTS PC	
20388	107002	000000		0	
20389	107004	105737	003630	TSTB TR0	: IF EITHER SRC1=0 OR SRC2=0 BUT NOT BOTH
20390	107010	001004		BNE 1\$	: THEN SKIP THIS TEST CONDITION.
20391	107012	105737	003634	TSTB TR2	: SRC1=0
20392	107016	001404		BEQ 2\$	: BRANCH IF SRC2 ALSO = 0.
20393	107020	000417		BR 3\$	: SKIP TEST
20394	107022	105737	003634	1\$: TSTB TR2	: SRC1 NOT = 0
20395	107026	001414		BEQ 3\$	: SKIP TEST IF SRC2 = 0
20396	107030	113701	003630	2\$: MOVB TR0,R1	: ENTRY 2; ALIGNED SRC1-SRC2 LEAST SIGN DIGIT
20397	107034	006201		ASR R1	: TR3=TR1+(TR0/2)-(TR2/2)
20398	107036	113702	003634	MOVB TR2,R2	
20399	107042	006202		ASR R2	
20400	107044	160201		SUB R2,R1	
20401	107046	063701	003632	ADD TR1,R1	
20402	107052	010137	003636	MOV R1,TR3	
20403	107056	000207		RTS PC	
20404	107060	062706	000002	3\$: ADD #2,SP	: FIXUP STACK POINTER
20405	107064	000137	042620	JMP NXTC	: SKIP TEST CONDITION - INVALID
20406	107070	000000		0	
20407					

20408	107072			T45:		
20409	107072	040004			.WORD 40004	
20410	107074	100037			.WORD 100037	;DATA DESC. TYPE 4 - PACKED STRING
20411	107076	003364			.WORD SSTG12	; ALL DIGITS IDENTICAL = 3 SIGN +
20412	107100	000000			.WORD 0	
20413	107102	060075			.WORD 060075	;DATA DESC. TYPE 3 - PACKED STRING
20414						; ALL DIGITS IDENTICAL = 3, SIGN -
20415	107104	000000			.WORD 0	
20416	107106	060014			.WORD 060014	;DATA DESC TYPE 3 - PACKED STRING
20417						; ALL DIGITS IDENTICAL=0, SIGN POS
20418	107110	000000			.WORD 0	
20419	107112	060015			.WORD 060015	;DATA DESC TYPE 3 - PACKED STG
20420						; ALL DIGITS IDENTICAL = 0, SIGN -
20421	107114	000000			.WORD 0	
20422	107116	000000			.WORD 0	
20423	107120	000000			.WORD 0	
20424	107122	000000			.WORD 0	
20425	107124	000000			.WORD 0	
20426						
20427	107126			T46A:		
20428	107126	040001			.WORD 40001	;FOR CMPN ONLY
20429	107130	113701	003630		MOVB TR0,R1	;ENTRY 1; NO OVERLAP

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
PARAMETER TABLES (LENGTHS,ADDRESSES,ETC)

F 14  
PAGE 180 SEQUENCE 382

20431 107134 063701 003632

ADD TR1,R1

; TR3=TR1+TR0+SEP CONST

20433	107140	067701	072452	ADD @PTP14,R1
20434	107144	010137	003636	MOV R1,TR3

20436	107150	000207			RTS PC	
20437	107152	000000			0	
20438	107154			T46:		
20439	107154	040004			.WORD 40004	;FOR CMPN ONLY
20440	107156	113701	003630		MOV B TR0,R1	;ENTRY 1: NO OVERLAP
20441	107162	063701	003632		ADD TR1,R1	; TR3=TR1+TR0+SEP CONST
20442	107166	067701	072424		ADD @PTP14,R1	
20443	107172	010137	003636		MOV R1,TR3	
20444	107176	000207			RTS PC	
20445	107200	000000			0	
20446	107202	023727	002442	000001	CMP S1TYPE,#1	;SKIP TEST IS EITHER SRC DATA TYPE IS NOT 0 OR 1.
20447	107210	101066			BHI 2\$	
20448	107212	023727	002444	000001	CMP S2TYPE,#1	
20449	107220	101062			BHI 2\$	
20450	107222	105737	003630		TSTB TR0	;ENTRY 2: ADJACENT
20451	107226	001453			BEQ 1\$	; TR3=TR1+TR0
20452	107230	113701	003630		MOV B TR0,R1	; REDUNDANT WITH ENTRY 4 IF TR0=0
20453	107234	063701	003632		ADD TR1,R1	
20454	107240	010137	003636		MOV R1,TR3	
20455	107244	000207			RTS PC	
20456	107246	000000			0	
20457	107250	023727	002442	000001	CMP S1TYPE,#1	;SKIP TEST IS EITHER SRC DATA TYPE IS NOT 0 OR 1.
20458	107256	101043			BHI 2\$	
20459	107260	023727	002444	000001	CMP S2TYPE,#1	
20460	107266	101037			BHI 2\$	
20461	107270	113701	003630		MOV B TR0,R1	;ENTRY 3: PARTIAL OVERLAP
20462	107274	113702	003634		MOV B TR2,R2	; TR3=TR1+TR0-(TR2/2)
20463	107300	006202			ASR R2	; REDUNDANT WITH ENTRY 4 IF
20464	107302	160201			SUB R2,R1	; TR0-(TR2/2) =0
20465	107304	005701			TST R1	
20466	107306	001423			BEQ 1\$	
20467	107310	063701	003632		ADD TR1,R1	
20468	107314	010137	003636		MOV R1,TR3	
20469	107320	000207			RTS PC	
20470	107322	000000			0	
20471	107324	023727	002442	000001	CMP S1TYPE,#1	;SKIP TEST IS EITHER SRC DATA TYPE IS NOT 0 OR 1.
20472	107332	101015			BHI 2\$	
20473	107334	023727	002444	000001	CMP S2TYPE,#1	
20474	107342	101011			BHI 2\$	
20475	107344	013737	003632	003636	MOV TR1,TR3	;ENTRY 4: COMPLETE OVERLAP
20476	107352	000207			RTS PC	; TR3=TR1
20477	107354	000000			0	
20478	107356	062706	000002	1\$:	ADD #2,SP	;FIXUP STACK POINTER
20479	107362	000137	042334		JMP REDNTC	;SKIP ENTRY TEST CONDITION - REDUNDANT
20480	107366	062706	000002	2\$:	ADD #2,SP	
20481	107372	000137	042620		JMP NXTC	;SKIP TEST CONDITION - ILLEGAL
20482						
20483	107376			T47:		
20484	107376	040004			.WORD 40004	
20485	107400	070607			.WORD 070607	;DATA DESC TYPE 3 - ZONFD STRING
20486						; ALL DIGITS IDENTICAL = 8, SIGN -
20487						; HIGH NIBBLE = 1
20488	107402	000000			.WORD 0	
20489	107404	077603			.WORD 077603	;DATA DESC TYPE 3 - ZONED STRING

20490					: ALL DIGITS IDENTICAL = 8, SIGN +
20491					: HIGH NIBBLE = 17
20492	107406	000000	.WORD 0		
20493	107410	070403	.WORD 070403		: DATA DESC TYPE 3 - ZONED STG
20494					: ALL DIGITS IDENTICAL = 0, SIGN +
20495					: HIGH NIBBLE = 1
20496	107412	000000	.WORD 0		
20497	107414	074007	.WORD 074007		: DATA DESC TYPE 3 - ZONED STRING
20498					: ALL DIGITS IDENTICAL = 0, SIGN -
20499					: HIGH NIBBLE = 8
20500	107416	000000	.WORD 0		
20501	107420	000000	.WORD 0		
20502	107422	000000	.WORD 0		
20503	107424	000000	.WORD 0		
20504	107426	000000	.WORD 0		
20505	107430	000000	.WORD 0		
20506					
20507	107432			T50:	
20508	107432	000406	.WORD 406		
20509	107434	000000	.WORD 0		: RND.DGT=0, SHFT.CNT=0
20510	107436	002776	.WORD 002776		: RND.DGT=5, SHFT.CNT=-2
20511	107440	004402	.WORD 004402		: RND.DGT=9, SHFT.CNT=+2
20512	107442	001775	.WORD 001775		: RND.DGT=3, SHFT.CNT=-3
20513	107444	000775	.WORD 000775		: RND.DGT=1, SHFT.CNT=-3
20514	107446	000005	.WORD 000005		: RND.DGT=0, SHFT.CNT=5
20515	107450	000000	.WORD 0		
20516	107452	000000	.WORD 0		
20517	107454	000000	.WORD 0		
20518	107456	000000	.WORD 0		
20519	107460	000000	.WORD 0		
20520					
20521	107462			T50A:	
20522	107462	000401	.WORD 401		
20523	107464	002777	.WORD 002777		
20524					
20525	107466	000402	.WORD 402	T50B:	
20526	107470	003777	.WORD 003777		: RND.DGT = 7, SHFT.CT=-1
20527	107472	003402	.WORD 003402		: RND.DGT = 7, SHFT.CT=2
20528	107474	000000	.WORD 0		
20529					
20530	107476	000404	.WORD 404	T50C:	
20531	107500	003777	.WORD 3777		: RND.DGT = 7, SHFT.CT=-1
20532	107502	003776	.WORD 3776		: RND.DGT = 7, SHFT.CT=-2
20533	107504	000400	.WORD 0400		: RND.DGT = 1, SHFT.CT=0
20534	107506	004003	.WORD 4003		: RND.DGT = 8, SHFT.CT=3
20535	107510	000000	.WORD 0		
20536					
20537	107512	000402	.WORD 402	T50D:	
20538	107514	001376	.WORD 1376		: RND.DGT = 2, SHFT.CT=-2
20539	107516	004775	.WORD 4775		: RND.DGT = 9, SHFT.CT=-3
20540	107520	000000	.WORD 0		
20541					
20542	107522			T51:	
20543	107522	001004	.WORD 1004		: PACKED STRINGS



20544	107524	100012		.WORD 100012	;DATA DESC TYPE 4 - PRE SPECIFIED STRING
20545					; OF TEN DIGITS
20546	107526	003250		.WORD SSTG2	;+2,147,483,648
20547	107530	100012		.WORD 100012	
20548	107532	003304		.WORD SSTG4	;+2,147,483,647
20549	107534	100012		.WORD 100012	
20550	107536	003324		.WORD SSTG6	;-2,147,483,648
20551	107540	100012		.WORD 100012	
20552	107542	003344		.WORD SSTG10	;-2,147,483,649
20553	107544	100012		.WORD 100012	
20554	107546	003256		.WORD SSTG2A	;+4,294,967,294
20555	107550	100012		.WORD 100012	
20556	107552	003264		.WORD SSTG2B	;+42,949,672,940
20557	107554	000000		.WORD 0	
20558	107556	000000		.WORD 0	
20559	107560	000000		.WORD 0	
20560	107562	000000		.WORD 0	
20561					
20562	107564		T52:		
20563	107564	001004		.WORD 1004	;ZONED STRINGS
20564	107566	110012		.WORD 110012	;DATA DESC TYPE 4 - PRE SPECIFIED STRING
20565					; OF 10 DIGITS
20566	107570	003272		.WORD SSTG3	;+2,147,483,648
20567	107572	110012		.WORD 110012	
20568	107574	003312		.WORD SSTG5	;+2,147,483,647
20569	107576	110012		.WORD 110012	
20570	107600	003332		.WORD SSTG7	;-2,147,483,648
20571	107602	110012		.WORD 110012	
20572	107604	003352		.WORD SSTG11	;-2,147,483,649
20573	107606	000000		.WORD 0	
20574	107610	000000		.WORD 0	
20575	107612	000000		.WORD 0	
20576	107614	000000		.WORD 0	
20577					
20579	107616	000401	T53:	.WORD 401	
20580	107620	177777		.WORD 177777	
20581	107622	000000		.WORD 0	
20582	107624	000000		.WORD 0	
20583					
20584	107626	000401	T54:	.WORD 401	
20585	107630	111111		.WORD 111111	
20586	107632	000000		.WORD 0	
20587	107634	000000		.WORD 0	
20588					
20589					
20590	107636	000401	T55:	.WORD 401	
20591	107640	122222		.WORD 122222	
20592	107642	000000		.WORD 0	
20593	107644	000000		.WORD 0	
20594					
20595	107646	000401	T56:	.WORD 401	
20596	107650	133333		.WORD 133333	
20597	107652	000000		.WORD 0	
20598	107654	000000		.WORD 0	

PDP-11 CIS INST EXERCISER  
CZKEEB.P11

MACY11 27(655) 29-SEP-80 09:22  
PARAMETER TABLES (LENGTHS,ADDRESSES,ETC)

K 14  
PAGE 182-3 SEQUENCE 387

20599  
20600 107656 000401  
20601 107660 144444

T57: .WORD 401  
.WORD 144444

20603	107662	000000			.WORD 0	
20604	107664	000000			.WORD 0	
20605						
20606	107666	000401			T60: .WORD 401	
20607	107670	155555			.WORD 155555	
20608	107672	000000			.WORD 0	
20609	107674	000000			.WORD 0	
20610						
20611	107676				T61: .WORD 40010	:LOAD DESC - DESC ADDRESS TABLE
20612	107676	040010			BIC #7,TINST	:VARIABLE LENGTH ENTRIES: 8 ENTRIES
20613	107700	042737	000007	050012	MOV TBADR,TR0	:ENTRY 1;RN=R0;OPCODE = 0760X0
20614	107706	013737	001640	003630	MOV TBADR,R2	: LOAD PTR ADDRESS INTO R0
20615	107714	013702	001640		MOV R2,R1	: LOAD 1ST PTR WITH 1ST DESC ADDR.
20616	107720	010201			ADD TBLN,R1	: 1ST DESC ADDR=TBADR+TBLN-10
20617	107722	063701	001642		SUB #10,R1	
20618	107726	162701	000010		MOV R1,(R2)	
20619	107732	010112			SUB #10,R1	: LOAD 2ND PTR WITH 2ND DESC ADDR.
20620	107734	162701	000010		MOV R1,2(R2)	: 2ND DESC ADDR=TBADR+TBLN-20
20621	107740	010162	000002		SUB #10,R1	: LOAD 3RD PTR WITH 3RD DESC ADDR.
20622	107744	162701	000010		MOV R1,4(R2)	: 3RD DESC ADDR=TBADR+TBLN-30
20623	107750	010162	000004		MOV TINST,EINSTR	
20624	107754	013737	050012	046136	RTS PC	
20625	107762	000207			0	
20626	107764	000000				
20627						
20628	107766	052737	000001	050012	BIS #1,TINST	:ENTRY 2;RN=R1;OPCODE=0760X1
20629	107774	013737	001640	003632	MOV TBADR,TR1	
20630	110002	000744			BR 1\$	
20631	110004	000000			0	
20632						
20633	110006	042737	000007	050012	BIC #7,TINST	:ENTRY 3;RN=R2;OPCODE=0760X2
20634	110014	052737	000002	050012	BIS #2,TINST	
20635	110022	013737	001640	003634	MOV TBADR,TR2	
20636	110030	000731			BR 1\$	
20637	110032	000000			0	
20638						
20639	110034	052737	000003	050012	BIS #3,TINST	:ENTRY 4;RN=R3;OPCODE=0760X3
20640	110042	013737	001640	003636	MOV TBADR,TR3	
20641	110050	000721			BR 1\$	
20642	110052	000000			0	
20643						
20644	110054	042737	000007	050012	BIC #7,TINST	:ENTRY 5;RN=R4;OPCODE=0760X4
20645	110062	052737	000004	050012	BIS #4,TINST	
20646	110070	013737	001640	003640	MOV TBADR,TR4	
20647	110076	000706			BR 1\$	
20648	110100	000000			0	
20649						
20650	110102	052737	000005	050012	BIS #5,TINST	:ENTRY 6;RN=R5;OPCODE=0760X5
20651	110110	013737	001640	003642	MOV TBADR,TR5	
20652	110116	000676			BR 1\$	
20653	110120	000000			0	
20654						
20655	110122	042737	000007	050012	BIC #7,TINST	:ENTRY 7;RN=R6;OPCODE=0760X6
20656	110130	052737	000006	050012	BIS #6,TINST	

```

20657 110136 013737 001640 003644      MOV TBADR,TR6
20658 110144 000663                    BR 1$
20659 110146 000000                    0
20660
20661 110150 052737 000007 050012      BIS #7,TINST          ;ENTRY 8;RN=R7;OPCODE=0760X7
20662 110156 013702 001640            MOV TBADR,R2
20663 110162 063702 001642            ADD TLEN,R2
20664 110166 162702 000010            SUB #10,R2
20665 110172 010237 050014            MOV R2,TINST+2
20666 110176 162702 000010            SUB #10,R2
20667 110202 010237 050016            MOV R2,TINST+4
20668 110206 023727 050012 076067     CMP TINST,#076067
20669 110214 001021                    BNE 2$
20670 110216 162702 000010            SUB #10,R2
20671 110222 010237 050020            MOV R2,TINST+6
20672 110226 013737 001700 050022     MOV KBR2,TINST+10
20673 110234 013737 001672 050024 3$:  MOV KHALT,TINST+12
20674 110242 013737 001672 050026     MOV KHALT,TINST+14
20675 110250 013737 050012 046136     MOV TINST,EINSTR
20676 110256 000207                    RTS PC
20677 110260 013737 001676 050020 2$:  MOV KBR3,TINST+6
20678 110266 013737 001672 050022     MOV KHALT,TINST+10
20679 110274 000757                    BR 3$
20680 110276 000000                    0

```

```

20681
20682
20683 110300      STARS
20688           ;*****

```

```

(1)
(1)           .SBTTL READ A DECIMAL NUMBER FROM THE TTY
(1)
(1)           ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
(1)           ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
(1)           ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
(1)           ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
(1)           ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
(1)           ;*POSITIVE 32767 TO NEGATIVE 32768.
(1)           ;*CALL:
(1)           ;*      RDDEC          ;;READ A DECIMAL NUMBER
(1)           ;*      RETURN HERE    ;;NUMBER IS ON TOP OF THE STACK
(1)
(1)           $RDDEC: MOV      (SP),-(SP)      ;;PROVIDE SPACE FOR
(1)           110302 016666 000004 000002   MOV      4(SP),2(SP)      ;;THE INPUT NUMBER
(3)           110310 010046                   MOV      R0,-(SP)        ;;PUSH R0 ON STACK
(3)           110312 010146                   MOV      R1,-(SP)        ;;PUSH R1 ON STACK
(3)           110314 010246                   MOV      R2,-(SP)        ;;PUSH R2 ON STACK
(1)           110316 104402 1$:              RDLIN      ;;READ AN ASCII LINE
(1)           110320 012600                   MOV      (SP)+,R0        ;;ADDRESS OF 1ST CHAR.
(1)           110322 010037 110446           MOV      R0,6$          ;;SAVE INCASE OF BAD INPUT
(1)           110326 005046                   CLR      -(SP)          ;;CLEAR DATA WORD
(1)           110330 005002                   CLR      R2             ;;SIGN SET POSITIVE
(1)           110332 122710 000055           CMPB     #'-',(R0)      ;;SEE IF A MINUS SIGN WAS TYPED
(1)           110336 001001                   BNE     2$             ;;BR IF NO MINUS SIGN

```

```

(1) 110340 112002          MOVB      (R0)+,R2      ;;SAVE FOR LATER USE
(1) 110342 112001          2$: MOVB      (R0)+,R1      ;;PICKUP THIS CHARACTER
(1) 110344 001424          BEQ       3$           ;;GET OUT IF ZERO
(1) 110346 122701 000060  CMPB      #'0,R1      ;;MAKE SURE THIS CHARACTER
(1) 110352 003032          BGT       5$           ;;IS A DIGIT BETWEEN 0 & 9
(1) 110354 122701 000071  CMPB      #'9,R1
(1) 110360 002427          BLT       5$
(1) 110362 032716 170000  BIT       #'C7777,(SP) ;;DON'T LET NUMBER GET TO BIG
(1) 110366 001024          BNE       5$           ;;BR IF NUMBER WOULD OVERFLOW
(1) 110370 006316          ASL      (SP)          ;;*2
(1) 110372 011646          MOV      (SP),-(SP)   ;;SAVE FOR LATER
(1) 110374 006316          ASL      (SP)          ;;*4
(1) 110376 006316          ASL      (SP)          ;;*8
(1) 110400 062616          ADD      (SP)+,(SP)   ;;*10
(1) 110402 102416          BVS      5$           ;;OVERFLOW ISN'T ALLOWED
(1) 110404 162701 000060  SUB      #'0,R1      ;;STRIP AWAY THE ASCII JUNK
(1) 110410 060116          ADD      R1,(SP)     ;;ADD IN THIS DIGIT
(1) 110412 102412          BVS      5$           ;;OVERFLOW ISN'T ALLOWED
(1) 110414 000752          BR       2$           ;;LOOP
(1) 110416 005702          3$: TST      R2        ;;CHECK IF NUMBER IS NEG
(1) 110420 001401          BEQ      4$           ;;BR IF NO
(1) 110422 005416          NEG      (SP)         ;;YES--NEGATE THE NUMBER
(1) 110424 012666 000012  4$: MOV      (SP)+,12(SP) ;;SAVE THE RESULT
(3) 110430 012602          MOV      (SP)+,R2     ;;POP STACK INTO R2
(3) 110432 012601          MOV      (SP)+,R1     ;;POP STACK INTO R1
(3) 110434 012600          MOV      (SP)+,R0     ;;POP STACK INTO R0
(1) 110436 000002          RTI
(1) 110440 005726          5$: TST      (SP)+     ;;CLEAN PARTIAL NUMBER FROM STACK
(1) 110442 105010          CLRB     (R0)        ;;SET A TERMINATOR
(1) 110444 10440C          TYPE     ;;TYPE THE INPUT UP TO BAD CHAR.
(1) 110446 000000          6$: .WORD    0        ;;POINTER GOES HERE
(1) 110450 104400 110456  TYPE     ,SQUES      ;;'?' 'CR' & 'LF'
(1) 110454 000720          BR       1$           ;;TRY AGAIN
(1) 110456 077           SQUES: .ASCII  "?"    ;;QUESTION MARK
(1) 110457 015           $CRLF: .ASCII  <15>  ;;CARRIAGE RETURN
(1) 110460 000012          $LF:    .ASCIIZ <12> ;;LINEFEED
20689 ;*****
(1)
(1)
(1)
(1) 110462 177560          $TKS: .WORD    177560 ;;TTY KBD STATUS
(1) 110464 177562          $TKB: .WORD    177562 ;;TTY KBD BUFFER
(1) ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*CALL:
(1) ;* RDCHR      ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;* RETURN HERE ;;CHARACTER IS ON THE STACK
(1) ;
(1)
(1) 110466 011646          $RDCHR: MOV      (SP),-(SP) ;;PUSH DOWN THE PC
(1) 110470 016666 000004 000002  MOV      4(SP),2(SP) ;;SAVE THE PS
(1) 110476 105777 177760  1$: TSTB     @TKS      ;;WAIT FOR
(1) 110502 100375          BPL      1$           ;;A CHARACTER
(1) 110504 117766 177754 000004  MOVB     @TKB,4(SP)   ;;READ THE TTY

```

```

(1) 110512 042766 177600 000004      BIC      #*(C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1) 110520 000002                      RTI        ;;GO BACK TO USER
(2)                                     ;*****
(1)                                     ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                     ;*CALL:
(1)                                     ;*
(1)                                     ;*   IN                               ;;INPUT A STRING FROM THE TTY
(1)                                     ;*   RETURN HERE                       ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                     ;*                                     ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1) 110522 010346                      $PDLIN: MOV    R3,-(SP)      ;;SAVE R3
(1) 110524 012703 110630                1$:  MOV    #STTYIN,R3     ;;GET ADDRESS
(1) 110530 022703 110640                2$:  CMP    #STTYIN+8.,R3  ;;BUFFER FULL?
(1) 110534 101405                      BLOS    4$                ;;BR IF YES
(1) 110536 104401                      RDCHR                               ;;GO READ ONE CHARACTER FROM THE TTY
(1) 110540 112613                      MOVB    (SP)+,(R3)        ;;GET CHARACTER
(1) 110542 122713 000177                CMPB    #177,(R3)        ;;IS IT A RUBOUT
(1) 110546 001003                      BNE     3$                ;;SKIP IF NOT
(1) 110550 104400 110456                4$:  TYPE    ,SQUES        ;;TYPE A '?'
(1) 110554 000763                      BR      1$                ;;CLEAR THE BUFFER AND LOOP
(1) 110556 111337 110626                3$:  MOVB    (R3),9$       ;;ECHO THE CHARACTER
(1) 110562 104400 110626                TYPE    ,9$
(1) 110566 122723 000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1) 110572 001356                      BNE     2$                ;;LOOP IF NOT RETURN
(1) 110574 105063 177777                CLRB    -1(R3)           ;;CLEAR RETURN (THE 15)
(1) 110600 104400 110460                TYPE    ,SLF            ;;TYPE A LINE FEED
(1) 110604 012603                      MOV     (SP)+,R3         ;;RESTORE R3
(1) 110606 011646                      MOV     (SP),-(SP)       ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 110610 016666 000004 000002        MOV     4(SP),2(SP)      ;;FIRST ASCII CHARACTER ON IT
(1) 110616 012766 110630 000004        MOV     #STTYIN,4(SP)
(1) 110624 000002                      RTI        ;;RETURN
(1) 110626 000                      9$:  .BYTE   0             ;;STORAGE FOR ASCII CHAR. TO TYPE
(1) 110627 000                      .BYTE   0             ;;TERMINATOR
(1) 110630 000010                      $TTYIN: .BLKB  8.       ;;RESERVE 8 BYTES FOR TTY INPUT
(1)                                     ;*****
(1)                                     ;.SBTTL TYPE ROUTINE
(1)                                     ;*ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                     ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                     ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                     ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                     ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                     ;*
(1)                                     ;*CALL:
(1)                                     ;*1) USING A TRAP INSTRUCTION
(1)                                     ;*   TYPE    ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCII STRING
(1)                                     ;*OR
(1)                                     ;*   TYPE
(1)                                     ;*   MESADR
(1)                                     ;*
(1) 110640 105737 111213                $TYPE: TSTB    $TPFLG    ;;IS THERE A TERMINAL?
(1) 110644 100002                      BPL     1$                ;;BR IF YES
(1) 110646 000000                      HALT                               ;;HALT HERE IF NO TERMINAL

```

20690

4

```

(1) 110650 000430          BR      3$          ;; LEAVE
(1) 110652 010046          MOV     RO,-(SP)    ;; SAVE RO
(1) 110654 017600 000002  1$:  MOV     @2(SP),RO  ;; GET ADDRESS OF ASCIZ STRING
(1) 110660 122737 000001  001134  CMPB   #APTENV,$ENV ;; RUNNING IN APT MODE
(1) 110666 001011          BNE    62$          ;; NO,GO CHECK FOR APT CONSOLE
(1) 110670 132737 000100  001135  BITB   #APTSPOOL,$ENVM ;; SPOOL MESSAGE TO APT
(1) 110676 001405          BEQ    62$          ;; NO,GO CHECK FOR CONSOLE
(1) 110700 010037 110710  MOV     RO,61$     ;; SETUP MESSAGE ADDRESS FOR APT
(1) 110704 004737 001156  JSR    PC,$ATY3    ;; SPOOL MESSAGE TO APT
(1) 110710 000000          .WORD  0          ;; MESSAGE ADDRESS
(1) 110712 132737 000040  001135  61$:  BITB   #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
(1) 110720 001003          BNE    60$          ;; YES,SKIP TYPE OUT
(1) 110722 112046          2$:  MOVB   (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 110724 001005          BNE    4$          ;; BR IF IT ISN'T THE TERMINATOR
(1) 110726 005726          TST   (SP)+       ;; IF TERMINATOR POP IT OFF THE STACK
(1) 110730 012600          60$:  MOV     (SP)+,RO  ;; RESTORE RO
(1) 110732 062716 000002  3$:  ADD     #2,(SP)   ;; ADJUST RETURN PC
(1) 110736 000002          RTI                    ;; RETURN
(1) 110740 122716 000011  4$:  CMPB   #THT,(SP)  ;; BRANCH IF <HT>
(1) 110744 001426          BEQ    8$          ;;
(1) 110746 122716 000200  CMPB   #TCRLF,(SP) ;; BRANCH IF NOT <CRLF>
(1) 110752 001004          BNE    5$          ;;
(1) 110754 005726          TST   (SP)+       ;; POP <CR><LF> EQUIV
(1) 110756 104400          TYPE                    ;; TYPE A CR AND LF
(1) 110760 110457          $CRLF
(1) 110762 000757          BR      2$          ;; GET NEXT CHARACTER
(1) 110764 004737 111046  5$:  JSR    PC,$TYPEC  ;; GO TYPE THIS CHARACTER
(1) 110770 123726 111212  6$:  CMPB   $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
(1) 110774 001352          BNE    2$          ;; IF NO GO GET NEXT CHAR.
(1) 110776 013746 111210  MOV     $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
(1)                          ;; AND THE NULL CHAR.
(1) 111002 105366 000001  7$:  DECB   1(SP)      ;; DOES A NULL NEED TO BE TYPED?
(1) 111006 002770          BLT   6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK
(1) 111010 004737 111046  JSR    PC,$TYPEC  ;; GO TYPE A NULL
(1) 111014 105337 111176  DECB   $CHARCNT   ;; DO NOT COUNT AS A COUNT
(1) 111020 000770          BR      7$          ;; LOOP
(1)
(1)                          ;HORIZONTAL TAB PROCESSOR
(1)
(1) 111022 112716 000040  8$:  MOVB   #40,(SP)   ;; REPLACE TAB WITH SPACE
(1) 111026 004737 111046  9$:  JSR    PC,$TYPEC  ;; TYPE A SPACE
(1) 111032 132737 000007  111176  BITB   #7,$CHARCNT ;; BRANCH IF NOT AT
(1) 111040 001372          BNE    9$          ;; TAB STOP
(1) 111042 005726          TST   (SP)+       ;; POP SPACE OFF STACK.
(1) 111044 000726          BR      2$          ;; GET NEXT CHARACTER
(1) 111046 105777 000132  $TYPEC: TSTB   @2TPS  ;; WAIT UNTIL PRINTER IS READY
(1) 111052 100375          BPL   $TYPEC
(1) 111054 105777 070606  TSTB   @TKS       ;; CHECK FOR XOFF
(1) 111060 100027          BPL 2$          ;; BRANCH IF NO RECEIVE CHAR PRESENT
(1) 111062 117737 070624 111202  MOVB   @TKB,XOCHAR ;; READ AND SAVE TTY CHAR
(1) 111070 042737 177600 111202  BIC    #^C177,XOCHAR
(1) 111076 023727 111202 000023  CMP    XOCHAR,#023 ;; IS CHAR A ^S
(1) 111104 001015          BNE 2$          ;; BRANCH IF NO
(1) 111106 105777 070554  3$:  TSTB   @TKS       ;; WAIT FOR XON

```

```

(1) 111112 100375          BPL      3$
(1) 111114 117737 070572 064776  MOVB    @TKB,RCHAR
(1) 111122 042737 177600 064776  BIC     #^C177,RCHAR
(1) 111130 023727 064776 000021  CMP     RCHAR,#021      ;IGNORE ALL CHARS EXCEPT ^Q
(1) 111136 001363          BNE     3$
(1) 111140 116677 000002 000040 2$:  MOVB    2(SP),@STPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 111146 122766 000015 000002  CMPB   #15,2(SP)      ;;BRANCH IF
(1) 111154 001003          BNE     1$              ;;NOT <CR>
(1) 111156 105037 111176          CLRB   $CHARCNT        ;;
(1) 111162 000406          BR     $TYPEX          ;;EXIT
(1) 111164 122766 000012 000002 1$:  CMPB   #12,2(SP)      ;;BRANCH IF
(1) 111172 002002          BGE     $TYPEX          ;;<LF>
(1) 111174 105227          INCB   (PC)+           ;;INC SPACE
(1) 111176 000000          $CHARCNT: .WORD 0      ;;COUNT
(1) 111200 000207          $TYPEX:  RTS          PC
(1) 111202 000000          XOCHAR: .WORD 0
(1)                               ;; EQUATES
(1)                               THT=11
(1)                               TCRLF=200
(1)
(1) 111204 177564          $TPS:   .WORD 177564    ;;TTY PRINTER STATUS REG. ADDRESS
(1) 111206 177566          $TPB:   .WORD 177566    ;;TTY PRINTER BUFFER REG. ADDRESS
(1) 111210 000          $NULL:  .BYTE 0         ;;CONTAINS NULL CHARACTER FOR FILLS
(1) 111211 002          $FILLS: .BYTE 2         ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 111212 012          $FILLC: .BYTE 12        ;;INSERT FILL CHARS. AFTER A 'LINE FEED'
(1) 111213 000          $TPFLG: .BYTE 0        ;;'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
20691 *****
(1)
(1)                               .SBTTL TRAP DECODER
(1)
(1)                               ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)                               ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)                               ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1)                               ;*GO TO THAT ROUTINE.
(1)
(1) 111214 010046          $TRAP:  MOV     RO,-(SP)    ;;SAVE RO
(1) 111216 016600 000002  MOV     2(SP),RO        ;;GET TRAP ADDRESS
(1) 111222 005740          TST    -(RO)           ;;BACKUP BY 2
(1) 111224 111000          MOVB   (RO),RO         ;;GET RIGHT BYTE OF TRAP
(1) 111226 006300          ASL    RO              ;;POSITION FOR INDEXING
(1) 111230 016000 111236  MOV     $TRPAD(RO),RO    ;;INDEX TO TABLE
(1) 111234 000200          RTS    RO              ;;GO TO ROUTINE
(1)
(1)                               .SBTTL TRAP TABLE
(1)
(1)                               ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(1)                               ;*BY THE "TRAP" INSTRUCTION.
(1)
(1)                               ; ROUTINE
(1)                               ; -----
(1) 111236          $TRPAD:  $TYPE  ;;CALL=TYPE  TRAP+0(104400) TTY TYPEOUT ROUTINE
(1) 111236 110640          $RDCHR  ;;CALL=RDCHR  TRAP+1(104401) TTY TYPEIN CHARACTER ROUTINE
(1) 111240 110466

```



(3) 111242 110522  
 (3) 111244 110300  
 20692  
 (1)  
 (1)  
 (1)  
 (1)  
 (1) 111246 012737 111370 000024  
 (1) 111254 012737 000340 000026  
 (3) 111262 010046  
 (3) 111264 010146  
 (3) 111266 010246  
 (3) 111270 010346  
 (3) 111272 010446  
 (3) 111274 010546  
 (1) 111276 010637 111374  
 (1) 111302 012737 111314 000024  
 (1) 111310 000000  
 (1) 111312 000776  
 (1)  
 (1)  
 (1) 111314 013706 111374  
 (1) 111320 005037 111374  
 (1) 111324 005237 111374  
 (1) 111330 001375  
 (3) 111332 012605  
 (3) 111334 012604  
 (3) 111336 012603  
 (3) 111340 012602  
 (3) 111342 012601  
 (3) 111344 012600  
 (1) 111346 012737 111246 000024  
 (1) 111354 012737 000340 000026  
 (1) 111362 104400  
 (1) 111364 111374  
 (1) 111366 000002  
 (1) 111370 000000  
 (1) 111372 000776  
 (1) 111374 000000  
 (1) 111376 005015 047520 042527  
 (1) 111404 000122  
 (1)

```

$RDLIN ;;CALL=RDLIN TRAP+2(104402) TTY TYPEIN STRING ROUTINE
$RDDEC ;;CALL=RDDEC TRAP+3(104403) READ A DECIMAL NUMBER FROM TTY
:*****
.SBTTL POWER DOWN AND UP ROUTINES
:POWER DOWN ROUTINE
$PWRDN: MOV $SILLUP,@#PWRVEC ;;SET FOR FAST UP
MOV #340,@#PWRVEC+2 ;;PRIO:7
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
MOV R4,-(SP) ;;PUSH R4 ON STACK
MOV R5,-(SP) ;;PUSH R5 ON STACK
MOV SP,$SAVR6 ;;SAVE SP
MOV #PWRUP,@#PWRVEC ;;SET UP VECTOR
HALT
BR -2 ;;HANG UP

:POWER UP ROUTINE
$PWRUP: MOV $SAVR6,SP ;;GET SP
CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
1$: INC $SAVR6 ;;WAIT FOR THE INC
BNE 1$ ;;OF WORD
MOV (SP)+,R5 ;;POP STACK INTO R5
MOV (SP)+,R4 ;;POP STACK INTO R4
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
MOV (SP)+,R0 ;;POP STACK INTO R0
MOV #PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
MOV #340,@#PWRVEC+2 ;;PRIO:7
TYPE REPORT THE POWER FAILURE
$PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
RTI
$SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
BR -2 ;; BEFORE THE POWER DOWN WAS COMPLETE
$SAVR6: 0 ;;PUT THE SP HERE
$POWER: .ASCIZ <15><12>'POWER'

```

20694  
 20699 000377  
 20700 076130  
 20701  
 20702  
 20703  
 20704  
 20705  
 20706  
 20707  
 20708  
 20709

```

.EVEN
:*****
FILL=377
MOVCI=76130
:CIS STACK 'PROBE AHEAD' MEMORY MANAGEMENT ABORT TESTS
:
:NOTE: THESE THREE TESTS ARE FOR THE 11/44 ONLY
:THEY ARE NOT TO BE RUN ON THE 11/23 OR 11/24 BECAUSE
:THESE PROCESSORS TREAT PROBE AHEAD DIFFERENTLY - THEY
:ONLY PROBE AHEAD SPECIFIC NUMBER OF WORDS NEEDED FOR
:GIVEN INST.
:
```

20710  
 20711  
 20712  
 20713  
 20714  
 20715  
 20716  
 20717  
 20718  
 20719  
 20720  
 20721  
 20722  
 20723  
 20724  
 20725  
 20726 111406 005737 002156  
 20727 111412 001002  
 20728 111414 000137 053354  
 20729  
 20730 111420 005737 002154  
 20731 111424 001373  
 20732 111426 005037 172516  
 20733 111432 010637 112556  
 20734  
 20735 111436 012737 000000 172340  
 20736 111444 012737 000200 172342  
 20737 111452 012737 000400 172344  
 20738 111460 012737 000600 172346  
 20739 111466 012737 001000 172350  
 20740 111474 012737 001200 172352  
 20741 111502 012737 001400 172354  
 20742 111510 012737 177600 172356  
 20743  
 20744 111516 012737 000000 172240  
 20745 111524 012737 000200 172242  
 20746 111532 012737 000400 172244  
 20747 111540 012737 000600 172246  
 20748 111546 012737 001000 172250  
 20749 111554 012737 001200 172252  
 20750 111562 012737 001400 172254  
 20751 111570 012737 177600 172256  
 20752  
 20753 111576 012737 000000 177640  
 20754 111604 012737 000200 177642  
 20755 111612 012737 000400 177644  
 20756 111620 012737 000600 177646  
 20757 111626 012737 001000 177650  
 20758 111634 012737 001200 177652  
 20759 111642 012737 001400 177654  
 20760 111650 012737 177600 177656  
 20761  
 20762  
 20763

THE NEXT THREE TESTS ARE AIMED AT TESTING THE KT PAGE FAULT ROM (11/44 SCHEMATIC PAGE K1-8) AND ASSOCIATED LOGIC.

EACH OF THESE 3 TESTS SETUP THE STACK POINTER SUCH THAT WHEN THE CIS PROCESSOR CHECKS TO SEE IF THERE IS ENOUGH SPACE ON THE STACK (200 BYTES) THE ANSWER FOUND SHOULD BE NO BECAUSE A PORTION OF THE STACK IS IN PROTECTED MEMORY.

EACH OF THESE 3 TESTS VERIFY THAT THE CIS INSTRUCTION ABORTS UNDER SEVERAL CONDITIONS OF MEMORY MANAGEMENT PAGE PROTECTION. ALL OF THE PAGE PROTECTION DATA IS LISTED IN TABLE 'PDRTAB'.

; SETUP PAR'S FOR DIRECT MAPPING

PROBAH: TST MMFLG ;11/70 TYPE MEM MGMT?  
 BNE 4\$ ;BRANCH IF YES  
 5\$: JMP DONE ;SKIP THESE MEM MGMT ABORT TESTS

4\$: TST PT34 ;IS THIS AN 11/34 TYPE PROCESSOR?  
 BNE 5\$ ;BRANCH IF YES  
 CLR @MMR3 ;CLEAR OUT D-SPACE ENABLES  
 MOV SP,STK1 ;SAVE STACK POINTER

MOV #0,@#KIPAR0 ;SETUP KERNEL I-SPACE PAR'S  
 MOV #200,@#KIPAR1  
 MOV #400,@#KIPAR2  
 MOV #600,@#KIPAR3  
 MOV #1000,@#KIPAR4  
 MOV #1200,@#KIPAR5  
 MOV #1400,@#KIPAR6  
 MOV #177600,@#KIPAR7

MOV #0,@#SIPAR0 ;SETUP SUPERVISOR I-SPACE PAR'S  
 MOV #200,@#SIPAR1  
 MOV #400,@#SIPAR2  
 MOV #600,@#SIPAR3  
 MOV #1000,@#SIPAR4  
 MOV #1200,@#SIPAR5  
 MOV #1400,@#SIPAR6  
 MOV #177600,@#SIPAR7

MOV #0,@#UIPAR0 ;SETUP USER I-SPACE PAR'S  
 MOV #200,@#UIPAR1  
 MOV #400,@#UIPAR2  
 MOV #600,@#UIPAR3  
 MOV #1000,@#UIPAR4  
 MOV #1200,@#UIPAR5  
 MOV #1400,@#UIPAR6  
 MOV #177600,@#UIPAR7

; SETUP PDR'S FOR R/W ACCESS

20764				:	
20765					
20766	111656	012700	172300		MOV #KIPDR0,RO
20767	111662	012720	177406	1\$:	MOV #177406,(RO)+
20768	111666	020027	172316		CMP RO,#KIPDR7
20769	111672	101773			BLOS 1\$
20770					
20771	111674	012700	172200		MOV #SIPDR0,RO
20772	111700	012720	177406	2\$:	MOV #177406,(RO)+
20773	111704	020027	172216		CMP RO,#SIPDR7
20774	111710	101773			BLOS 2\$
20775					
20776	111712	012700	177600		MOV #UIPDR0,RO
20777	111716	012720	177406	3\$:	MOV #177406,(RO)+
20778	111722	020027	177616		CMP RO,#UIPDR7
20779	111726	101773			BLOS 3\$
20780					

```
20782 ;*****
20783 ;SBTTL MEMORY MGMT ABORT TESTS
20784 ;KERNEL MODE CIS STACK PROBE AHEAD MEM MGMT ABORT TESTS
20785 ;
20786
20787 111730 KMTSTS:
20788 111730 005237 001120 INC $TESTN ;UPDATE TEST NUMBER FOR APT
20789 111734 012737 112434 000250 MOV #MMHDLR,@#MMVEC ;SETUP MEM MGMT INTERRUPT VECTOR
20790 111742 012737 000340 000252 MOV #PR7,@#MMVEC+2
20791 111750 012701 112040 MOV #1$,P1 ;SETUP INTR RETURN ADDRESS
20792 111754 012700 112560 MOV #PDRTAB,R0
20793 111760 012706 060070 3$: MOV #60070,SP
20794
20795 111764 011037 172304 2$: MOV (R0),@#KIPDR2 ;PROTECT PART OF STACK
20796 111770 012737 000001 177572 MOV #1,@#MMRO ;TURN ON MEMORY MGMT
20797
20798 111776 004737 112456 JSR PC,SAVR ;SAVE REGISTERS
20799
20800 112002 076130 MOVCI ;EXECUTE THE CIS INSTRUCTION
20801 112004 112446 SRC.PTR
20802 112006 112452 DST.PTR
20803 112010 000377 FILL
20804 112012 000240 NOP ;TO LOOP ON THIS TEST REPLACE
20805 112014 000240 NOP ; THE TWO NOPS WITH A JMP TO 3$.
20806 112016 PRINTB #NOABO ;PRINT CIS INST FAILED TO ABORT
(6) 112016 012746 013561 MOV #NOABO,-(SP)
(3) 112022 010600 MOV SP,R0
(4) 112024 004737 065410 JSR PC,FPRINT
20807 112030 012737 000001 001116 MOV #1,$FATAL ;SET APT FATAL ERROR INDICATOR
20808 112036 000000 HALT ;CIS INSTRUCTION SHOULD HAVE ABORTED BUT DIDN'T
20809
20810 112040 004737 112510 1$: JSR PC,RESR ;RESTORE REGISTERS
20811 112044 062700 000002 ADD #2,R0 ;UPDATE PROTECTION SCHEME TO NEXT TABLE CASE
20812 112050 005710 TST (R0) ;ANY CASES LEFT TO TRY?
20813 112052 001344 BNE 2$ ;BRANCH IF YES
20814
20815 112054 005037 177572 CLR @#MMRO ;NO - PREPARE TO EXIT TEST
20816 112060 012737 177406 172304 MOV #177406,@#KIPDR2 ;RESTORE R/W ACCESS TO STACK AREA
20817 112066 000400 BR SMTSTS ;GO TO NEXT TEST
20818
20819
```

```

20821 ;*****
20822 ;SUPERVISOR MODE CIS STACK PROBEAHEAD MEMORY MGMT ABORT TESTS
20823 ;
20824 ;
20825 ;
20826 112070 SMTSTS:
20827 112070 005237 001120 INC $TESTN ;UPDATE TEST NUMBER FOR APT
20828 112074 012737 112434 000250 MOV #MMHDLR,@#MMVEC ;SETUP MEM MGMT INTERRUPT VECTOR
20829 112102 012737 040340 000252 MOV #040340,@#MMVEC+2
20830 112110 012701 112214 MOV #1$,R1 ;SETUP INTR RETURN ADDRESS
20831 112114 012700 112560 MOV #PDRTAB,R0
20832 112120 012737 040340 177776 MOV #040340,@#PSW ;SWITCH TO SUPERVISOR MODE
20833 112126 012706 060070 3$: MOV #60070,SP
20834 ;
20835 112132 011037 172204 2$: MOV (R0),@#SIPDR2 ;PROTECT PART OF STACK
20836 112136 012737 000001 177572 MOV #1,@#MMRO ;TURN ON MEMORY MGMT
20837 ;
20838 112144 004737 112456 JSR PC,SAVR ;SAVE REGISTERS
20839 ;
20840 112150 076130 MOVCI ;EXECUTE THE CIS INSTRUCTION
20841 112152 112446 SRC.PTR
20842 112154 112452 DST.PTR
20843 112156 000377 FILL
20844 ;
20845 112160 000240 NOP ;TO LOOP ON THIS TEST, REPLACE THE
20846 112162 000240 NOP ; TWO NOPS WITH A JMP TO 3$.
20847 112164 012737 000340 177776 MOV #340,@#PSW ;SWITCH BACK TO KERNEL MODE BEFORE HALT
20848 112172 PRINTB #NOABO ;PRINT CIS INST FAILED TO ABORT
(6) 112172 012746 013561 MOV #NOABO,-(SP)
(3) 112176 010600 MOV SP,R0
(4) 112200 004737 065410 JSR PC,FPRINT
20849 112204 012737 000001 001116 MOV #1,$FATAL
20850 112212 000000 HALT ;CIS INSTRUCTION SHOULD HAVE ABORTED BUT DIDN'T
20851 ;
20852 112214 004737 112510 1$: JSR PC,RESR ;RESTORE REGISTERS
20853 112220 062700 000002 ADD #2,R0 ;UPDATE PROTECTION SCHEME TO NEXT TABLE CASE
20854 112224 005710 TST (R0) ;ANY CASES LEFT TO TRY?
20855 112226 001341 BNE 2$ ;BRANCH IF YES
20856 ;
20857 112230 005037 177572 CLR @#MMRO ;NO - PREPARE TO EXIT TEST
20858 112234 012737 177406 172204 MOV #177406,@#SIPDR2 ;RESTORE R/W ACCESS TO STACK AREA
20859 112242 000400 BR UMTSTS ;GO TO NEXT TEST
20860
20861

```

```

20863 ;*****
20864 ;
20865 ;USER MODE CIS STACK PROBEAHEAD MEM MGMT ABORT TESTS
20866 ;
20867 ;
20868 112244 UMTSTS:
20869 112244 005237 001120 INC $TESTN ;UPDATE TEST NUMBER FOR APT
20870 112250 012737 112434 000250 MOV #MMHDLR,@#MMVEC ;SETUP MEM MGMT INTERRUPT VECTOR
20871 112256 012737 140340 000252 MOV #140340,@#MMVEC+2
20872 112264 012701 112370 MOV #1$,R1 ;SETUP INTR RETURN ADDRESS
20873 112270 012700 112560 MOV #PDRTAB,R0
20874 112274 012737 140340 177776 MCV #140340,@#PSW ;SWITCH TO USER MODE
20875 112302 012706 060070 3$: MOV #60070,SP
20876
20877 112306 011037 177604 2$: MOV (R0),@#UIPDR2 ;PROTECT PART OF STACK
20878 112312 012737 000001 177572 MOV #1,@#MMRO ;TURN ON MEMORY MGMT
20879
20880 112320 004737 112456 JSR PC,SAVR ;SAVE REGISTERS
20881
20882 112324 076130 MOVCI ;EXECUTE THE CIS INSTRUCTION
20883 112326 112446 SRC.PTR
20884 112330 112452 DST.PTR
20885 112332 000377 FILL
20886
20887 112334 000240 NOP ;TO LOOP ON THIS TEST REPLACE THE
20888 112336 000240 NOP ; TWO NOPS WITH A JMP TO 3$.
20889 112340 012737 000340 177776 MOV #340,@#PSW ;SWITCH BACK TO KERNEL MODE BEFORE HALT
20890 112346 PRINTB #NOABO ;PRINT CIS INST FAILED TO ABORT
(6) 112346 012746 013561 MOV #NOABO,-(SP)
(3) 112352 010600 MOV SP,R0
(4) 112354 004737 065410 JSR PC,FPRINT
20891 112360 012737 000001 001116 MOV #1,$FATAL ;SET APT FATAL ERROR INDICATOR
20892 112366 000000 HALT ;CIS INSTRUCTION SHOULD HAVE ABORTED BUT DIDN'T
20893
20894 112370 004737 112510 1$: JSR PC,RESR ;RESTORE REGISTERS
20895 112374 062700 000002 ADD #2,R0 ;UPDATE PROTECTION SCHEME TO NEXT TABLE CASE
20896 112400 005710 TST (R0) ;ANY CASES LEFT TO TRY?
20897 112402 001341 BNE 2$ ;BRANCH IF YES
20898
20899 112404 005037 177572 CLR @#MMRO ;NO - PREPARE TO EXIT TEST
20900 112410 012737 177406 177604 MOV #177406,@#UIPDR2 ;RESTORE R/W ACCESS TO STACK AREA
20901 112416 012737 000000 177776 MOV #0,@#PSW ;SWITCH BACK TO KERNEL MODE
20902 112424 013706 112556 MOV STK1,SP ;RESTORE NORMAL STACK POINTER
20903 112430 000137 053354 JMP DONE
20904

```

```

20906 ;MEMORY MANAGEMENT TRAP HANDLER
20907 ;
20908 112434 ;MMHDLR:
20909 112434 005037 177572 CLR @MMR0 ;TURN OFF MEM MGMT
20910 112440 005726 TST (SP)+ ;FIX UP STACK
20911 112442 005726 TST (SP)+
20912 112444 000111 JMP (R1) ;RETURN VIA R1
20913
20914
20915
20916 ;CIS INSTRUCTION SOURCE AND DESTINATION DESCRIPTORS
20917 ;
20918 112446 000001 SRC.PTR: .WORD 1
20919 112450 076000 .WORD 76000
20920 112452 000001 DST.PTR: .WORD 1
20921 112454 076001 .WORD 76001
20922
20923 ;SUBROUTINES
20924 ;
20925 112456 010037 112542 SAVR: MOV R0,SVR0 ;SAVE REGISTERS
20926 112462 010137 112544 MOV R1,SVR1
20927 112466 010237 112546 MOV R2,SVR2
20928 112472 010337 112550 MOV R3,SVR3
20929 112476 010437 112552 MOV R4,SVR4
20930 112502 010537 112554 MOV R5,SVR5
20931 112506 000207 RTS PC
20932
20933 112510 013700 112542 RESR: MOV SVR0,R0 ;RESTORE REGISTERS
20934 112514 013701 112544 MOV SVR1,R1
20935 112520 013702 112546 MOV SVR2,R2
20936 112524 013703 112550 MOV SVR3,R3
20937 112530 013704 112552 MOV SVR4,R4
20938 112534 013705 112554 MOV SVR5,R5
20939 112540 000207 RTS PC
20940
20941 112542 000000 SVR0: .WORD 0
20942 112544 000000 SVR1: .WORD 0
20943 112546 000000 SVR2: .WORD 0
20944 112550 000000 SVR3: .WORD 0
20945 112552 000000 SVR4: .WORD 0
20946 112554 000000 SVR5: .WORD 0
20947
20948 112556 000000 STK1: .WORD 0
20949
20950 ;PROTECTION TABLE (WORD FORMAT = PDR FORMAT)
20951 112560 PDRTAB:
20952 112560 177000 177000 ;ACF=00 ED=0 PLF=176
20953 112562 177410 177410 ;ACF=00 ED=1 PLF=177
20954 112564 177400 177400 ;ACF=00 ED=0 PLF=177
20955 112566 100010 100010 ;ACF=00 ED=1 PLF=0
20956
20957 112570 177002 177002 ;ACF=01 ED=0 PLF=176
20958 112572 177412 177412 ;ACF=01 ED=1 PLF=177
20959 112574 177402 177402 ;ACF=01 ED=0 PLF=177

```

20960	112576	100012	100012	:ACF=01	ED=1	PLF=0
20961						
20962	112600	177004	177004	:ACF=10	ED=0	PLF=176
20963	112602	177414	177414	:ACF=10	ED=1	PLF=177
20964	112604	177404	177404	:ACF=10	ED=0	PLF=177
20965	112606	100014	100014	:ACF=10	ED=1	PLF=0
20966						
20967	112610	177006	177006	:ACF=11	ED=0	PLF=176
20968	112612	177416	177416	:ACF=11	ED=1	PLF=177
20969	112614	000000	0			

20970  
20971  
20972  
20974 120000  
20979  
20980  
20981  
20982 120000 000200  
20983 120400 111111  
20984 120402 000100  
20985 120602  
20986  
20987 120602 111111  
20988 120604 000000  
20989  
20990  
20991  
20992 120606 000400  
20993 121206  
20994  
20995  
20996  
20997 000001

```

.=120000
.SBTTL          CIS INST EXECUTION STACK
;CIS INSTRUCTION EXECUTION STACK
;
PRECSK: .BLKW ^D128
        .WORD 111111
        .BLKW ^D64
CSTACK:                                     ;NOTE THIS STACK AREA IS INITIALIZED TO
                                           ; 055555 PRIOR TO EACH TEST.
PSTCSK: .WORD 111111
SAVKCC: .WORD 0
.SBTTL          TRANSLATION TABLES FOR MOVTC
;TRANSLATION TABLES FOR MOVTC
;
XLTBL1: .BLKB ^D256
ELTBL:
.EVEN
.END

```



AADDN	007554	8160#	15058								
AADDP	007660	8168#	15114								
AASHN	007637	8166#	15100								
AASHP	007741	8174#	15156								
ABASE =	000000	6405									
ABUFO	011446	8202#	14299								
ACASZ	067664	10969	14932#								
ACCOCT	065102	10803	10808	10813	14387	14391	14410#				
ACCSEE	016111	8271#	10802								
ACDW1 =	000000	6405									
ACDW2 =	000000	6405									
ACINST	002462	6779#	10970	14908	14912	14915	14932*	14933*	14934*	14935	14947
ACMPC	007534	8158#	15046								
ACMPN	007574	8162#	15072								
ACMPP	007700	8170#	15128								
ACOUT	011334	8196#	14273	14288							
ACPUOP=	000000	6405									
ACVTLN	007647	8167#	15107								
ACVTLP	007751	8175#	15163								
ACVTNL	007604	8163#	15079								
ACVTNP	007626	8165#	15093								
ACVTPL	007710	8171#	15135								
ACVTPN	007615	8164#	15086								
ADHDR	013755	8229#	14314								
ADDW0 =	000000	6405									
ADDW1 =	000000	6405									
ADDW10=	000000	6405									
ADDW11=	000000	6405									
ADDW12=	000000	6405									
ADDW13=	000000	6405									
ADDW14=	000000	6405									
ADDW15=	000000	6405									
ADDW2 =	000000	6405									
ADDW3 =	000000	6405									
ADDW4 =	000000	6405									
ADDW5 =	000000	6405									
ADDW6 =	000000	6405									
ADDW7 =	000000	6405									
ADDW8 =	000000	6405									
ADDW9 =	000000	6405									
ADEVCT=	000000	6405									
ADEVN =	000000	6405									
ADIQ	040272	10904	10914	10933#							
ADIVP	007731	8173#	15149								
ADJEOP	046152	11978	11984	11989	12000#						
ADJI	050344	12308	12324#								
ADJR	051264	12309	12314	12318	12322	12329	12458#				
AEADR	002174	6660#	12399*	14299							
AEDTA	002176	6661#	12401*	14299							
AENV =	000000	6405									
AENVN =	000000	6405									
AFATAL=	000000	6405									
ALOCC	007472	8154#	15022								
AL2D	007762	8176#	15170								





CONTRL=	177746	6356#												
CONVM	067506	14828	14834	14854	14858	14870#								
CPUERR=	177766	6356#												
CR =	000015	6356#	6562#	8203	8227	8259	8260	8261	8263	8268	8269	8270	8273	8278
		8279	8280	8281	8288	8293	8294	8295	8296	8297				
CRLF =	000200	6356#	14521											
CSCC	050046	12265	12267#											
CSTACK	120602	12155	12156	13799	13802	13835	13838	20985#						
CTACT	001762	6589#	14196*	15191	15217*									
CTLC	042040	11249#	11257											
DASH	016477	8285#	12441											
DBLK	066336	14635	14656	14664#										
DDISP =	177570	6356#	6575											
DECINS	002440	6767#	11047*	11271*	11367	11498	11549	12492						
DECPRT	067164	14736	14749	14762	14777	14791	14804#							
DECTTB	004264	7215	7217#											
DECTYP	004262	7215#	11143											
DECZO	022230	8844	8851	8899	8926#									
DEN	002162	6654#	10840*	10965*	13215	13323*	13327*	13332*	13335	13397	13426	13450	13473	13506
		13533	13549	15207										
DENS	001746	6583#	10763*	10773*	10818*	10862								
DESTBU	003162	6878	6879#											
DI	061174	10835*	10940*	13745#										
DIC	061152	13632	13742#											
DIGTBL	067542	14829	14835	14855	14859	14885#								
DISPLA	001740	6575#	10765*	10774*	11203*									
DISPRE	000174	6384#	10765	10774										
DIVDD	003156	6877#	13636	13940										
DIVDS	003152	6875#	13634	13635	13938	13939								
DIVPI =	076175	6358#	13633	13937										
DN	066776	14725	14729	14764#										
DONE	053354	12811	12827#	20728	20903									
DSCPIR=	152525	6382#	7638	7663	7695	7698	7716	7719	7742	14101				
DSTYPO	043474	11532	11548#											
DSTYP1	043540	11534	11561#											
DSTYP2	043576	11536	11571#											
DSTYP3	043676	11538	11592#											
DSTYP4	044574	11545	11747#	11918										
DSTYP5	045642	11546	11916#											
DST.PT	112452	20802	20842	20884	20920#									
DSWR =	177570	6356#	6574											
DTBL	066326	14637	14660#											
DVTST	037154	6386	10752	10772#	11251	12454	12889							
EADAD	024242	9209	9256#											
EADAD1	024250	9252	9255	9257#										
EADDN	023412	8445	9153#											
EADDP	023402	8461	9151#	10303	10379	10579								
EADIT	024526	9290	9309#											
EADSB1	023740	9202	9205#											
EADSUB	023472	9152	9154	9159	9161	9164#								
EADSUM	025104	9146*	9257*	9380#										
EAIS	024206	9219	9245	9250#										
EANEG	025056	9206*	9253*	9282	9369#									
EAODD	025114	9164*	9172*	9182	9227	9296	9384#							



EGTOPA	024442	9272	9293#												
EGTOPB	024474	9280	9301#												
EINST	017354	8343*	8352	8374#											
EINSTR	046136	11129*	11994#	20624*	20675*										
EIRSTK	017356	8344*	8358	8375#											
EIRT2	034224	10285	10418#	10499											
EISP	033450	10325	10336#												
EISTG	045740	6759	11559	11569	11590	11700	11745	11898	11904	11908	11912	11914	11940#		
ELISTA	017404	8348	8354	8306#											
ELISTB	017530	8355	8428#												
ELOCC	021246	8439	8725#												
ELSD	025130	8807	8830	8832	8840*	8841	8859*	8862	9015	9055	9063	9181*	9197*	9226*	
		9237*	9297*	9305*	9332*	9337*	9390#	9494*	9522*	9641*	9657*	9764*	9774*	9779*	
		9796*	9832*	9842*	9847*	9868*	9886*	9937*	9962*	9981*	9993*	10012*	10026*	10038*	
		10057*	10063*	10066*	10067*	10102*	10136*	10171*	10186*	10210*	10216*	10260*	10267*	10315*	
		10348*	14810*	14824*	14837										
ELS1M	033072	10165*	10177*	10206	10277#										
ELS2M	033074	10166*	10192*	10207	10278#										
ELTBL	121206	10984	20993#												
EL2	036222	10629#	10648	10652	10656	10661	10666								
EL2D0	036216	8428	10628#												
EL2D1	036322	8429	10646#												
EL2D2	036330	8430	10650#												
EL2D3	036336	8431	10654#												
EL2D4	036344	8432	10658#												
EL2D5	036360	8433	10663#												
EL2D6	036374	8434	10668#												
EL2D7	036440	8435	10679#												
EL2324	001710	6556#	13039												
EL3	036504	10691#	10704	10708	10712	10716	10720								
EL3D0	036500	8453	10690#												
EL3D1	036554	8454	10702#												
EL3D2	036562	8455	10706#												
EL3D3	036570	8456	10710#												
EL3D4	036576	8457	10714#												
EL3D5	036604	8458	10718#												
EL3D6	036612	8459	10722#												
EL3D7	036672	8460	10736#												
EL44	001706	6555#	13060												
EL74	001704	6554#	13041												
EMADR	002200	6662#	12400*	14298											
EMASK	025106	9028*	9035*	9036	9381#										
EMATOT	020664	8640#	8692												
EMDTA	002202	6663#	12402*	14298											
EMID	034022	10334	10346	10386#											
EMOUT	011146	8193#	14269												
EMOVC	017652	8436	8471#												
EMOVRC	020356	8437	8574#												
EMOVTC	020074	8438	8519#												
EMPTR	002142	6645#	11031*	11232*	12817*	14198	14262								
EMRBKD	020472	8595#													
EMRFIL	020520	8598	8602	8605#											
EMRFWD	020510	8594	8601#	8604											
EMSDP	030636	9784	9802	9852	9862	9883*	9894*	9898#							



ERRBUF	002150	6648#	12301*	12396*	12409	14296											
ERRCC	002144	6646#	12299*	12340*	12404	14286	14291										
ERRCT	002054	6618#	10839*	12419*	12438	14353*											
ERRREG	002146	6647#	12300*	12357*	12407	14271											
ERRS	002056	6619#	12403*	12423*	12436												
ERRSTK	002152	6649#	11218*	12292*	12411												
ERRVEC=	000004	6356#	10825*	10826*	13099*	13102*	13105*	13127*									
ERSAV	034142	10283	10408#	10496													
ERSNEG	025062	8809*	8816*	8884*	8888*	8936*	8938*	8946*	8952*	8966*	8970*	9020*	9022	9071*			
		9075	9096	9113	9187	9191	9201	9251*	9254*	9256*	9258	9336*	9340*	9347			
		9371#	9490*	9529*	9534	9645	9782	9794*	9799	9850	9867*	9870	10020*	10034*			
		10085	10229	10234	10240	10242	10317	14842									
ERT1	034420	10364	10374	10449#	10534												
ERT2	034442	10372	10376	10387	10422	10450#	10582	10585	10591*	10596*	10599						
ERT3	034464	10451#	10516	10518*	10530	10539	10542	10553	10556	10562	10574						
ERT4	034506	10452#	10525	10527*	10532	10564	10566	10576	10578								
ERT5	034530	10453#															
ERO	003670	7062#	11963*	14268	14767												
EROR	003706	7073#	12115	12122*	12343	14270	14275										
ER1	003672	7063#	11964*	14268	14768												
ER1R	003710	7074#	12116	12123*	12345	14270											
ER2	003674	7064#	11965*	14268	14772												
ER2R	003712	7075#	12117	12124*	12347	14270											
ER3	003676	7065#	11966*	14268	14773												
ER3R	003714	7076#	12118	12125*	12349	14270											
ER4	003700	7066#	11967*	14268	14775												
ER4R	003716	7077#	12119	12126*	12351	14270											
ER5	003702	7067#	11968*	14268	14776												
ER5R	003720	7078#	12120	12127*	12353	14270											
ER6	003704	7068#	11969*														
ER6R	003722	7079#	12155*	12355	14270												
ESBTD	035670	10569	10573#														
ESCC	031730	10075	10079#														
ESCF	026330	9429*	9449*	9509	9520	9557#											
ESCNC	021170	8441	8710#														
ESDC	030640	9882*	9891*	9894	9899#												
ESEED	001744	6582#	10751*	10800	10819*												
ESGN	032142	10125#															
ESGNA	025076	9377#															
ESGNB	025100	9378#															
ESGNC	025102	9379#															
ESISRC	031214	9932	9971	9974#	10001												
ESKPC	021300	8440	8736#														
ESLSD	025122	8830*	8859	9387#													
ESNK	021372	8779#	9183	9199	9231	9242	9298	9306	9643	9662	9890	9965	9983	10014			
		10029	10064	10103	10173	10188	10212	10218	10262	10269	10316	10350	14833	14839			
ESOPSW	034572	10393	10409*	10457#	10605	10609											
ESOSTK	034570	10392	10408*	10456#	10604												
ESPNC	021112	8442	8695#														
ESPND	021326	8699	8707	8714	8722	8729	8733	8740	8744#								
ESPOS	035054	10336*	10362	10383*	10474#	10498*	10544*	10549	10592*								
ESRO	034574	10297	10410*	10430	10458#	10511	10617										
ESR1	034576	10298	10411*	10431	10459#	10512	10615										
ESR2	034600	10308	10314	10315	10337	10343	10348	10412*	10439	10460#	10520						



ESR3	034602	10313	10347	10413*	10440	10461#	10521								
ESR4	034604	10390	10414*	10462#	10602										
ESR5	034606	10391	10415*	10463#	10603										
ESS1SN	035136	10491#	10504*	10594											
ESS2SN	035134	10490#	10509*	10594											
ESTORE	0173	8335*	8336*	8337*	8338*	8339*	8340*	8363	8364	8365	8366	8367	8368	8372#	
ESUBCT	035140	10492#	10559*	10570*	10587*	10588*	10589*	10590*	10591						
ESUBF	025124	9156*	9163*	9185	9362*	9388#									
ESUBN	023460	8446	9162#												
ESUBP	023420	8462	9155#	10535	10567										
ES1	025052	9184*	9189*	9208	9251	9256	9367#	9644*	9647*	9706	9719	9732	9781*	9786*	
		9791	9794	9849*	9854*	9859	9867								
ES1D	033060	10213*	10219	10272#											
ES1NSD	033062	10181*	10200	10203	10223	10273#									
ES2	025054	9200*	9203*	9208	9254	9368#									
ES2NSD	033064	10196*	10200	10274#											
ETLSD	033076	9926*	9937	9993	10037	10038	10102	10136	10279#						
ETMPRO	032130	8787*	8818	8828*	8857	9004*	9043	9049*	9059	10120#					
ETMPR1	032132	8788*	8807	8814	8819	8829*	8832	8834	8836	8858	8864	8867	8876	8891	
		8896	8903	8907	8914	9005*	9015	9017	9044	9050*	9055	9060	9065	9068	
		9073	9088	9091	9104	9110	9120	10121#							
ETMPR2	032134	9007*	9045	9052*	9061	10122#									
ETNZ1	032150	9915*	9998*	10016	10128#	10142*									
ETOPSW	034610	10287	10290*	10302	10304*	10305	10366	10368*	10369	10378	10380*	10381	10424	10433	
		10442	10464#	10501	10503*	10504	10506	10508*	10509						
ETRSTK	034554	10301	10365	10377	10423	10432	10441	10455#							
ETSTS1	034300	10286	10428#	10500											
ETSTS2	034350	10289	10437#	10505											
EVRTAB	026056	9259	9261	9263	9322	9433	9437	9467	9491	9501	9551#	9935	9991	10036	
		10047	10051	10094	10133										
EVRTAD	025416	9463	9466#												
EVTABA	026334	9469	9559#												
EVTPAS	026326	9265*	9310*	9313	9320*	9321	9456*	9457	9464*	9468	9556#				
EVTSGN	026324	9439*	9442*	9490	9555#										
EVTSSV	026332	9205*	9313*	9321*	9338	9352	9426*	9473*	9483*	9539	9558#	9748*	9768*	9771	
		9815*	9836*	9839											
EVTWRP	025532	9458	9489#												
EVTXT	025666	9511	9513	9517#											
EVXTBP	035126	10295*	10300	10306	10324	10329	10330*	10357*	10358*	10359*	10361	10487#			
EVXTVB	035124	10296*	10305*	10331*	10354*	10355*	10356*	10370	10486#						
EXLTCS	062226	13922	13924	13942#											
EXL2	036252	10635#	10677	10686	10700	10734	10746								
EXMD	034110	10398#	10607												
EXT	024560	9277	9316#												
EXTBK	063232	11256	14157#												
EXTBP	035060	10295	10359	10476#											
EXTVB	035102	10296	10356	10485#											
EXTYP	044350	11641	11642	11655	11677	11682	11684	11697	11700#						
EXT1	024740	9344#													
EXZD	036140	10547	10609#												
EZDBCK	050660	12342	12385#												
EZDBEG	036212	10615*	10623#	12389											
EZDEND	036214	10617*	10618*	10619*	10624#	12391	12392								
EZDF	036210	8341*	10497*	10612*	10622#	12333	12341	14693							





FORM1	010006	12929	12942	12956	12968										
FORM10	010662	8179#	15007	15013											
FORM11	010734	8188#	15089	15096											
FORM12	011016	8189#	15103	15159											
FORM13	011174	8190#	15110	15166											
FORM14	011254	8194#	14268												
FORM15	011362	8195#	14270												
FORM16	011371	8197#	14279												
FORM17	011375	8198#	14281												
FORM18	011405	8199#	14289												
FORM19	013716	8200#	14293												
FORM2	010066	8228#	14317												
FORM20	014041	8180#	15019												
FORM21	014074	8230#	14318												
FORM22	014077	8231#	14197	15214	15216										
FORM23	014114	8232#	14733												
FORM24	014132	8233#	14746	14757											
FORM25	014150	8234#	14744	14759											
FORM26	014173	8235#	14764												
FORM27	014216	8236#	14778												
FORM3	010154	8237#	14816												
FORM30	014222	8181#	15025	15031											
FORM31	014252	8238#	15190												
FORM32	014256	8239#	15189												
FORM33	014276	8240#	14847												
FORM34	014302	8241#	14860												
FORM35	014307	8242#	14844												
FORM36	014314	8243#	14846												
FORM37	014376	8244#	11058	12428											
FORM38	014415	8245#	12438												
FORM39	014477	8246#	13209												
FORM4	010222	8247#	13210												
FORM40	014567	8182#	15037	15043											
FORM41	014647	8248#	15193												
FORM42	014745	8250#	15173	15179											
FORM43	015034	8251#	13624	13930											
FORM44	015115	8252#	13625	13931											
FORM45	015170	8253#	13673												
FORM46	015247	8254#	13674												
FORM47	015320	8255#	13675												
FORM48	015401	8256#	13707												
FORM49	015457	8257#	13696												
FORM5	010276	8258#	13697												
FORM6	010362	8183#	15049												
FORM7	010434	8184#	15055												
FORM8	010530	8185#	15061	15068	15117	15124	15145	15152							
FORM9	010606	8186#	15075	15131											
FPRINT	065410	8187#	15082	15138											
		10789	10802	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968	
		11058	12229	12289	12291	12428	12438	12440	12441	13206	13209	13210	13215	13505	
		13506	13528	13532	13533	13548	13549	13621	13624	13625	13655	13658	13670	13673	
		13674	13675	13693	13696	13697	13704	13707	13724	13727	13927	13930	13931	14197	
		14265	14267	14268	14269	14270	14273	14279	14281	14288	14289	14293	14298	14299	
		14314	14317	14318	14480#	14733	14744	14746	14757	14759	14764	14778	14816	14844	























PSW	= 177776	6356#	12275	13759	20832*	20847*	20874*	20889*	20901*					
PTO	045114	11809#	11843											
PTP	001566	6491#	11100	11106	12709	12910	12918							
PTPTR	002136	6643#	11266	11321	11325	11348	11371	11463	11482	11494	11504	11510	12911*	12919*
		13026												
PTP01	001570	6493#	11083	11114										
PTP02	001572	6494#												
PTP03	001574	6495#												
PTP04	001576	6496#												
PTP05	001600	6497#												
PTP06	001602	6498#												
PTP07	001604	6499#												
PTP10	001606	6500#												
PTP11	001610	6501#												
PTP12	001612	6502#												
PTP13	001614	6503#	20074	20084	20197	20205								
PTP14	001616	6504#	19457	19463	19529	19535	19584	19592	19992	20000	20375	20385	20433	20442
PTP15	001620	6505#												
PTP16	001622	6506#												
PTP17	001624	6507#												
PTP20	001626	6508#	19705	19711	19722	19733	19739	19747	19757	19771	19777	19798	19802	19810
		19814	19820	19828										
PTP21	001630	6509#												
PTP22	001632	6510#												
PTP23	001634	6511#												
PTP24	001636	6512#	11086	11120										
PTQV	002212	6667#	12805	13051*	13069*	13123*								
PTW1	002130	6640#	12715*	12717*	12718	12720	12730*	12731*	12732	12735*	12736*	12738*	12739	12766
PTYPLO	045254	6761	11841#											
PTYPLS	045300	6763	11850#											
PTYPSZ	045244	6758	11837#											
PTYPTA	002416	6758#	11802											
PTYPI0	045106	6760	11805#											
PTYPTS	045264	6762	11845#											
PTY3P	044066	11594	11631#											
PT34	002154	6650#	13124*	13231	13330	13352	13519	20730						
PWRVEC=	000024	6356#	10765*	10774*	20692*									
PZCODE	002472	6783#	11045*	14699	14703	14706	14709	14713	14716	14719	14722	14726	14765	14770
		14779	14784	15062*	15069*	15076*	15083*	15090*	15097*	15104*	15111*	15118*	15125*	15132*
		15139*	15146*	15153*	15160*	15167*								
QDISP	011477	8203#	14304											
QRYFLG	002204	6664#	12420*	14176*	14189*	14301								
QUES	015552	8259#	14395	14454										
QVHDR	016032	8270#	10997											
QVMODE	002206	6665#	10757*	10772*	10994	12802	12832	12867	12871*					
QVST	036766	6387	10757#	12873										
RANDOM	001760	6588#	10846*	10952*	11049	11165	11193	11378	11421	11441	12426	12488	12692	12784
		12885*	13304	13466	14917									
RANDSC	002510	6791#	14104											
RANDTA	001770	6592#	11544*	11771	11786	11818	11875	11897	11917*					
RANGE	065000	14312	14386#	14396										
RCHAR	064776	13559*	13560*	13562	13564	13566	14160*	14165*	14166*	14167	14169	14173	14178	14182
		14186	14336*	14337	14339*	14340*	14341	14343	14349	14356	14358	14360	14362	14375
		14379#	14436*	14457*	14938*	14939	14941*	14942*	14943	14945	14949*	20690*		









R10	=X000000	20926	20934*											
R11	=X000001	6356#												
R12	=X000002	6356#												
R13	=X000003	6356#												
R14	=X000004	6356#												
R15	=X000005	6356#												
R2	=X000002	6356#	8337	8365*	8478*	8482*	8487*	8488*	8494	8498	8503	8505*	8526*	8530*
		8535*	8536*	8542	8549	8557	8559*	8581*	8585*	8590*	8591*	8593	8596	8601
		8606	8626*	8628	8632	8634	8637*	8639	8642	8658	8660	8662	8664*	8672*
		8674	8677	8679	8684*	8686*	8689	8703*	8704	8718*	8719	8745*	8746*	8747*
		8748*	8749*	8750*	8751*	8800*	8802*	8803	8806*	8810	8812	8817*	8821*	8822*
		8823*	8824*	8855*	8856*	8879*	8880	8881*	8882	8885*	8889*	8900*	8928*	8929
		8930*	8931	8933*	8934	8939*	8941	8943*	8944	8947*	8948*	8950	8953*	8954*
		8956	8962*	8963*	8964	8967*	8968*	9007	9014*	9019*	9024*	9026*	9027	9045*
		9052	9053*	9054*	9058	9061*	9077*	9079*	9082*	9083*	9084*	9085*	9086	9094*
		9095*	9098*	9099	9102*	9232	9243	9273	9281	9327*	9328	9355*	9356*	9357*
		9358*	9359*	9360*	9361*	9434*	9440	9444*	9447	9454*	9455*	9460*	9500*	9503*
		9504	9526*	9542*	9543*	9544*	9545*	9546*	9547*	9548*	9664	9691	9758*	9769
		9805*	9806*	9807*	9808*	9809*	9810*	9811*	9826*	9837	9873*	9874*	9875*	9876*
		9877*	9878*	9879*	9892	9938*	9967*	9968	9984*	9985	9987*	9996*	10030	10071
		10104	10111*	10112*	10113*	10114*	10115*	10116*	10117*	10135*	10140	10174	10189	10213
		10219	10249*	10250*	10251*	10252*	10253*	10254*	10255*	10351	10353*	10354	10357	10398*
		10401*	10402*	10403*	10404*	10629*	10630	10631	10632*	10633	10634	10670*	10671	10672
		10673*	10674	10675	10691*	10692	10693	10694*	10695	10696	10697*	10698	10699	10724*
		10725	10726	10727*	10728	10729	10730*	10731	10732	10981*	10985*	10986	11083*	11084*
		11086	11114*	11115*	11116	11118*	11119*	11120	11267*	11282*	11294*	11306*	11371*	11372*
		11376*	11380	11382	11428*	11431	11432	11613*	11614*	11615	11625*	11626*	11627*	11628
		11650*	11652*	11653*	11654	11659*	11663*	11665*	11800*	11801*	11802*	11803	11950*	11951*
		12028*	12030*	12031*	12032	12033*	12037*	12039*	12047*	12048*	12050*	12052	12053*	12054
		12056*	12060*	12072*	12075*	12078*	12081*	12083*	12133*	12135*	12136	12191*	12192	12271
		12361*	12373*	12381	12386*	12389	12391*	12394	12398*	12400	12402	12722*	12723*	12724
		12734*	12742*	12746*	12747*	12748	12768*	12769	12773	13151*	13166*	13193	13210	13755
		13773*	13801*	13803*	13837*	13839	14093*	14104*	14116*	14120*	14122*	14123	14275*	14276
		14484	14487*	14488	14490	14492	14494	14496	14498	14500	14502	14504	14508*	14509*
		14513*	14514	14516	14518	14524*	14525*	14529*	14530*	14531	14533*	14537*	14539*	14544*
		14545*	14546*	14549*	14552*	14559*	14563*	14564*	14575*	14576*	14577	14607*	14608*	14609
		14633	14636*	14640*	14643*	14648	14653*	14812*	14813*	14815*	14817*	14827*	14848*	14849*
		14850*	14851*	14852*	14853*	14856*	14857*	14873	19482*	19483*	19484	19494*	19495*	19496
		19498	19511*	19512*	19513	19519*	19520*	19521	19554*	19555*	19556	19566*	19567*	19568
		19570	19751*	19752*	19753	19824*	19825	20398*	20399*	20400	20462*	20463*	20464	20615*
		20616	20619*	20621*	20623*	20662*	20663*	20664*	20665	20666*	20667	20670*	20671	20688*
		20692*	20927	20935*										
R3	=X000003	6356#	8338	8366*	8483*	8484*	8485	8492*	8496	8498*	8501*	8531*	8532*	8533
		8540*	8544	8549*	8555*	8586*	8587*	8588	8595*	8599	8605*	8606	8608*	8627*
		8630	8632	8634	8636*	8638	8644	8648	8652	8654	8656*	8673*	8675	8679
		8681*	8687*	9210*	9213*	9216	9217	9263*	9314*	9319*	9322*	9327	9435*	9443*
		9445	9452*	9461*	9632*	9667*	9670	9674*	9679*	9685*	9688*	9693*	9699	9704
		9715	9721*	9726*	9727	9738	9917*	9918*	9946*	9948*	9949*	9952*	9953	9956
		9958	9961	9977	9980	9999*	10055*	10067	10069	10161*	10162*	10165	10168	10169
		10176*	10177	10180	10206*	10209	10222*	10223	12034*	12037	12040*	12061*	12073*	12076*
		12079*	12082*	12084*	12090*	12098*	12100*	12101*	12193*	12194	12272	12708*	12709*	12748
		12751*	12756*	12757*	12773	12775	12778*	13152*	13167*	13195	13210	13756	13774*	14483
		14510*	14526*	14534*	14558*	14565*	14566*	14579*	14581*	14582*	14583*	14584	14611*	14638*



SGNBYT	025166	8803*	8873*	8880*	8894*	8910*	8917*	8929*	9416#	14848	14856			
SGPRO	003004	6829#	13753*	13771	13787	13819								
SGPRO6	061206	13608	13752#	13919										
SGPR1	003006	6830#	13754*	13772	13788	13823								
SGPR2	003010	6831#	13755*	13773	13789	13825								
SGPR3	003012	6832#	13756*	13774	13790	13827								
SGPR4	003014	6833#	13757*	13775	13791	13829								
SGPR5	003016	6834#	13758*	13776	13792	13831								
SGPR6	003020	6835#	13762*	13764*	13794	13795	13817	13833						
SIGN	002124	6638#	11605*	11606*	11636*	11637*	11638*	11639*	11640	11645	11674	11712*	11728*	11743*
		11744	11775*	11776*	11777*	11778*	11779*	11780*	11823*	11825*	11834*			
SIPARO=	172240	6356#	13103	13244*	20744*									
SIPAR1=	172242	6356#	13245*	20745*										
SIPAR2=	172244	6356#	13246*	20746*										
SIPAR3=	172246	6356#	13247*	20747*										
SIPAR4=	172250	6356#	13248*	20748*										
SIPAR5=	172252	6356#	13249*	20749*										
SIPAR6=	172254	6356#	13250*	20750*										
SIPAR7=	172256	6356#	13251*	20751*										
SIPDR0=	172200	6356#	13356	20771										
SIPDR1=	172202	6356#	13479*											
SIPDR2=	172204	6356#	13421*	20835*	20858*									
SIPDR3=	172206	6356#												
SIPDR4=	172210	6356#												
SIPDR5=	172212	6356#	13430*											
SIPDR6=	172214	6356#												
SIPDR7=	172216	6356#	20773											
SIZEHI=	177762	6356#												
SIZELO=	177760	6356#												
SIZEPT	054252	10827	13034#											
SL	= 000057	6564#	8261											
SLCRLF	015571	8261#	14429											
SMTS1S	112070	20817	20826#											
SN	066566	14705	14715	14733#										
SP	=X000006	6356#	6406*	8342	8347*	10765*	10774*	10789*	10798	10799	10802*	10807	10812	10817
		10877*	10882*	10884*	10891*	10930*	10932*	10933*	10946*	10955*	10958*	10968*	11058*	12148
		12151*	12156*	12157	12199	12216*	12218*	12228*	12229*	12230	12232	12277	12278	12280
		12281*	12289*	12291*	12428*	12438*	12440*	12441*	13097	13098	13108	13109	13111	13112
		13114	13115	13206*	13209*	13210*	13215*	13501	13503	13505*	13506*	13508	13509	13510
		13522	13526	13528*	13530	13532*	13533*	13536	13537	13538	13546	13548*	13549*	13574*
		13609	13611	13619	13621*	13624*	13625*	13655*	13658*	13667	13670*	13673*	13674*	13675*
		13677*	13683	13684	13690	13693*	13696*	13697*	13701	13704*	13707*	13710*	13724*	13727*
		13761	13762	13764	13786	13814	13844*	13851*	13853*	13867*	13869	13870	13872	13873
		13883*	13885*	13901*	13903	13904	13906	13907	13914*	13917	13921	13925	13927*	13930*
		13931*	13960	13961	13977	13978	13986*	13988*	13995*	13997	13998	14000	14001	14190*
		14197*	14265*	14267*	14268*	14269*	14270*	14273*	14279*	14281*	14288*	14289*	14293*	14298*
		14299*	14314*	14317*	14318*	14326*	14327*	14366*	14367*	14368*	14390	14397	14400	14404*
		14411*	14414*	14415*	14416	14420*	14422	14424	14426	14430*	14431*	14432*	14433*	14434*
		14436	14439	14441	14443*	14444*	14445*	14446*	14447*	14450*	14451*	14455*	14457	14459*
		14473*	14554	14555*	14630*	14631*	14657	14658	14733*	14744*	14746*	14757*	14759*	14764*
		14778*	14816*	14844*	14846*	14847*	14860*	14871*	14875*	14876	14925*	15004*	15007*	15010*
		15013*	15016*	15019*	15022*	15025*	15028*	15031*	15034*	15037*	15040*	15043*	15046*	15049*
		15052*	15055*	15058*	15061*	15065*	15068*	15072*	15075*	15079*	15082*	15086*	15089*	15093*
		15096*	15100*	15103*	15107*	15110*	15114*	15117*	15121*	15124*	15128*	15131*	15135*	15138*

		15142*	15145*	15149*	15152*	15156*	15159*	15163*	15166*	15170*	15173*	15176*	15179*	15189*
		15190*	15193*	15194*	15206*	15214*	15216*	19505*	19577*	19761*	19832*	20404*	20478*	20480*
		20688*	20689*	20690*	20691*	20692*	20733	20793*	20806*	20833*	20848*	20875*	20890*	20902*
		20910	20911											
SPCV	002036	6611#	13638	13742*										
SPHAND	002140	6644#	11214*	11340	11360	11450	11523	12358	12486	13026*	14696			
SRC.PT	112446	20801	20841	20883	20918#									
SRNGST	062452	11051	14007#											
SRNGSV	062666	11419	14052#											
SRNGSW	062616	11131	14040#											
SRNGSX	062476	11461	14014#											
SRNGSY	062546	11204	14027#											
SRO	= 177572	6356#												
SR1	= 177574	6356#												
SR2	= 177576	6356#												
SR3	= 172516	6356#												
SSP	=X000006	6356#												
SSTG1	003244	6886#	19961											
SSTG10	003344	6929#	20552											
SSTG11	003352	6933#	20572											
SSTG12	003364	6939#	19839	20125	20135	20178	20411							
SSTG13	003444	6966#	20253	20264	20277									
SSTG14	003604	7017#	20140											
SSTG15	003612	7021#	20143											
SSTG16	003620	7025#	20148											
SSTG2	003250	6876	6891#	20546										
SSTG2A	003256	6895#	20554											
SSTG2B	003264	6899#	20556											
SSTG3	003272	6903#	20566											
SSTG4	003304	6909#	20548											
SSTG5	003312	6913#	20568											
SSTG6	003324	6919#	20550											
SSTG7	003332	6923#	20570											
STACK	= 001100	6356#	10765	10774										
START	036760	6384	10754#	12449	12861									
STATCG	061434	13616	13813#	13923										
STATPS	002564	6818#	12219*	13786*	13814									
STATRO	002566	6819#	12188*	13787*	13819									
STATR1	002570	6820#	12190*	13788*	13823									
STATR2	002572	6821#	12192*	13789*	13825									
STATR3	002574	6822#	12194*	13790*	13827									
STATR4	002576	6823#	12196*	13791*	13829									
STATR5	002600	6824#	12198*	13792*	13831									
STATR6	002602	6825#	12199*	12202*	12216	13794*	13817							
STGAD	002114	6634#	11527*	11552	11562	11572	11619	11669	11687	11698	11724	11763	11842	11867
		11894												
STGDIG	002306	6697#	11615*	11653	11810*									
STGDS1	002106	6631#	11519*	11528	11529*	11555	11566	11574	11576	11582*	11584*	11593	11605	11607*
		11608*	11609*	11610*	11611*	11613	11621	11712	11713	11719*	11720*	11735	11762*	11767
		11768	11793*	11795*	11797	11868*	11869	11871	11882*	11884*	11886			
STGDS2	002110	6632#	11520*	11565	11566*	11577	11580	11581*	11585*	11721*	11733	11766	11768*	11770*
		11775	11791	11792*	11796*	11797*	11870	11871*	11880	11881*	11885*	11886*		
STGLN	002112	6633#	11522*	11525*	11551*	11553	11557*	11563	11567*	11578	11587*	11596	11604*	11616
		11622*	11623	11703	11711*	11726	11730	11736*	11737*	11738	11753	11761*	11764	11781









TER4R	002262	6687#	12119*											
TER5R	002264	6688#	12120*											
TER6R	002266	6689#												
TE0	025134	9167*	9297	9392#										
TE2	025136	9168*	9305	9393#										
TE4	025140	9169*	9332	9335	9337	9352	9360	9394#						
TFORE	050754	12359	12380	12388	12403#									
TFS	044320	11617	11692#	11783										
THT =	000011	20690#												
TIMOUT	002536	6805#	13850	13871	13882	13905	13985	13999						
TINRET	050030	12168*	12173*	12256	12262#	13501	13522							
TINST	050012	10680	10683	10737	10740	10743	11033*	11130*	11258	11313	11338	11358	11389	11416
		11945	11959	12018	12020	12022	12037*	12047*	12050*	12051	12056*	12068	12070	12141*
		12142*	12143*	12144*	12145*	12146*	12149	12153	12159	12255#	12307	12368	12370	12465
		12480	12494	12690	12698*	12701*	12819*	13215	13469	13471	13506	13526	13533	13549
		13609	13688	13921	14263	14738	14751	15170	15176	15185	20613*	20624	20628*	20633*
		20634*	20639*	20644*	20645*	20650*	20655*	20656*	20661*	20665*	20667*	20668	20671*	20672*
		20673*	20674*	20675	20677*	20678*								
		6559#	13559	14165	14336	14414	14450	14938	20690					
TKB	001712	6547#	13557	14163	14334	14412	14448	14936	20690					
TKS	001666	6356#												
TKVEC =	000060													
TL1C	103074	15325	15445	15557	15686	15807	15930	16040	16172	16294	19413#			
TL2C	103114	15327	15559	15688	15809	16042	19423#							
TL21C	105262	15598	19899#											
TL22C	105300	15600	19908#											
TNXP	053116	12714	12719	12721	12758#									
TOFMM	050040	12183*	12186*	12265#										
TOLTC	047720	10836*	12224	12234	12237#	12366	13993*							
TOMM	047774	12182*	12185*	12252#										
TOPC1	047762	10838*	12246	12248#	12364	13865*								
TOPC2	047752	10837*	12237	12246#	13685	13899*								
TOTTC	001420	6414#	10977*	11053	11199*	11203	11330	11333*	11408	11411*	12233	12828*	13142	13215
		13506	13524	13533	13549	15190								
TOTTCH	001416	6413#	10978*	11201*	11332*	11410*	12829*	13215	13506	13533	13549	15190		
TPB	001716	6561#	14375*											
TPERP	001652	6524#	11195											
TPRECS	002040	6612#	13639	13743*	13936*	13941								
TPS	001714	6560#	14373											
TPSW	001664	6546#	10783*	12205	12215*	12217*	12254*	12264	13146*	13155*	13161*	13170*	13188*	13202*
		13204*												
TPVEC =	000064	6356#												
TP19	106122	16460	16873	17238	17241	17712	17715	18137	18140	18442	18822	18825	18862	18865
		19161	19164	20133#										
TP19A	106132	16908	20138#											
TP19B	106150	18943	18946	20146#										
TP19C	106160	18902	20151#											
TP19D	106172	18905	20157#											
TP19E	106200	19241	19244	20161#										
TP99	106206	18481	20165#											
TRA	002102	6629#	11460*	11465	11501	11507								
TRAPVE=	000034	6356#	10765*	10774*										
TRAP10	012150	8209#	13532											
TRAP4	012132	8208#	13505											
TRL	002104	6630#	11456*	11464	11471*	11483	11497*	11500*	11501*	11502	11507*	11508*		



TYPLO	044234	6702	11668#	11815														
TYPLS	044310	6704	11686#	11695	11853													
TYPSET	051710	12532	12537	12553#	12582	12601	12616	12657	12688									
TYPSP	002332	6705	6706	6707#	6764	6765												
TYPSZ	044072	6699	11633#	11839														
TYPTAB	002312	6699#	11627															
TYPTO	044126	6701	11644#	11571	11816													
TYPTS	044244	6703	11673#	11690	11699	11848												
TYPUZ	044124	6700	11642#															
TYP3P	044354	11631	11702#															
TYP3Z	043706	11595#																
TYP4P	045314	11750	11855#															
TYP4Z	044610	11749	11752#															
TZ19	106462	16539	16576	17022	17453	17456	17493	17496	17929	17932	18355	18358	18568	18604				
TSARGC=	000001	20261#																
		10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#				
		11058#	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#				
		13506#	13528#	13532#	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#				
		13674#	13675#	13693#	13696#	13697#	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#				
		14265#	14267#	14268#	14269#	14270#	14273#	14279#	14281#	14288#	14289#	14293#	14298#	14299#				
		14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#	14778#	14816#	14844#	14846#				
		14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#	15028#	15031#				
		15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#				
		15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#				
		15121#	15124#	15128#	15131#	15135#	15138#	15142#	15145#	15149#	15152#	15156#	15159#	15163#				
		15166#	15170#	15173#	15176#	15179#	15189#	15190#	15193#	15206#	15214#	15216#	20806#	20848#				
TO	102560	20890#																
		15338	15374	15375	15376	15377	15378	15451	15486	15487	15563	15569	15610	15650				
		15651	15653	15654	15699	15736	15737	15738	15739	15740	15820	15857	15858	15859				
		15860	15861	15936	15970	15971	16046	16052	16090	16092	16093	16095	16096	16184				
		16221	16222	16224	16225	16306	16343	16344	16346	16347	16427	16461	16462	16463				
		16464	16465	16506	16540	16541	16542	16543	16544	16577	16578	16579	16580	16581				
		16583	16616	16650	16651	16652	16686	16687	16688	16689	16726	16759	16760	16761				
		16798	16837	16874	16875	16909	16910	16912	16949	16988	17023	17024	17056	17058				
		17061	17067	17073	17097	17102	17108	17114	17143	17146	17152	17158	17190	17196				
		17202	17237	17239	17240	17242	17243	17245	17246	17247	17273	17275	17278	17284				
		17287	17288	17290	17314	17319	17325	17331	17359	17362	17368	17374	17405	17411				
		17414	17415	17417	17452	17454	17455	17457	17458	17460	17461	17462	17492	17494				
		17495	17497	17498	17500	17501	17502	17504	17528	17530	17533	17539	17545	17571				
		17576	17582	17588	17617	17620	17626	17632	17664	17670	17676	17711	17713	17714				
		17716	17717	17719	17720	17721	17747	17749	17752	17758	17761	17762	17764	17788				
		17793	17799	17805	17835	17838	17844	17850	17881	17887	17890	17891	17893	17928				
		17930	17931	17933	17934	17936	17937	17938	17964	17966	17971	17976	18004	18011				
		18016	18048	18053	18058	18095	18100	18138	18139	18141	18142	18171	18173	18178				
		18183	18213	18220	18225	18260	18265	18270	18310	18315	18356	18357	18359	18360				
		18396	18401	18443	18444	18445	18446	18447	18482	18483	18484	18485	18486	18488				
		18522	18527	18569	18570	18571	18572	18573	18605	18606	18607	18608	18609	18611				
		18638	18640	18643	18649	18655	18681	18686	18692	18698	18727	18730	18736	18742				
		18774	18780	18786	18821	18823	18824	18826	18827	18829	18830	18831	18861	18863				
		18864	18866	18867	18869	18870	18871	18901	18903	18904	18906	18907	18909	18910				
		18911	18942	18944	18945	18947	18948	18950	18951	18952	18953	18954	18978	18980				
		18983	18989	18995	19019	19024	19030	19036	19066	19069	19075	19081	19113	19119				
		19125	19160	19162	19163	19165	19166	19168	19169	19170	19200	19202	19203	19205				
		19206	19208	19209	19210	19240	19242	19243	19245	19246	19248	19249	19250	19284#				

P  
C

B  
B

B  
C  
C  
C

T1A	102604	15368	15370	15482	15644	15646	15730	15732	15851	15853	15966	16086	16088	16215
		16337	16456	16458	16535	16537	16648	16757	16871	17020	17232	17234	17236	17447
		17449	17451	17706	17708	17710	17923	17925	17927	18132	18134	18350	18352	18437
		18440	18563	18566	18816	18818	18820	19155	19157	19159	19300#			
T1B	102630	16572	16574	19311#										
T1C	102642	17487	17489	19317#										
T1D	102654	18599	19323#											
T1E	102664	18602	19328#											
T1F	102672	18476	19332#											
T1G	102706	19195	19197	19199	19339#									
T1H	102714	18856	18858	19343#										
T1K	102730	18479	19350#											
T1L	102742	18860	18900	19356#										
T1M	102750	18937	18939	19360#										
T1N	102756	18896	19364#											
T1P	102764	17189	17404	17663	17880	18773	19112	19368#						
T1Q	103004	18898	19377#											
T1R	103014	19235	19239	19382#										
T1S	103022	19237	19386#											
T10	104126	15332	15450	15564	15605	15693	15814	15935	16047	16182	16304	16422	16501	16796
		16835	16947	16986	17064	17105	17149	17193	17281	17322	17365	17408	17536	17579
		17623	17667	17755	17796	17841	17884	17974	18014	18056	18098	18181	18223	18268
		18313	18646	18689	18733	18777	18986	19027	19072	19116	19637#			
T11	104142	15333	15694	15815	16423	16502	16613	16723	17069	17110	17154	17198	17286	17327
		17370	17413	17541	17584	17628	17672	17760	17801	17846	17889	18397	18523	18651
		18694	18738	18782	18991	19032	19077	19121	19645#					
T111	103034	18389	18515	19393#										
T112	103054	18392	18518	19403#										
T12	104160	15334	15566	15607	15695	15816	16049	16178	16300	16424	16503	16614	16724	17070
		17111	17155	17199	17328	17371	17542	17585	17629	17673	17802	17847	18398	18524
		18652	18695	18739	18783	18992	19033	19078	19122	19654#				
T13	104176	15335	15567	15608	15696	15817	16050	16179	16301	16425	16504	16615	16725	17071
		17112	17156	17200	17329	17372	17543	17586	17630	17674	17803	17848	18399	18525
		18653	18696	18740	18784	18941	18993	19034	19079	19123	19663#			
T14	104212	15336	15379	15568	15609	15655	15697	15741	15818	15862	16051	16097	16183	16226
		16305	16348	16426	16466	16505	16545	16582	16684	17072	17113	17157	17248	17330
		17373	17463	17491	17503	17544	17587	17631	17722	17804	17849	17939	18143	18182
		18224	18269	18314	18361	18448	18487	18574	18610	18654	18697	18741	18785	18832
		18872	18912	18994	19035	19080	19124	19171	19211	19251	19671#			
T14A	104224	17201	17675	19678#										
T15	104236	15447	19685#											
T15A	104256	15484	19695#											
T16	104402	17059	17100	17144	17188	17531	17574	17618	17662	19727#				
T16A	104264	17235	17709	18819	18859	18899	19158	19198	19238	19699#				
T16B	104342	18940	19716#											
T16Z	104754	17276	17317	17360	17403	17750	17791	17836	17879	19806#				
T16ZA	104706	17450	17490	17926	19794#									
T161	104564	18641	18684	18728	18772	18981	19022	19067	19111	19765#				
T17	105116	17106	17147	17191	17194	17580	17621	17665	17668	18690	18731	18775	18778	19028
		19070	19114	19117	19836#									
T17A	105156	19201	19204	19857#										
T2	103134	15326	15446	15558	15599	15687	15808	15931	16041	16173	16295	16417	16496	16612
		16685	16722	16793	16832	16907	16944	16983	17057	17098	17142	17186	17274	17315
		17358	17401	17529	17572	17616	17660	17748	17789	17834	17877	17965	18005	18047







XOCHAR	111202	14158	14160	14161*	20690#*								
XSCANC	007134	6441	8019#										
XSKPC	006776	6440	7957#										
XSPAN	007134	6442	8020#										
XSUBN	007044	6446	7984#										
XSUBP	007044	6454	7983#										
XT1	102564	17060	17101	17145	17277	17318	17361	17532	17575	17619	17751	17792	17837 18642
		18685	18729	18982	19023	19068	19290#						
YADDN	070536	14980	15057#										
YADDP	071156	14988	15113#										
YASHN	071052	14986	15099#										
YASHP	071472	14994	15155#										
YCMPC	070446	14978	15045#										
YCMPN	070642	14982	15071#										
YCMPP	071262	14990	15127#										
YCVTLN	071114	14987	15106#										
YCVTLP	071534	14995	15162#										
YCVTNL	070704	14983	15078#										
YCVTNP	071010	14985	15092#										
YCVTPL	071324	14991	15134#										
YCVTPN	070746	14984	15085#										
YDIVP	071430	14993	15148#										
YLOCC	070266	14974	15021#										
YL2D	071576	11031	14996	15169#									
YL3D	071636	14997	15175#										
YMATCH	070502	14979	15051#										
YMOV	070142	14971	15003#										
YMOVRC	070176	14972	15009#										
YMOVTC	070232	14973	15015#										
YMULP	071366	14992	15141#										
YORN	064546	10885	10892	10935	10947	10959	14305	14333#					
YSCANC	070356	14976	15033#										
YSKPL	070322	14975	15027#										
YSPAN	070412	14977	15039#										
YSUBN	070600	14981	15064#										
YSUBP	071220	14989	15120#										
ZCCR	002034	6610#	14960*	14962*	14963*	14964							
ZMSK	001766	6591#	11929										
ZONED	051714	12499	12557#										
ZPM	002460	6778#	11164*	11748	12496	12498	12500						
SAPTHD	001100	6400#											
SASTAT=	***** U	6406											
SATYC	001174	6406#											
SATY1	001150	6406#											
SATY3	001156	6406#	20690										
SATY4	001166	6406#											
SCHARC	111176	20690#*											
SCMTAG=	***** U	10765	10774										
SCPUOP	001142	6405#											
SCRLF	110457	20688#	20689	20690									
SDEVCT	001124	6405#											
SENV	001134	6405#	6406	20690									
SENVN	001135	6405#	6406	10765	10774	20690							
SETABL	001134	6405#											

P  
C  
  
R  
R  
R  
R  
  
S  
S  
  
S  
T  
T







COMEN	1443#	6356#																		
ENDCOM	1455#	6356#																		
ERROR	6356#																			
ESCAPE	1566#	6356#																		
GETPRI	1217#																			
MULT	3962#	6356#																		
M&SCOUN	6334#	10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#	11058#					
	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#	13506#	13528#	13532#					
	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#	13674#	13675#	13693#	13696#	13697#					
	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#	14265#	14267#	14268#	14269#	14270#	14273#	14279#					
	14281#	14288#	14289#	14293#	14298#	14299#	14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#					
	14778#	14816#	14844#	14846#	14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#					
	15028#	15031#	15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#					
	15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#	15121#	15124#					
	15128#	15131#	15135#	15138#	15142#	15145#	15149#	15152#	15156#	15159#	15163#	15166#	15170#	15173#	15176#					
	15179#	15189#	15190#	15193#	15206#	15214#	15216#	20806#	20848#	20890#										
M&SGNIN	6321#	10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#	11058#					
	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#	13506#	13528#	13532#					
	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#	13674#	13675#	13693#	13696#	13697#					
	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#	14265#	14267#	14268#	14269#	14270#	14273#	14279#					
	14281#	14288#	14289#	14293#	14298#	14299#	14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#					
	14778#	14816#	14844#	14846#	14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#					
	15028#	15031#	15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#					
	15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#	15121#	15124#					
	15128#	15131#	15135#	15138#	15142#	15145#	15149#	15152#	15156#	15159#	15163#	15166#	15170#	15173#	15176#					
	15179#	15189#	15190#	15193#	15206#	15214#	15216#	20806#	20848#	20890#										
M&SPRC	6291#	10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#	11058#					
	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#	13506#	13528#	13532#					
	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#	13674#	13675#	13693#	13696#	13697#					
	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#	14265#	14267#	14268#	14269#	14270#	14273#	14279#					
	14281#	14288#	14289#	14293#	14298#	14299#	14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#					
	14778#	14816#	14844#	14846#	14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#					
	15028#	15031#	15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#					
	15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#	15121#	15124#					
	15128#	15131#	15135#	15138#	15142#	15145#	15149#	15152#	15156#	15159#	15163#	15166#	15170#	15173#	15176#					
	15179#	15189#	15190#	15193#	15206#	15214#	15216#	20806#	20848#	20890#										
M&SPRIN	6279#	10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#	11058#					
	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#	13506#	13528#	13532#					
	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#	13674#	13675#	13693#	13696#	13697#					
	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#	14265#	14267#	14268#	14269#	14270#	14273#	14279#					
	14281#	14288#	14289#	14293#	14298#	14299#	14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#					
	14778#	14816#	14844#	14846#	14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#					
	15028#	15031#	15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#					
	15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#	15121#	15124#					
	15128#	15131#	15135#	15138#	15142#	15145#	15149#	15152#	15156#	15159#	15163#	15166#	15170#	15173#	15176#					
	15179#	15189#	15190#	15193#	15206#	15214#	15216#	20806#	20848#	20890#										
M&SPUT	6296#	10789#	10802#	10877#	10882#	10884#	10891#	10930#	10932#	10933#	10946#	10955#	10958#	10968#	11058#					
	12229#	12289#	12291#	12428#	12438#	12440#	12441#	13206#	13209#	13210#	13215#	13505#	13506#	13528#	13532#					
	13533#	13548#	13549#	13621#	13624#	13625#	13655#	13658#	13670#	13673#	13674#	13675#	13693#	13696#	13697#					
	13704#	13707#	13724#	13727#	13927#	13930#	13931#	14197#	14265#	14267#	14268#	14269#	14270#	14273#	14279#					
	14281#	14288#	14289#	14293#	14298#	14299#	14314#	14317#	14318#	14733#	14744#	14746#	14757#	14759#	14764#					
	14778#	14816#	14844#	14846#	14847#	14860#	14875#	15004#	15007#	15010#	15013#	15016#	15019#	15022#	15025#					
	15028#	15031#	15034#	15037#	15040#	15043#	15046#	15049#	15052#	15055#	15058#	15061#	15065#	15068#	15072#					
	15075#	15079#	15082#	15086#	15089#	15093#	15096#	15100#	15103#	15107#	15110#	15114#	15117#	15121#	15124#					



.SPTB	4583#	6347#	6405
.SPTH	4853#	6347#	6400
.SPTY	5016#	6347#	6406
.SASTA	4900#		
.SCATC	898#	6346#	6384
.SCMTA	1000#		
.SDB2D	4169#		
.SDB2O	4293#		
.SDIV	4064#		
.SEOP	2004#		
.SERRO	2422#		
.SERRT	2615#		
.SMULT	4000#		
.SPOWE	3713#	6346#	20692
.SRAND	3776#		
.SRDDE	3406#	6348#	20688
.SRDOC	3314#		
.SREAD	3098#	6348#	20689
.SR2AZ	4437#		
.SSAVE	3482#		
.SSB2D	4254#		
.SSB2O	4356#		
.SSCOP	2214#		
.SSIZE	3838#		
.SSUPR	4394#		
.STRAP	3582#	6346#	20691
.STYPB	3013#		
.STYPD	2935#		
.STYPE	2703#	6346#	20690
.STYPO	2838#		
.S40CA	929#		
.1170	482#	6346#	6356

ADD	6406	8355	8472	8474	8481	8488	8520	8522	8529	8536	8546	8554	8575	8577	8623
	8625	8669	8671	8696	8702	8711	8717	8727	8738	8796	8854	8878	8893	8909	8926
	8948	9013	9057	9080	9093	9098	9102	9112	9128	9139	9143	9452	9464	9469	9471
	9682	9685	9688	9691	9693	9724	9948	9952	9967	9984	10330	10331	10356	10359	10660
	10665	11076	11082	11085	11093	11100	11106	11113	11118	11119	11128	11143	11227	11231	11243
	11308	11372	11376	11393	11399	11404	11423	11426	11432	11486	11501	11566	11627	11653	11764
	11768	11797	11802	11867	11871	11886	11894	12003	12006	12031	12039	12040	12048	12052	12053
	12059	12083	12084	12086	12097	12101	12112	12134	12135	12324	12372	12373	12391	12392	12462
	12470	12559	12602	12617	12622	12666	12707	12709	12711	12712	12742	12746	12747	12751	12757
	12788	12789	12790	12816	12910	12918	12925	12938	12952	12964	13574	13733	13797	13834	13844
	13867	13901	13995	14084	14096	14115	14190	14234	14235	14239	14284	14326	14327	14366	14367
	14368	14403	14404	14420	14433	14434	14455	14459	14472	14473	14576	14608	14642	14873	14910
	14925	14959	15194	19456	19457	19470	19487	19505	19514	19528	19529	19542	19559	19577	19583
	19584	19591	19592	19704	19705	19710	19711	19721	19722	19732	19733	19738	19739	19746	19747
	19754	19761	19770	19771	19776	19777	19784	19789	19797	19798	19801	19802	19809	19810	19813
	19814	19819	19820	19823	19832	19991	19992	19999	20000	20073	20074	20083	20084	20091	20196
	20197	20204	20205	20210	20374	20375	20384	20385	20401	20404	20431	20433	20441	20442	20453
	20467	20478	20480	20617	20663	20688	20690	20811	20853	20895					
ASL	9031	9032	9033	9034	9082	9083	9084	9085	10355	10358	10587	10588	10589	10590	11075
	11092	11099	11105	11127	11141	11142	11158	11159	11226	11230	11242	11276	11277	11278	11279
	11288	11289	11290	11291	11300	11301	11302	11303	11626	11636	11637	11638	11639	11801	12574
	12575	12733	12815	12906	12907	12909	12917	12933	12944	12945	12947	12970	13732	14083	14095
	14444	14445	14446	14544	14545	14546	14958	20688	20691						
ASR	6406	8794	8821	8822	8823	8824	9012	10048	10618	11153	11154	11155	11163	11598	11599
	11600	11601	11607	11608	11609	11610	11705	11706	11707	11708	11714	11715	11716	11717	11755
	11756	11757	11758	11776	11777	11778	11779	11830	11831	11832	11833	11857	11858	11859	11860
	11865	11868	11893	12027	12567	12568	12569	12641	12642	12643	12673	12674	12675	12931	12958
	12959	14849	14850	14851	14852	14853	19495	19512	19520	19555	19567	19702	19708	19719	19730
	19736	19744	19750	19752	19768	19774	19782	19787	20071	20081	20089	20372	20382	20397	20399
	20463														
BCC	9453	9462	9668	9675	9680	9683	9686	9689	9692	9694	9725				
BEQ	6406	8495	8500	8504	8543	8551	8558	8598	8602	8607	8629	8631	8635	8655	8659
	8663	8676	8678	8680	8699	8714	8720	8729	8740	8742	8799	8805	8811	8815	8833
	8835	8837	8883	8915	8935	8957	9023	9042	9056	9121	9130	9132	9159	9171	9174
	9177	9188	9192	9202	9209	9219	9312	9329	9346	9431	9458	9510	9513	9519	9521
	9533	9636	9646	9659	9711	9755	9770	9783	9785	9792	9823	9838	9851	9853	9860
	9893	9906	9909	9934	9955	9976	9997	10019	10023	10046	10072	10075	10086	10092	10107
	10132	10141	10153	10156	10170	10185	10205	10224	10243	10309	10318	10320	10325	10338	10352
	10550	10595	10765	10774	10801	10829	10870	10971	10990	10993	10995	11050	11098	11104	11117
	11166	11194	11221	11270	11316	11368	11379	11398	11403	11417	11422	11442	11468	11473	11499
	11524	11532	11534	11536	11538	11540	11542	11550	11554	11564	11575	11579	11617	11646	11649
	11658	11662	11675	11680	11772	11785	11787	11814	11819	11822	11876	11896	11898	11902	11906
	11910	11977	11983	11988	12021	12023	12036	12045	12055	12065	12069	12094	12096	12132	12154
	12162	12164	12172	12177	12276	12288	12306	12313	12317	12321	12328	12334	12339	12356	12365
	12369	12382	12395	12427	12437	12447	12483	12485	12493	12497	12499	12501	12537	12547	12610
	12631	12661	12693	12704	12714	12719	12721	12725	12740	12749	12761	12774	12785	12797	12803
	12807	12809	12833	12846	12866	12868	12870	12902	13040	13141	13185	13187	13305	13315	13326
	13329	13331	13336	13391	13468	13470	13476	13502	13520	13523	13563	13565	13567	13612	13643
	13646	13663	13669	13689	13703	13721	13760	13944	14086	14100	14126	14159	14168	14207	14264
	14272	14277	14283	14287	14292	14297	14321	14338	14342	14357	14359	14361	14423	14425	14465
	14489	14491	14493	14495	14497	14499	14501	14503	14505	14515	14517	14541	14543	14645	14694
	14697	14704	14707	14710	14714	14717	14720	14723	14727	14741	14754	14766	14771	14780	14785
	14808	14822	14907	14909	14913	14918	14940	14944	15188	19468	19475	19486	19492	19497	19540
	19547	19558	19564	19569	20392	20395	20451	20466	20688	20690					

BGE	11883	20690													
BGT	11384	11435	11739	11953	14442	14652	20688								
BHI	8476	8524	8579	9245	9969	10201	10220	11444	11594	13666	13723	13736	14080	14219	20447
	20449	20458	20460	20472	20474										
BHIS	9353	9540	9986	11087	13692	13800	13805	13836	13842						
BIC	8789	8795	8806	8831	8856	8881	8930	8947	8954	8963	8967	9006	9014	9051	9053
	9095	9144	9455	9498	9656	9761	9801	9829	9872	9924	9946	9949	9951	9996	10054
	10061	10068	10096	10099	10162	10164	10304	10368	10380	10503	10508	11033	11054	11074	11149
	11152	11241	11281	11293	11305	11369	11488	11500	11525	11529	11530	11551	11602	11604	11606
	11611	11614	11709	11711	11718	11719	11728	11731	11759	11761	11762	11780	11809	11829	11861
	11863	11923	11926	11929	11938	12160	12204	12206	12217	12221	12268	12332	12335	12336	12481
	12562	12566	12573	12592	12599	12604	12614	12619	12624	12636	12640	12663	12668	12672	12695
	12701	12702	12717	12723	12736	12819	12903	12905	12916	12930	12943	12957	12969	13147	13155
	13156	13157	13158	13161	13188	13204	13312	13560	13592	13601	13607	13637	13677	13710	13918
	14077	14110	14119	14166	14217	14339	14340	14401	14402	14415	14443	14451	14509	14525	14530
	14539	14564	14566	14588	14616	14739	14752	14853	14857	14941	14942	14963	20613	20633	20644
	20655	20689	20690												
BICB	9036	9081	10321	10518	10527	11633	11891	11899	14131	14548					
BIS	9054	9146	9230	9241	9349	9351	9354	9504	9536	9538	9541	9664	9712	9717	9729
	9734	9789	9793	9804	9857	9861	9864	9925	10062	10067	10069	10083	10087	10100	10110
	10246	10248	10369	10370	10381	10395	10610	10910	10913	10926	11282	11294	11306	11720	11743
	11930	12207	12210	12214	12241	12247	12249	12253	12634	12698	12699	13146	13162	13170	13171
	13172	13173	13202	13321	13339	13343	13345	13614	13644	13647	13648	13746	13935	13955	13971
	14447	14589	14617	14643	14964	20628	20634	20639	20645	20650	20656	20661			
BISB	9037	9086	10323	10591	10596	11640	11903	11907	11911	11913	14268	14270	14289	14293	14317
	14318	14549	14961												
BIT	8798	8804	8934	8944	9029	9158	9170	9173	9176	9430	9635	9704	9730	9754	9822
	9905	9908	10018	10152	10155	10287	10290	10308	10337	10501	10506	10536	10568	11258	11313
	11338	11340	11358	11360	11389	11416	11450	11523	11648	11657	11661	11679	11748	11821	11889
	11901	11905	11909	11945	11959	12018	12275	12307	12358	12465	12486	12494	12507	12513	12519
	12536	12543	12550	12632	12690	12805	12901	13142	13325	13611	13619	13667	13688	13701	13759
	13925	14106	14206	14263	14337	14696	14703	14706	14709	14713	14716	14719	14722	14726	14765
	14770	14779	14784	14807	14939	15185	20688								
BITB	6406	8704	8719	10319	10765	10774	20690								
BLE	9932														
BLO	8477	8491	8525	8539	8580	8594	8965	9212	9246	9502	10202	10221	10988	11168	11175
	11182	13654	14532	14639	19759	19830									
BLOS	9142	9803	9863	11196	11925	11937	12380	12388	13359	13364	13369	13374	13379	14130	20689
	20769	20774	20779												
BLT	11121	11351	14440	14593	14621	14651	20688	20690							
BMI	8780	8791	8997	9009	9039	9223	9234	9324	9481	9707	9760	9767	9828	9835	9889
	9929	9978	10346	11874	12794										
BNE	6406	8350	8353	8361	8649	8705	8731	8808	8813	8843	8850	8863	8865	8871	8877
	8892	8897	8904	8908	8912	8919	8932	8942	8945	8951	8959	9016	9018	9030	9064
	9066	9069	9074	9076	9089	9092	9097	9105	9111	914	9186	9262	9277	9339	9438
	9446	9448	9479	9508	9703	9705	9714	9716	9731	9788	9800	9856	9871	9920	9959
	10017	10031	10033	10060	10082	10098	10105	10175	10190	10230	10235	10288	10291	10307	10502
	10507	10537	10569	10593	10831	10863	10876	11055	11057	11068	11200	11250	11259	11274	11286
	11298	11314	11331	11339	11341	11359	11361	11390	11409	11451	11491	11583	11624	11694	11697
	11727	11732	11742	11749	11782	11794	11890	11948	11962	12019	12067	12071	12088	12150	12201
	12209	12213	12224	12226	12234	12244	12286	12308	12342	12344	12346	12348	12350	12352	12354
	12359	12363	12367	12371	12390	12406	12408	12410	12412	12415	12468	12487	12489	12495	12508
	12514	12520	12526	12530	12535	12542	12558	12580	12582	12590	12597	12601	12606	12616	12621
	12626	12633	12647	12649	12665	12678	12680	12691	12697	12770	12772	12777	12811	12813	12978

CZKEEB.P11	EXERCISER CROSS REFERENCE	MACY11 CROSS REFERENCE	27(655) CROSS REFERENCE	29-SEP-80 CROSS REFERENCE	09:22 CROSS REFERENCE	PAGE 183-61 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE	SEQUENCE 449 CROSS REFERENCE
12982	12986	12990	13002	13006	13010	13014	13042	13044	13061	13063	13075	13093	13117	13145	
13190	13192	13194	13196	13198	13200	13229	13233	13303	13308	13317	13320	13338	13342	13353	
13396	13398	13420	13427	13444	13451	13472	13474	13478	13484	13488	13525	13527	13610	13620	
13816	13818	13821	13824	13826	13828	13830	13832	13840	13892	13922	13926	14103	14107	14124	
14128	14133	14170	14174	14179	14183	14187	14302	14344	14350	14363	14417	14427	14700	14743	
14756	14814	14818	14820	14838	14843	14916	14948	15192	15196	15200	15208	20213	20390	20669	
20688	20689	20690	20692	20727	20731	20813	20855	20897							
BPL 9138	9268	9275	9283	9287	9318	9348	9441	9535	9701	9720	9728	9733	9957	13558	
BR 14164	14335	14374	14413	14449	14587	14615	14937	20689	20690						
6406	8479	8486	8497	8502	8506	8527	8534	8548	8556	8560	8582	8589	8600	8604	
8609	8651	8657	8665	8685	8782	8801	8825	8838	8845	8852	8886	8890	8895	8901	
8906	8913	8920	8937	8999	9021	9025	9067	9070	9072	9078	9087	9090	9100	9103	
9108	9116	9118	9124	9134	9152	9154	9161	9190	9249	9252	9255	9271	9279	9291	
9334	9350	9423	9450	9465	9488	9511	9515	9537	9628	9696	9718	9776	9790	9844	
9858	9895	9911	9922	9943	9971	9989	10001	10021	10050	10077	10084	10109	10145	10158	
10178	10193	10227	10231	10236	10244	10247	10311	10322	10332	10334	10340	10384	10545	10571	
10648	10652	10656	10661	10666	10704	10708	10712	10716	10720	10752	10765	10771	10774	10791	
10804	10809	10814	10954	10957	10964	10966	10974	10998	11109	11171	11178	11185	11223	11283	
11295	11381	11425	11476	11515	11558	11568	11589	11641	11642	11651	11655	11660	11664	11666	
11671	11677	11682	11684	11690	11695	11699	11729	11734	11774	11790	11799	11824	11826	11879	
11898	11904	11908	11912	11914	12041	12049	12057	12077	12085	12102	12138	12152	12184	12211	
12237	12245	12246	12248	12252	12256	12265	12279	12290	12323	12383	12393	12439	12510	12516	
12522	12544	12594	12607	12627	12656	12687	12726	12743	12763	12836	12980	12984	12988	12992	
13004	13008	13012	13016	13059	13072	13088	13096	13104	13107	13110	13113	13160	13201	13203	
13213	13310	13333	13340	13344	13393	13405	13418	13435	13442	13458	13480	13482	13486	13490	
13568	13570	13572	13617	13664	13763	13843	13868	13902	13924	13973	13996	14105	14117	14121	
14162	14172	14177	14181	14185	14209	14266	14280	14285	14290	14313	14316	14319	14325	14347	
14355	14365	14389	14393	14396	14399	14452	14456	14460	14504	14512	14520	14523	14528	14536	
14553	14568	14580	14594	14622	14641	14647	14654	14745	14758	14769	14774	14783	14788	14826	
14830	14845	14911	14914	14921	15006	15012	15018	15024	15030	15036	15042	15048	15054	15060	
15067	15074	15081	15088	1509	15102	15109	15116	15123	15130	15137	15144	15151	15158	15165	
15172	15178	15198	15202	15210	15215	20393	20630	20636	20641	20647	20652	20658	20679	20688	
20689	20690	20692	20817	20859											
BVS 20688															
CCC 9451	9459	9665	9672	9677	9684	9723									
CLC 14230															
CLR 8341	8489	8509	8510	8511	8537	8563	8564	8565	8592	8612	8613	8614	8809	8817	
8873	8884	8885	8889	8900	8905	8936	8939	8952	8970	9020	9040	9071	9136	9153	
9157	9162	9164	9165	9166	9184	9200	9205	9206	9207	9220	9221	9257	9260	9264	
9265	9266	9270	9278	9316	9340	9344	9356	9357	9358	9359	9362	9422	9425	9426	
9427	9428	9429	9436	9439	9456	9466	9485	9489	9500	9531	9545	9546	9627	9630	
9631	9632	9633	9634	9644	9709	9736	9737	9748	9749	9750	9753	9758	9781	9806	
9807	9815	9816	9817	9818	9821	9826	9849	9874	9875	9882	9883	9903	9904	9912	
9913	9914	9915	9916	9938	9972	9987	9990	10020	10034	10043	10080	10112	10113	10116	
10144	10150	10151	10159	10199	10250	10251	10252	10253	10284	10293	10312	10336	10341	10388	
10389	10403	10404	10418	10420	10428	10437	10497	10498	10513	10514	10522	10523	10559	10600	
10601	10763	10764	10765	10769	10770	10772	10773	10774	10781	10783	10785	10819	10822	10824	
10826	10834	10839	10840	10841	10842	10843	10845	10846	10847	10967	10977	10978	10979	10980	
10981	11037	11045	11046	11047	11048	11089	11090	11198	11214	11215	11218	11440	11544	12033	
12062	12091	12129	12220	12222	12266	12299	12300	12301	12403	12413	12420	12538	12583	12584	
12585	12586	12652	12682	12734	12828	12829	12831	12871	12993	13037	13068	13073	13122	13159	
13234	13309	13323	13332	13334	13357	13362	13367	13372	13377	13499	13517	13545	13569	13595	
13602	13604	13854	13859	13886	13890	13894	13898	13916	13954	13958	13970	13974	13989	13991	
14122	14161	14188	14189	14268	14270	14289	14293	14295	14317	14318	14322	14324	14345	14411	



	14578	14610	14634	14636	14702	14724	14806	14827	14861	14932	14933	14934	15217	20688	20692
CLRB	20732	20815	20857	20899	20909										
CMP	6406	8545	8553	8701	8716	9229	9240	10066	14655	14832	20688	20689	20690		
	8352	8360	8475	8490	8494	8499	8503	8514	8523	8538	8542	8550	8557	8568	8578
	8593	8597	8601	8606	8617	8628	8630	8648	8658	8675	8677	8698	8713	8728	8739
	8810	8812	8882	8931	8941	8950	8956	8964	9141	9208	9244	9261	9437	9445	9447
	9457	9501	9512	9802	9862	9968	9985	10074	10200	10219	10242	10306	10324	10594	10765
	10774	10828	10987	10992	11086	11120	11167	11174	11181	11195	11220	11272	11284	11296	11350
	11380	11382	11396	11402	11434	11443	11533	11535	11537	11539	11541	11593	11645	11674	11693
	11696	11741	11813	11895	11924	11936	11952	11976	11981	11987	12020	12022	12044	12054	12064
	12066	12068	12070	12095	12131	12149	12153	12161	12163	12208	12212	12223	12233	12285	12287
	12304	12312	12316	12320	12327	12338	12343	12345	12347	12349	12351	12353	12355	12364	12366
	12368	12370	12379	12387	12389	12482	12484	12496	12498	12500	12534	12541	12546	12581	12589
	12593	12596	12600	12605	12609	12615	12620	12625	12648	12655	12664	12679	12686	12696	12748
	12773	12808	12977	12981	12985	12989	13001	13005	13009	13013	13189	13191	13193	13195	13197
	13199	13306	13314	13316	13318	13341	13358	13363	13368	13373	13378	13419	13443	13469	13471
	13477	13487	13501	13522	13524	13526	13609	13652	13690	13722	13734	13799	13804	13814	13817
	13819	13823	13825	13827	13829	13831	13835	13839	13841	13891	13921	14079	14099	14101	14123
	14125	14127	14132	14167	14169	14173	14178	14182	14186	14218	14276	14282	14341	14343	14349
	14356	14358	14360	14362	14416	14422	14424	14426	14439	14441	14464	14488	14490	14492	14494
	14496	14498	14500	14502	14504	14514	14516	14531	14650	14740	14742	14753	14755	14908	14912
	14915	14947	15199	19758	19829	20446	20448	20457	20459	20471	20473	20668	20689	20690	20768
20773	20778														
CMPB	6406	8632	8634	8652	8654	8660	8662	8679	8730	8741	8807	8814	8832	8834	8836
	8841	8864	8867	8876	8891	8896	8903	8907	8911	8914	8918	9015	9017	9055	9065
	9068	9073	9088	9091	9104	9110	9120	9211	9218	9352	9507	9539	9658	9933	9953
	9958	9975	10106	10131	10169	10184	10223	12381	12394	12718	13039	13041	13060	13562	13564
13566	14129	14540	14542	14837	14943	20212	20688	20689	20690						
COM	9284	9288	9443	9444	9721	9722	10858	12165	12527	14184	14962				
DEC	8636	8637	8656	8664	8706	8721	8732	8743	8840	8898	8968	9041	9123	9133	9478
	9496	9524	9887	9942	10093	10353	10592	11332	11333	11410	11411	11557	11567	11582	11587
	11622	11689	11736	11737	11740	11770	11792	11793	11798	11881	11882	11887	12038	12046	12242
	12397	12398	12738	12767	12771	14231	14353	14646	14817						
DECB	9267	9274	9323	9759	9766	9827	9834	9918	9927	9941	9999	10000	10027	10143	10345
	14586	14592	14614	14620	20690										
EMT	6356														
HALT	6384	6549	6707	8916	8961	9122	10790	11543	12173	12257	12258	12259	12260	12261	12548
	12611	13211	13507	13529	13534	13550	13626	13659	13676	13699	13708	13728	13932	20690	20692
INC	20808	20850	20892												
	6406	8682	8792	9010	9106	9135	9247	9248	9269	9285	9289	9310	9320	9454	9483
	9505	9514	9528	9663	9726	9768	9836	9891	10049	10076	10108	10176	10191	10222	10226
	10383	10544	10570	10619	10859	10985	11095	11192	11197	11199	11201	11329	11407	11581	11670
	11872	12419	12533	12560	12629	12659	12830	12976	13000	13613	13618	13651	13680	13934	13972
	14078	14640	14823	14824	19703	19709	19720	19731	19737	19745	19769	19775	19783	19788	20072
	20082	20090	20214	20373	20383	20692	20788	20827	20869						
INCB	9963	14836	20690												
IOT	6356														
JMP	6384	6386	6387	6388	8692	8707	8722	8733	8866	8872	8874	10003	10146	10546	10547
	10607	10677	10686	10700	10734	10746	10874	10881	10883	10886	10887	10888	10889	10893	10894
	10895	10896	10900	10903	10904	10909	10912	10914	10918	10921	10925	10927	10931	10936	10937
	10938	10939	10948	10949	10950	10951	10960	10961	10962	10963	11077	11244	11251	11257	11309
	11322	11327	11334	11353	11385	11405	11412	11436	11445	11513	11545	11546	11559	11569	11590
	11628	11631	11700	11745	11750	11783	11803	11815	11816	11839	11843	11848	11853	11918	11954
	11998	12024	12025	12139	12293	12416	12443	12444	12449	12454	12502	12503	12504	12505	12531

	12532	12539	12551	12553	12587	12653	12657	12684	12688	12700	12705	12753	12779	12786	12823
	12825	12861	12872	12873	12888	12889	12892	13027	13230	13511	13539	13685	13713	14306	14307
	14308	14309	19506	19578	19762	19833	20405	20479	20481	20728	20903	20912			
JSR	6406	8362	8781	8783	8844	8851	8899	8998	9000	9183	9199	9231	9242	9272	9280
	9290	9298	9306	9309	9333	9343	9463	9476	9486	9506	9527	9530	9643	9662	9757
	9765	9775	9780	9798	9825	9833	9843	9848	9869	9890	9940	9965	9983	9995	10014
	10029	10040	10064	10070	10103	10138	10173	10188	10212	10218	10228	10233	10239	10241	10262
	10269	10283	10285	10286	10289	10303	10316	10350	10367	10379	10394	10425	10434	10443	10496
	10499	10500	10505	10517	10526	10535	10543	10557	10567	10579	10586	10606	10789	10802	10803
	10808	10813	10827	10856	10873	10877	10879	10880	10882	10884	10885	10891	10892	10899	10901
	10908	10911	10917	10919	10920	10924	10930	10932	10933	10935	10946	10947	10955	10956	10958
	10959	10968	10969	10972	11034	11051	11058	11069	11070	11071	11131	11133	11169	11172	11176
	11179	11183	11186	11204	11213	11256	11262	11264	11268	11317	11319	11323	11325	11342	11345
	11362	11364	11374	11391	11394	11400	11418	11419	11452	11454	11457	11461	11462	11469	11470
	11480	11481	11492	11495	11503	11509	11647	11656	11678	11773	11788	11811	11820	11846	11851
	11877	11900	11922	11928	11935	11974	11978	11979	11984	11985	11989	11993	12026	12043	12063
	12092	12130	12166	12169	12178	12229	12240	12282	12289	12291	12302	12309	12310	12314	12315
	12318	12319	12322	12325	12329	12337	12428	12438	12440	12441	12442	12694	12821	12852	12924
	12937	12951	12963	12979	12983	12987	12991	13003	13007	13011	13015	13025	13206	13208	13209
	13210	13215	13311	13324	13505	13506	13528	13532	13533	13548	13549	13561	13608	13615	13616
	13621	13623	13624	13625	13631	13632	13640	13655	13657	13658	13670	13672	13673	13674	13675
	13682	13687	13693	13695	13696	13697	13700	13704	13706	13707	13724	13726	13727	13919	13923
	13927	13929	13930	13931	13942	14076	14108	14118	14141	14143	14171	14197	14198	14205	14216
	14262	14265	14267	14268	14269	14270	14273	14279	14281	14288	14289	14293	14294	14298	14299
	14305	14312	14314	14315	14317	14318	14369	14387	14391	14437	14458	14511	14527	14535	14567
	14705	14708	14711	14715	14718	14721	14725	14729	14733	14736	14744	14746	14749	14757	14759
	14762	14764	14777	14778	14791	14816	14828	14833	14834	14839	14844	14846	14847	14854	14858
	14860	14875	14946	14950	15004	15005	15007	15010	15011	15013	15016	15017	15019	15022	15023
	15025	15028	15029	15031	15034	15035	15037	15040	15041	15043	15046	15047	15049	15052	15053
	15055	15058	15059	15061	15065	15066	15068	15072	15073	15075	15079	15080	15082	15086	15087
	15089	15093	15094	15096	15100	15101	15103	15107	15108	15110	15114	15115	15117	15121	15122
	15124	15128	15129	15131	15135	15136	15138	15142	15143	15145	15149	15150	15152	15156	15157
	15159	15163	15164	15166	15170	15171	15173	15176	15177	15179	15189	15190	15193	15206	15214
	15216	20690	20798	20806	20810	20838	20848	20852	20880	20890	20894				
MFPI	12277	13761													
MOV	6406	8335	8336	8337	8338	8339	8340	8342	8343	8344	8345	8346	8347	8348	8356
	8357	8358	8359	8363	8364	8365	8366	8367	8368	8471	8473	8478	8480	8482	8483
	8485	8487	8492	8493	8498	8507	8512	8513	8515	8519	8521	8526	8528	8530	8531
	8533	8535	8540	8541	8549	8561	8566	8567	8569	8574	8576	8581	8583	8585	8586
	8588	8590	8595	8596	8605	8610	8615	8616	8618	8622	8624	8626	8627	8633	8640
	8641	8642	8643	8644	8645	8646	8650	8653	8661	8668	8670	8672	8673	8674	8681
	8684	8686	8687	8688	8691	8695	8697	8710	8712	8725	8726	8736	8737	8745	8746
	8747	8748	8749	8750	8751	8753	8787	8788	8793	8797	8816	8818	8819	8828	8829
	8830	8857	8858	8859	8880	8888	8929	8938	8946	8966	9004	9005	9007	9011	9027
	9028	9035	9043	9044	9045	9049	9050	9052	9059	9060	9061	9151	9155	9156	9160
	9163	9167	9168	9169	9172	9175	9178	9179	9180	9181	9182	9189	9195	9196	9197
	9198	9203	9224	9225	9226	9227	9228	9232	9235	9236	9237	9238	9239	9243	9250
	9251	9253	9254	9256	9258	9259	9263	9273	9281	9294	9295	9296	9297	9302	9303
	9304	9305	9313	9321	9322	9325	9326	9330	9331	9332	9335	9336	9337	9341	9342
	9355	9360	9361	9424	9432	9433	9434	9435	9449	9467	9468	9470	9472	9473	9490
	9491	9494	9497	9499	9522	9523	9525	9526	9529	9542	9543	9544	9547	9548	9629
	9637	9640	9641	9642	9647	9654	9657	9660	9661	9669	9670	9671	9676	9681	9687
	9690	9695	9708	9735	9738	9739	9740	9741	9747	9751	9752	9756	9762	9763	9764
	9771	9772	9773	9774	9777	9778	9779	9786	9794	9795	9796	9797	9805	9808	9809

9810	9811	9819	9820	9824	9830	9831	9832	9839	9840	9841	9842	9845	9846	9847
9854	9865	9866	9867	9868	9873	9876	9877	9878	9879	9884	9885	9886	9894	9902
9907	9910	9917	9921	9923	9926	9935	9936	9937	9939	9950	9960	9961	9962	9964
9970	9979	9980	9981	9982	9988	9991	9992	9993	9994	9998	10010	10011	10012	10013
10024	10025	10026	10028	10036	10037	10038	10039	10047	10051	10053	10055	10056	10057	10058
10065	10073	10090	10094	10095	10101	10102	10111	10114	10115	10117	10133	10134	10135	10136
10137	10142	10149	10154	10157	10161	10163	10165	10166	10167	10168	10171	10172	10177	10179
10181	10182	10183	10186	10187	10192	10194	10196	10206	10207	10208	10209	10210	10211	10213
10214	10215	10216	10217	10240	10249	10254	10255	10258	10259	10260	10261	10265	10266	10267
10268	10294	10295	10296	10297	10298	10299	10300	10301	10302	10305	10310	10313	10314	10315
10326	10327	10328	10329	10339	10342	10343	10344	10344	10348	10349	10354	10357	10360	10361
10362	10363	10364	10365	10366	10371	10372	10373	10373	10375	10376	10377	10378	10386	10387
10390	10391	10392	10393	10398	10401	10402	10408	10409	10410	10411	10412	10413	10414	10415
10419	10421	10422	10423	10424	10429	10430	10431	10432	10433	10438	10439	10440	10441	10442
10504	10509	10511	10512	10515	10516	10520	10521	10524	10525	10529	10530	10531	10532	10533
10534	10538	10539	10540	10541	10542	10552	10553	10554	10555	10556	10561	10562	10563	10564
10565	10566	10573	10574	10575	10576	10577	10578	10581	10582	10583	10584	10585	10598	10599
10602	10603	10604	10605	10609	10612	10615	10617	10628	10629	10630	10631	10632	10633	10634
10635	10636	10637	10638	10639	10640	10641	10642	10643	10647	10651	10655	10659	10664	10669
10670	10671	10672	10673	10674	10675	10676	10680	10681	10682	10683	10684	10685	10690	10691
10692	10693	10694	10695	10696	10697	10698	10699	10703	10707	10711	10715	10719	10723	10724
10725	10726	10727	10728	10729	10730	10731	10732	10733	10737	10738	10739	10740	10741	10742
10743	10744	10745	10751	10754	10757	10765	10768	10774	10784	10789	10802	10807	10812	10817
10818	10820	10823	10825	10835	10836	10837	10838	10857	10860	10865	10866	10867	10868	10877
10882	10884	10891	10930	10932	10933	10940	10946	10952	10953	10955	10958	10965	10968	10975
10976	10982	10983	10984	11031	11053	11058	11072	11073	11080	11081	11083	11084	11091	11101
11102	11107	11108	11112	11114	11115	11123	11124	11125	11129	11130	11140	11144	11147	11150
11156	11161	11164	11170	11173	11177	11180	11184	11187	11203	11216	11217	11219	11222	11224
11225	11228	11229	11232	11235	11266	11267	11271	11275	11287	11299	11321	11347	11366	11371
11420	11427	11428	11429	11430	11431	11456	11460	11471	11474	11475	11477	11478	11479	11487
11494	11497	11502	11507	11518	11519	11520	11521	11522	11526	11527	11528	11552	11562	11572
11576	11577	11584	11585	11596	11603	11605	11613	11615	11619	11625	11650	11652	11659	11663
11665	11669	11687	11698	11703	11710	11724	11730	11753	11760	11763	11765	11766	11767	11795
11796	11800	11807	11810	11812	11823	11825	11827	11834	11838	11847	11852	11855	11862	11864
11866	11869	11870	11884	11885	11892	11917	11927	11949	11950	11951	11963	11964	11965	11966
11967	11968	11969	12012	12013	12014	12015	12016	12017	12028	12032	12034	12037	12042	12047
12050	12051	12056	12060	12061	12072	12073	12075	12076	12078	12079	12081	12082	12090	12098
12100	12105	12106	12107	12108	12109	12110	12115	12116	12117	12118	12119	12120	12122	12123
12124	12125	12126	12127	12133	12136	12141	12142	12143	12144	12145	12146	12148	12151	12155
12156	12157	12158	12159	12168	12173	12182	12183	12185	12186	12187	12188	12189	12190	12191
12192	12193	12194	12195	12196	12197	12198	12199	12205	12215	12216	12218	12219	12227	12228
12229	12230	12232	12235	12239	12254	12264	12267	12269	12270	12271	12272	12273	12274	12278
12280	12281	12289	12291	12292	12340	12357	12360	12361	12385	12386	12396	12399	12400	12421
12422	12423	12428	12430	12431	12432	12433	12434	12435	12438	12440	12441	12480	12509	12511
12515	12517	12521	12523	12561	12563	12565	12570	12572	12577	12591	12598	12603	12613	12618
12623	12635	12637	12639	12644	12650	12651	12662	12667	12669	12671	12676	12681	12683	12706
12708	12710	12715	12722	12730	12756	12768	12775	12778	12787	12804	12814	12817	12844	12883
12884	12885	12886	12887	12900	12904	12911	12915	12919	12929	12942	12956	12968	13017	13026
13035	13047	13049	13050	13051	13054	13066	13067	13069	13071	13078	13079	13080	13081	13082
13083	13084	13085	13086	13087	13099	13102	13105	13120	13121	13123	13124	13125	13126	13127
13149	13150	13151	13152	13153	13154	13164	13165	13166	13167	13168	13169	13174	13206	13207
13209	13210	13215	13235	13236	13237	13238	13239	13240	13241	13242	13244	13245	13246	13247
13248	13249	13250	13251	13253	13254	13255	13256	13257	13258	13259	13260	13262	13263	13264
13265	13266	13267	13268	13269	13271	13272	13273	13274	13275	13276	13277	13278	13280	13281

	13282	13283	13284	13285	13286	13287	13322	13327	13356	13361	13366	13371	13376	13383	13385
	13386	13388	13389	13392	13394	13400	13406	13408	13413	13421	13430	13437	13445	13453	13460
	13479	13481	13485	13489	13491	13503	13505	13506	13508	13528	13530	13532	13533	13536	13546
	13548	13549	13571	13573	13600	13621	13622	13624	13625	13630	13638	13639	13641	13655	13656
	13658	13670	13671	13673	13674	13675	13681	13693	13694	13696	13697	13704	13705	13707	13711
	13712	13724	13725	13727	13731	13737	13742	13743	13744	13753	13754	13755	13756	13757	13758
	13762	13764	13771	13772	13773	13774	13775	13776	13786	13787	13788	13789	13790	13791	13792
	13794	13795	13801	13802	13803	13833	13837	13838	13850	13851	13852	13853	13858	13860	13862
	13864	13865	13871	13872	13873	13882	13883	13884	13885	13888	13889	13893	13896	13899	13905
	13906	13907	13914	13917	13927	13928	13930	13931	13936	13941	13945	13953	13959	13969	13976
	13985	13986	13987	13988	13993	13999	14000	14001	14009	14010	14011	14015	14016	14017	14021
	14023	14024	14028	14030	14031	14035	14036	14037	14041	14042	14043	14047	14048	14049	14053
	14054	14055	14059	14060	14062	14082	14093	14094	14097	14098	14104	14116	14120	14134	14135
	14142	14144	14160	14175	14176	14180	14196	14197	14208	14210	14229	14236	14242	14243	14265
	14267	14268	14269	14270	14273	14274	14275	14278	14279	14281	14288	14289	14293	14298	14299
	14314	14317	14318	14351	14352	14364	14390	14397	14398	14400	14430	14431	14432	14436	14457
	14466	14467	14468	14469	14470	14471	14481	14482	14483	14484	14485	14486	14510	14521	14526
	14533	14534	14550	14554	14555	14556	14557	14558	14559	14560	14565	14584	14612	14618	14630
	14631	14632	14635	14637	14653	14657	14658	14712	14728	14733	14734	14735	14738	14744	14746
	14747	14748	14751	14757	14759	14760	14761	14764	14767	14768	14772	14773	14775	14776	14778
	14781	14782	14786	14787	14789	14790	14804	14805	14809	14810	14812	14815	14816	14825	14831
	14844	14846	14847	14848	14850	14871	14872	14875	14876	14905	14919	14920	14922	14935	14957
	15004	15007	15010	15013	15016	15019	15022	15025	15028	15031	15034	15037	15040	15043	15046
	15049	15052	15055	15058	15061	15062	15065	15068	15069	15072	15075	15076	15079	15082	15083
	15086	15089	15090	15093	15096	15097	15100	15103	15104	15107	15110	15111	15114	15117	15118
	15121	15124	15125	15128	15131	15132	15135	15138	15139	15142	15145	15146	15149	15152	15153
	15156	15159	15160	15163	15166	15167	15170	15173	15176	15179	15189	15190	15193	15206	15214
	15216	19455	19458	19461	19464	19469	19471	19476	19478	19481	19482	19488	19493	19494	19499
	19502	19510	19511	19515	19518	19519	19522	19527	19530	19533	19536	19541	19543	19548	19550
	19553	19554	19560	19565	19566	19571	19574	19582	19585	19590	19593	19706	19712	19717	19723
	19734	19740	19748	19755	19756	19772	19778	19785	19790	19799	19803	19811	19815	19821	19826
	19827	19990	19993	19998	20001	20004	20075	20085	20092	20198	20206	20211	20376	20386	20402
	20434	20443	20454	20468	20475	20614	20615	20616	20619	20621	20623	20624	20629	20635	20640
	20646	20651	20657	20662	20665	20667	20671	20672	20673	20674	20675	20677	20678	20688	20689
	20690	20691	20692	20733	20735	20736	20737	20738	20739	20740	20741	20742	20744	20745	20746
	20747	20748	20749	20750	20751	20753	20754	20755	20756	20757	20758	20759	20760	20766	20767
	20771	20772	20776	20777	20789	20790	20791	20792	20793	20795	20796	20806	20807	20816	20828
	20829	20830	20831	20832	20833	20835	20836	20847	20848	20849	20858	20870	20871	20872	20873
	20874	20875	20877	20878	20889	20890	20891	20900	20901	20902	20925	20926	20927	20928	20929
	20930	20933	20934	20935	20936	20937	20938								
	6406	8496	8501	8505	8544	8547	8552	8555	8559	8599	8603	8608	8700	8703	8715
MOVB	8718	8800	8802	8803	8855	8879	8894	8910	8917	8928	8933	8943	8953	8962	9019
	9024	9026	9058	9077	9079	9094	9099	9115	9117	9140	9145	9210	9213	9214	9215
	9314	9319	9327	9442	9474	9475	9477	9484	9487	9492	9495	9503	9928	9947	9966
	10044	10063	10089	10986	11094	11096	11145	11146	11148	11151	11157	11162	11555	11565	11580
	11621	11654	11676	11681	11683	11712	11713	11721	11733	11735	11744	11775	11789	11791	11808
	11828	11842	11878	11880	12401	12402	12732	12766	13559	14165	14336	14375	14414	14450	14487
	14508	14513	14518	14524	14529	14537	14547	14563	14574	14577	14590	14606	14609	14648	14811
	14856	14874	14938	14945	14949	14960	15197	15201	15203	15209	15211	19701	19707	19718	19729
	19735	19743	19749	19751	19767	19773	19781	19786	19796	19800	19808	19812	19818	19822	19824
	20070	20080	20088	20195	20203	20209	20371	20381	20396	20398	20429	20440	20452	20461	20462
	20688	20689	20690	20691											
NEG	14575	14607	20688												
NOP	6548	6811	6812	6813	10805	10806	10810	10811	10815	10816	11210	11211	11212	12168	12182

	12183	12250	12262	12364	12366	12853	12854	12855	13078	13079	13080	13081	13083	13084	13085
	13086	20804	20805	20845	20846	20887	20888								
RESET	12851														
ROL	9666	9667	9673	9674	9678	9679	11236	11237	11238	11239	11240	14232	14233	14237	14238
	14240	14241	14579	14581	14582	14583	14585	14611	14613						
ROR	9460	9461													
RTI	13535	13551	13593	13649	13946	20688	20689	20690	20692						
RTS	6406	8351	8369	8516	8570	8619	8647	8754	8784	8820	8860	8940	8949	8955	8969
	8971	9001	9046	9062	9147	9299	9307	9315	9363	9482	9549	9742	9812	9880	9896
	10118	10256	10263	10270	10405	10416	10426	10435	10444	10620	10644	11835	11931	11939	11940
	12007	12473	12912	12920	12926	12934	12939	12948	12953	12960	12965	12971	12994	13018	13128
	13175	13205	13216	13289	13346	13492	13575	13738	13745	13748	13765	13778	13776	13845	13874
	13908	13962	13979	14002	14012	14018	14025	14032	14038	14044	14050	14056	14064	14087	14136
	14145	14191	14199	14211	14220	14244	14328	14371	14376	14388	14392	14405	14421	14435	14474
	14561	14595	14623	14659	14695	14698	14701	14730	14737	14750	14763	14792	14862	14877	14926
	14951	14965	15008	15014	15020	15026	15032	15038	15044	15050	15056	15063	15070	15077	15084
	15091	15098	15105	15112	15119	15126	15133	15140	15147	15154	15161	15168	15174	15180	15218
	19459	19465	19472	19479	19489	19500	19503	19516	19523	19531	19537	19544	19551	19561	19572
	19575	19586	19594	19713	19724	19741	19760	19779	19791	19804	19816	19831	19994	20002	20005
	20076	20086	20093	20199	20207	20215	20377	20387	20403	20436	20444	20455	20469	20476	20625
	20676	20690	20691	20931	20939										
SOB	14552														
SUB	6406	8354	8484	8532	8584	8587	8591	8638	8639	8689	8744	9216	9217	9493	10052
	10091	10180	10195	11348	11508	11514	12002	12005	12030	12074	12080	12099	12202	12461	12463
	12469	12471	12758	13504	13531	13547	13719	13975	14633	14638	14813	19462	19463	19477	19484
	19498	19513	19521	19534	19535	19549	19556	19570	19753	19757	19825	19828	20400	20464	20618
	20620	20622	20664	20656	20670	20688									
SWAB	11160	11280	11292	11304	11597	11704	11754	11856	12576	12731	12735	12908	12932	12946	
TRAP	20691														
TST	6406	8349	8508	8562	8611	8690	8752	8779	8790	8848	8958	8996	9008	9022	9038
	9075	9096	9113	9129	9137	9185	9187	9191	9201	9222	9233	9276	9282	9286	9311
	9328	9338	9345	9347	9440	9509	9520	9532	9645	9699	9702	9706	9710	9713	9715
	9719	9727	9732	9769	9782	9784	9787	9791	9799	9837	9850	9852	9855	9859	9870
	9892	9919	9930	10016	10030	10045	10059	10071	10081	10085	10097	10104	10140	10174	10189
	10203	10229	10234	10317	10351	10549	10765	10774	10798	10799	10800	10830	10862	10869	10875
	10970	10989	10994	11049	11056	11067	11097	11103	11116	11165	11193	11249	11269	11315	11330
	11367	11378	11408	11421	11441	11467	11472	11489	11498	11531	11549	11553	11563	11574	11578
	11616	11623	11726	11738	11771	11781	11784	11786	11818	11873	11875	11897	12035	12087	12093
	12171	12176	12200	12225	12333	12341	12362	12404	12407	12409	12411	12414	12426	12436	12445
	12488	12492	12525	12529	12557	12579	12630	12646	12660	12677	12692	12703	12713	12720	12724
	12739	12760	12769	12776	12784	12791	12795	12802	12810	12812	12832	12865	12867	12869	13043
	13062	13074	13092	13097	13098	13100	13101	13103	13106	13108	13109	13111	13112	13114	13115
	13116	13140	13184	13186	13228	13231	13302	13304	13328	13330	13335	13337	13352	13390	13395
	13397	13426	13450	13466	13473	13475	13483	13509	13510	13519	13537	13538	13642	13645	13662
	13683	13684	13720	13856	13869	13870	13887	13903	13904	13943	13960	13961	13977	13978	13990
	13997	13998	14085	14158	14271	14286	14291	14296	14301	14320	14644	14649	14693	14699	14821
	14842	14906	14917	15191	15195	15207	19467	19474	19485	19491	19496	19539	19546	19557	19563
	19568	20465	20688	20690	20691	20726	20730	20812	20854	20896	20910	20911			
TSTB	6406	8862	9063	9131	9317	9480	9517	9534	9888	9956	9977	10022	10032	13557	14163
	14334	14373	14412	14448	14819	14936	20389	20391	20394	20450	20689	20690			
WAIT	13957														
.ASCII	7311	7313	7315	7317	7319	7321	7323	7325	7327	7329	7331	7333	7335	7337	7339
	7341	7343	7345	7347	7349	7351	7353	7355	7357	7359	7361	7363	7365	7367	7369
	7371	7373	7375	7377	7379	7381	7383	7385	7387	7389	7391	7393	7395	7397	7399



	7401	7403	7405	7407	7409	7411	7413	7415	7417	7419	7421	7423	7425	7427	7429
	7431	7433	7435	7437	7439	7441	7443	7445	7447	7449	7451	7453	7455	7457	7459
	7461	7463	7465	7467	7469	7471	7473	7475	7477	7479	7481	7483	7485	7487	7489
	7491	7493	7495	7497	7499	7502	7504	20688							
.ASCIZ	8151	8152	8153	8154	8155	8156	8157	8158	8159	8160	8161	8162	8163	8164	8165
	8166	8167	8168	8169	8170	8171	8172	8173	8174	8175	8176	8177	8179	8180	8181
	8182	8183	8184	8185	8186	8187	8188	8189	8190	8191	8192	8193	8194	8195	8196
	8197	8198	8199	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210	8211
	8212	8213	8214	8215	8216	8217	8218	8219	8220	8221	8222	8223	8224	8225	8226
	8227	8228	8229	8230	8231	8232	8233	8234	8235	8236	8237	8238	8239	8240	8241
	8242	8243	8244	8245	8246	8247	8248	8249	8250	8251	8252	8253	8254	8255	8256
	8257	8258	8259	8260	8261	8263	8268	8269	8270	8271	8273	8278	8279	8280	8281
	8285	8286	8288	8293	8294	8295	8296	8297	20688	20692					
.BLKB	6856	6858	9551	20689	20992										
.BLKW	6826	6879	8372	10449	10450	10451	10452	10453	10455	10465	10466	10467	10468	10469	10470
	10471	10472	10473	10485	14664	20982	20984								
.BYTE	6405	6406	6708	6709	6710	6711	6712	6713	6714	6715	6716	6717	6718	6719	6720
	6721	6722	6723	6724	6725	6726	6727	6728	6729	6730	6731	6732	6733	6734	6735
	6736	6737	6738	6739	6740	6741	6742	6743	6744	6745	6746	6747	6748	6749	6750
	6751	6752	6753	6754	6755	6756	6757	6855	6857	6862	6863	6864	6865	6866	6867
	6868	6869	6886	6887	6888	6889	7254	7255	7256	7257	7258	7259	7260	7261	7262
	7263	7264	7265	7266	7267	7268	7269	7270	7271	7272	7273	7274	7275	7276	7277
	7278	7279	7280	7281	7282	7283	7284	7285	7286	7287	7288	7289	7290	7291	7292
	7293	7294	7295	7296	7297	7298	7299	7300	7301	7302	7303	9396	9397	9398	9399
	9400	9401	9402	9403	9404	9405	9406	9407	9408	9409	9410	9411	9412	9413	9414
	9415	9590	9591	9592	9593	9594	9595	9596	9597	9598	9599	9600	9601	9602	9603
	9604	9605	9606	9607	9608	9609	9610	9611	9612	9613	9614	9615	9616	9617	9618
	9619	9620	14597	14598	14599	14600	14879	14880	14881	14882	14885	14886	14887	14888	14889
	14890	14891	14892	14893	14894	14895	14896	14897	14898	14899	14900	20689	20690		
.ENABL	4	6235													
.END	20997														
.ENDC	5785	6234	6351	6352	6355	6356	6384	6389	6394	6396	6397	6400	6401	6404	6405
	6406	6407	6529	6541	7501	7524	7530	7543	7549	7571	7575	7586	7590	7595	7598
	7608	7614	7627	7630	7634	7637	7648	7651	7659	7662	7687	7694	7709	7714	7735
	7741	7771	8264	8267	8274	8277	8284	8289	8292	10756	10762	10765	10774	10789	10796
	10802	10821	10855	10864	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968
	11001	11026	11030	11035	11058	11061	11066	12229	12289	12291	12428	12429	12438	12440	12441
	12450	12453	12798	12801	12822	12826	12840	12843	12847	12850	12859	12862	12882	13206	13209
	13210	13215	13387	13401	13404	13409	13412	13417	13425	13431	13434	13438	13441	13449	13454
	13457	13461	13464	13505	13506	13528	13532	13533	13548	13549	13585	13621	13624	13625	13655
	13658	13670	13673	13674	13675	13693	13696	13697	13704	13707	13724	13727	13880	13927	13930
	13931	14111	14114	14197	14265	14267	14268	14269	14270	14273	14279	14281	14288	14289	14293
	14298	14299	14314	14317	14318	14733	14744	14746	14757	14759	14764	14778	14816	14844	14846
	14847	14860	14875	15004	15007	15010	15013	15016	15019	15022	15025	15028	15031	15034	15037
	15040	15043	15046	15049	15052	15055	15058	15061	15065	15068	15072	15075	15079	15082	15086
	15089	15093	15096	15100	15103	15107	15110	15114	15117	15121	15124	15128	15131	15135	15138
	15142	15145	15149	15152	15156	15159	15163	15166	15170	15173	15176	15179	15189	15190	15193
	15206	15214	15216	20683	20688	20689	20690	20691	20692	20693	20806	20848	20890	20975	20978
.EQUIV	6356														
.EVEN	6405	6406	8332	9621	12457	14476	15221	20692	20996						
.IF	5132	5786	6350	6351	6353	6356	6384	6385	6390	6396	6397	6399	6400	6402	6405
	6406	6407	6518	6530	7310	7519	7525	7538	7544	7568	7572	7583	7587	7593	7596
	7603	7609	7625	7628	7632	7635	7646	7649	7657	7660	7681	7688	7705	7710	7730
	7736	7744	8262	8265	8272	8275	8282	8287	8290	10753	10758	10765	10774	10789	10792

	10747	10802	10848	10861	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968
	10991	11002	11027	11032	11052	11058	11062	12229	12289	12291	12425	12428	12438	12440	12441
	12448	12451	12796	12799	12818	12824	12837	12841	12845	12848	12856	12860	12864	13206	13209
	13210	13215	13384	13399	13402	13407	13410	13415	13423	13429	13432	13436	13439	13447	13452
	13455	13459	13462	13505	13506	13528	13532	13533	13548	13549	13580	13621	13624	13625	13655
	13658	13670	13673	13674	13675	13693	13696	13697	13704	13707	13724	13727	13876	13927	13930
	13931	14109	14112	14197	14265	14267	14268	14269	14270	14273	14279	14281	14288	14289	14293
	14298	14299	14314	14317	14318	14733	14744	14746	14757	14759	14764	14778	14816	14844	14846
	14847	14860	14875	15004	15007	15010	15013	15016	15019	15022	15025	15028	15031	15034	15037
	15040	15043	15046	15049	15052	15055	15058	15061	15065	15068	15072	15075	15079	15082	15086
	15089	15093	15096	15100	15103	15107	15110	15114	15117	15121	15124	15128	15131	15135	15138
	15142	15145	15149	15152	15156	15159	15163	15166	15170	15173	15176	15179	15189	15190	15193
.IFF	15206	15214	15216	15319	20683	20688	20689	20690	20691	20692	20806	20848	20890	20973	20976
	6397	6400	6405	6406	6407	10765	10774	10789	10802	10877	10882	10884	10891	10930	10932
	10933	10946	10955	10958	10968	11058	12229	12289	12291	12428	12438	12440	12441	13206	13209
	13210	13215	13505	13506	13528	13532	13533	13548	13549	13621	13624	13625	13655	13658	13670
	13673	13674	13675	13693	13696	13697	13704	13707	13724	13727	13927	13930	13931	14197	14265
	14267	14268	14269	14270	14273	14279	14281	14288	14289	14293	14298	14299	14314	14317	14318
	14733	14744	14746	14757	14759	14764	14778	14816	14844	14846	14847	14860	14875	15004	15007
	15010	15013	15016	15019	15022	15025	15028	15031	15034	15037	15040	15043	15046	15049	15052
	15055	15058	15061	15065	15068	15072	15075	15079	15082	15086	15089	15093	15096	15100	15103
	15107	15110	15114	15117	15121	15124	15128	15131	15135	15138	15142	15145	15149	15152	15156
	15159	15163	15166	15170	15173	15176	15179	15189	15190	15193	15206	15214	15216	20683	20688
	20689	20690	20691	20692	20806	20848	20890								
.IFT	19283	20689													
.IFTF	19271	20578	20689												
.IIF	6351	6384	6405	10765	10774	10789	10802	10877	10882	10884	10891	10930	10932	10933	10946
	10955	10958	10968	11058	12229	12289	12291	12428	12438	12440	12441	13206	13209	13210	13215
	13505	13506	13528	13532	13533	13548	13549	13621	13624	13625	13655	13658	13670	13673	13674
	13675	13693	13696	13697	13704	13707	13724	13727	13927	13930	13931	14197	14265	14267	14268
	14269	14270	14273	14279	14281	14288	14289	14293	14298	14299	14314	14317	14318	14733	14744
	14746	14757	14759	14764	14778	14816	14844	14846	14847	14860	14875	15004	15007	15010	15013
	15016	15019	15022	15025	15028	15031	15034	15037	15040	15043	15046	15049	15052	15055	15058
	15061	15065	15068	15072	15075	15079	15082	15086	15089	15093	15096	15100	15103	15107	15110
	15114	15117	15121	15124	15128	15131	15135	15138	15142	15145	15149	15152	15156	15159	15163
	15166	15170	15173	15176	15179	15189	15190	15193	15206	15214	15216	20688	20689	20690	20691
.IRP	20806	20848	20890												
	6396	6406	10789	10802	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968
	11058	12229	12289	12291	12428	12438	12440	12441	13206	13209	13210	13215	13505	13506	13528
	13532	13533	13548	13549	13621	13624	13625	13655	13658	13670	13673	13674	13675	13693	13696
	13697	13704	13707	13724	13727	13927	13930	13931	14197	14265	14267	14268	14269	14270	14273
	14279	14281	14288	14289	14293	14298	14299	14314	14317	14318	14733	14744	14746	14757	14759
	14764	14778	14816	14844	14846	14847	14860	14875	15004	15007	15010	15013	15016	15019	15022
	15025	15028	15031	15034	15037	15040	15043	15046	15049	15052	15055	15058	15061	15065	15068
	15072	15075	15079	15082	15086	15089	15093	15096	15100	15103	15107	15110	15114	15117	15121
	15124	15128	15131	15135	15138	15142	15145	15149	15152	15156	15159	15163	15166	15170	15173
	15176	15179	15189	15190	15193	15206	15214	15216	20688	20692	20806	20848	20890		
.LIST	2	5126	5130	6345	6349	6356	6384	6396	6405	10765	10774	10779	10780	10789	10802
	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968	11058	12229	12289	12291
	12428	12438	12440	12441	13206	13209	13210	13215	13505	13506	13528	13532	13533	13548	13549
	13621	13624	13625	13655	13658	13670	13673	13674	13675	13693	13696	13697	13704	13707	13724
	13727	13927	13930	13931	14197	14265	14267	14268	14269	14270	14273	14279	14281	14288	14289
	14293	14298	14299	14314	14317	14318	14733	14744	14746	14757	14759	14764	14778	14816	14844
	14846	14847	14860	14875	15004	15007	15010	15013	15016	15019	15022	15025	15028	15031	15034

CZKEEB.P11	MACY11	27(655)	29-SEP-80	09:22	PAGE	183-69	SEQUENCE	457	15037	15040	15043	15046	15049	15052	15055	15058	15061	15065	15068	15072	15075	15079	15082																							
.MACRO	39	81	166	301	482	898	929	1000	1157	1203	1217	1245	1338	1364	1395	1443	1455	1499	1533	1566	1579	1600	1613	1646	1695	1741	1778	1825	1858	1888																
.MCALL	1944	1952	2004	2214	2422	2615	2703	2838	2935	3013	3098	3314	3406	3482	3582	3713	3776	3838	3962	4000	4064	4169	4254	4293	4356	4394	4437	4535	4583	4853																
.MEXIT	6405	10789	10802	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968	11058	12229	12289	12291	12428	12438	12440	12441	13206	13209	13210	13215	13505	13506	13528	13532																
.NLIST	1	3	5127	6236	6344	6356	6384	6396	6405	10765	10774	10777	10778	10789	10802	10877	10882	10884	10891	10930	10932	10933	10946	10955	10958	10968	11058	12229	12289	12291	12428	12438	12440	12441	13206	13209	13210	13215	13505	13506	13528	13532				
.PAGE	6408	6420	6430	6487	6793	6838	6871	6882	7084	7116	7149	7181	7211	7249	7510	8144	8300	8370	8469	8572	8620	8666	8693	8708	8723	8734	8972	9126	9149	9364	9622	9745	9813	9881	9900	10147	10406	10446	10493	10748	13029	13129	13218	13578	13909	
.REPT	8303	8334	8371	8470	8518	8573	8621	8667	8694	8709	8724	8735	8756	8973	9127	14066	14252	15260	15318	15363	15401	15476	15513	15638	15724	15763	15846	15886	15961	15996	16125	16209	16247	16331	16369	16451	16530	16750	16824	16866	16975	17015	17227	17442	17701	
.SBTTL	9150	9418	9623	9746	9814	9901	10148	10281	10407	10447	10494	10627	10749	13030	13130	17918	17998	18040	18082	18127	18207	18252	18297	18345	18432	18558	18811	18972	19150	19272	20781	20820	20862	20905	20684	20686	20689	20691	20695	20696	20806	20848	20890	7249	7510	
	13219	13579	13910	14067	14253	15000	15227	15255	15261	15290	15320	15439	15551	15680	15802	20781	20820	20862	20905	20684	20686	20689	20691	20695	20696	20806	20848	20890	7249	7510	8301	8973	9127	13030	13130	15802	15846	15886	15961	15996	17015	17227	17442	17701	18972	19150



.TITLE	15925	16034	16165	16287	16409	16490	16604	16714	16787	16938	17051	17268	17523	17742	17959
.WORD	18166	18384	18510	18633	18973	19279	20688	20689	20690	20691	20692	20783	20979	20989	
6351	6384	6397	6400	6405	6406	6413	6414	6415	6416	6425	6426	6427	6428	6435	6436
6437	6438	6439	6440	6441	6442	6443	6444	6445	6446	6447	6448	6449	6450	6451	6451
6452	6453	6454	6455	6456	6457	6458	6459	6460	6461	6462	6469	6470	6471	6472	6472
6473	6474	6475	6476	6477	6478	6479	6480	6481	6482	6483	6484	6492	6493	6494	6494
6495	6496	6497	6498	6499	6500	6501	6502	6503	6504	6505	6506	6507	6508	6509	6509
6510	6511	6512	6519	6520	6521	6522	6523	6524	6525	6542	6544	6545	6546	6550	6550
6551	6552	6553	6554	6555	6556	6567	6568	6569	6570	6571	6572	6573	6574	6575	6575
6581	6582	6583	6584	6585	6586	6587	6588	6589	6590	6591	6592	6593	6594	6595	6595
6596	6597	6598	6599	6600	6601	6602	6603	6604	6605	6606	6607	6608	6609	6610	6610
6611	6612	6613	6614	6615	6616	6617	6618	6619	6620	6621	6622	6623	6624	6625	6625
6626	6627	6628	6629	6630	6631	6632	6633	6634	6635	6636	6637	6638	6639	6640	6640
6641	6642	6643	6644	6645	6646	6647	6648	6649	6650	6651	6652	6654	6655	6657	6657
6658	6659	6660	6661	6662	6663	6664	6665	6666	6667	6668	6669	6670	6671	6672	6672
6673	6674	6675	6676	6677	6678	6679	6680	6681	6682	6683	6684	6685	6686	6687	6687
6688	6689	6690	6691	6692	6693	6694	6695	6696	6697	6698	6699	6700	6701	6702	6702
6703	6704	6705	6706	6758	6759	6760	6761	6762	6763	6764	6765	6766	6767	6768	6768
6769	6770	6771	6772	6776	6777	6778	6779	6780	6781	6782	6783	6784	6785	6786	6786
6787	6788	6789	6791	6792	6796	6797	6798	6799	6800	6801	6802	6803	6804	6805	6805
6807	6808	6809	6810	6811	6812	6813	6814	6815	6816	6818	6819	6820	6821	6822	6822
6823	6824	6825	6829	6830	6831	6832	6833	6834	6835	6841	6842	6843	6844	6845	6845
6846	6847	6875	6876	6877	6878	6891	6892	6893	6895	6896	6897	6899	6900	6901	6901
6903	6904	6905	6906	6907	6909	6910	6911	6913	6914	6915	6916	6917	6919	6920	6920
6921	6923	6924	6925	6926	6927	6929	6930	6931	6933	6934	6935	6936	6937	6939	6939
6940	6941	6942	6943	6944	6945	6946	6948	6949	6950	6951	6952	6953	6954	6955	6955
6957	6958	6959	6960	6961	6962	6963	6964	6966	6967	6968	6969	6970	6971	6972	6972
6973	6974	6975	6976	6977	6978	6979	6980	6981	6983	6984	6985	6986	6987	6988	6988
6989	6990	6991	6992	6993	6994	6995	6996	6997	6998	7000	7001	7002	7003	7004	7004
7005	7006	7007	7008	7009	7010	7011	7012	7013	7014	7015	7017	7018	7019	7021	7021
7022	7023	7025	7026	7027	7028	7035	7036	7037	7038	7039	7040	7041	7042	7047	7047
7048	7049	7050	7051	7052	7053	7057	7062	7063	7064	7065	7066	7067	7068	7073	7073
7074	7075	7076	7077	7078	7079	7083	7088	7089	7090	7091	7092	7093	7094	7095	7095
7096	7097	7098	7099	7100	7101	7102	7103	7104	7105	7106	7107	7108	7109	7110	7110
7111	7112	7113	7114	7115	7120	7121	7122	7123	7124	7125	7126	7127	7128	7129	7129
7130	7131	7132	7133	7134	7135	7136	7137	7138	7139	7140	7141	7142	7143	7144	7144
7145	7146	7147	7153	7154	7155	7156	7157	7158	7159	7160	7161	7162	7163	7164	7164
7165	7166	7167	7168	7169	7170	7171	7172	7173	7174	7175	7176	7177	7178	7179	7179
7180	7185	7186	7187	7188	7189	7190	7191	7192	7193	7194	7195	7196	7197	7198	7198
7199	7200	7201	7202	7203	7204	7205	7206	7207	7208	7209	7210	7215	7217	7218	7218
7219	7220	7221	7222	7223	7224	7225	7226	7227	7228	7229	7230	7231	7232	7233	7233
7234	7235	7236	7237	7238	7239	7240	7241	7242	7243	7244	7245	7246	7247	7248	7248
7253	7312	7314	7316	7318	7320	7322	7324	7326	7328	7330	7332	7334	7336	7338	7338
7340	7342	7344	7346	7348	7350	7352	7354	7356	7358	7360	7362	7364	7366	7368	7368
7370	7372	7374	7376	7378	7380	7382	7384	7386	7388	7390	7392	7394	7396	7398	7398
7400	7402	7404	7406	7408	7410	7412	7414	7416	7418	7420	7422	7424	7426	7428	7428
7430	7432	7434	7436	7438	7440	7442	7444	7446	7448	7450	7452	7454	7456	7458	7458
7460	7462	7464	7466	7468	7470	7472	7474	7476	7478	7480	7482	7484	7486	7488	7488
7490	7492	7494	7496	7498	7500	7503	7505	7506	7507	7508	7509	7520	7521	7522	7522
7523	7531	7532	7539	7540	7541	7542	7550	7551	7552	7553	7554	7555	7556	7557	7557
7558	7559	7569	7570	7576	7577	7584	7585	7591	7592	7594	7599	7604	7605	7606	7606
7607	7615	7616	7624	7626	7631	7633	7638	7639	7643	7644	7645	7647	7652	7656	7656
7658	7663	7664	7665	7666	7667	7680	7682	7683	7684	7685	7686	7695	7696	7697	7697

7698	7699	7704	7706	7707	7708	7715	7716	7717	7718	7719	7720	7729	7731	7732
7733	7734	7742	7743	7779	7780	7781	7782	7783	7784	7785	7786	7787	7788	7789
7790	7791	7792	7793	7794	7795	7796	7797	7798	7799	7800	7801	7802	7808	7809
7810	7811	7812	7813	7814	7815	7816	7817	7818	7819	7820	7821	7822	7823	7824
7825	7826	7827	7828	7829	7830	7831	7832	7839	7840	7841	7842	7843	7844	7845
7846	7847	7848	7849	7850	7851	7852	7853	7854	7855	7856	7857	7858	7859	7860
7861	7862	7863	7868	7869	7870	7871	7872	7873	7874	7875	7876	7877	7878	7879
7880	7881	7882	7883	7884	7885	7886	7887	7888	7889	7890	7891	7892	7899	7900
7901	7902	7903	7904	7905	7906	7907	7908	7909	7910	7911	7912	7913	7914	7915
7916	7917	7918	7919	7920	7926	7927	7928	7929	7930	7931	7932	7933	7934	7935
7936	7937	7938	7939	7940	7941	7942	7943	7944	7945	7946	7947	7948	7949	7958
7959	7960	7961	7962	7963	7964	7965	7966	7967	7968	7969	7970	7971	7972	7973
7974	7975	7976	7987	7988	7989	7990	7991	7992	7993	7994	7995	7996	7997	7998
7999	8000	8001	8002	8003	8004	8005	8006	8007	8008	8009	8010	8011	8012	8013
8014	8021	8022	8023	8024	8025	8026	8027	8028	8029	8030	8031	8032	8033	8034
8035	8036	8037	8038	8039	8040	8041	8042	8043	8044	8045	8052	8053	8054	8055
8056	8057	8058	8059	8060	8061	8062	8063	8064	8065	8066	8067	8068	8069	8070
8071	8072	8073	8074	8081	8082	8083	8084	8085	8086	8087	8088	8089	8090	8091
8092	8093	8094	8095	8096	8097	8098	8099	8100	8109	8110	8111	8112	8113	8114
8115	8116	8117	8118	8119	8120	8121	8122	8123	8124	8125	8131	8132	8133	8134
8135	8136	8137	8138	8139	8140	8141	8142	8143	8373	8374	8375	8376	8377	8378
8379	8380	8381	8382	8383	8384	8385	8386	8387	8388	8389	8390	8391	8392	8393
8394	8395	8396	8397	8398	8399	8400	8401	8402	8403	8404	8405	8406	8407	8408
8409	8410	8411	8412	8413	8414	8415	8416	8417	8418	8419	8420	8421	8422	8423
8424	8425	8426	8427	8428	8429	8430	8431	8432	8433	8434	8435	8436	8437	8438
8439	8440	8441	8442	8443	8444	8445	8446	8447	8448	8449	8450	8451	8452	8453
8454	8455	8456	8457	8458	8459	8460	8461	8462	8463	8464	8465	8466	8467	8468
9365	9366	9367	9368	9369	9370	9371	9372	9373	9374	9375	9376	9377	9378	9379
9380	9381	9382	9383	9384	9385	9386	9387	9388	9389	9390	9391	9392	9393	9394
9416	9550	9552	9553	9554	9555	9556	9557	9558	9559	9560	9561	9562	9563	9564
9565	9566	9567	9568	9569	9570	9571	9572	9573	9574	9575	9576	9577	9578	9579
9580	9581	9582	9583	9584	9585	9586	9587	9588	9589	9743	9744	9898	9899	10120
10121	10122	10123	10124	10125	10126	10127	10128	10272	10273	10274	10275	10276	10277	10278
10279	10454	10456	10457	10458	10459	10460	10461	10462	10463	10464	10474	10475	10476	10477
10478	10479	10480	10481	10482	10483	10484	10486	10487	10488	10489	10490	10491	10492	10622
10623	10624	11245	11246	11247	12255	14245	14246	14247	14248	14249	14250	14379	14970	14971
14972	14973	14974	14975	14976	14977	14978	14979	14980	14981	14982	14983	14984	14985	14986
14987	14988	14989	14990	14991	14992	14993	14994	14995	14996	14997	15230	15231	15232	15233
15234	15235	15236	15237	15238	15239	15240	15241	15242	15243	15244	15245	15246	15247	15248
15249	15250	15251	15258	15265	15266	15267	15268	15269	15270	15271	15272	15273	15274	15275
15276	15277	15278	15279	15280	15281	15282	15283	15284	15285	15286	15294	15295	15296	15297
15298	15299	15300	15301	15302	15303	15304	15305	15306	15307	15308	15309	15310	15311	15312
15313	15314	15315	15323	15324	15325	15326	15327	15328	15329	15330	15331	15332	15333	15334
15335	15336	15337	15338	15339	15340	15341	15342	15343	15344	15366	15367	15368	15369	15370
15371	15372	15373	15374	15375	15376	15377	15378	15379	15380	15381	15382	15383	15384	15385
15386	15387	15404	15405	15406	15407	15408	15409	15410	15411	15412	15413	15414	15415	15416
15417	15418	15419	15420	15421	15422	15423	15424	15425	15443	15444	15445	15446	15447	15448
15449	15450	15451	15452	15453	15454	15455	15456	15457	15458	15459	15460	15461	15462	15463
15464	15480	15481	15482	15483	15484	15485	15486	15487	15488	15489	15490	15491	15492	15493
15494	15495	15496	15497	15498	15499	15500	15501	15517	15518	15519	15520	15521	15522	15523
15524	15525	15526	15527	15528	15529	15530	15531	15532	15533	15534	15535	15536	15537	15538
15555	15556	15557	15558	15559	15560	15561	15562	15563	15564	15565	15566	15567	15568	15569
15570	15571	15572	15573	15574	15575	15576	15596	15597	15598	15599	15600	15601	15602	15603
15604	15605	15606	15607	15608	15609	15610	15611	15612	15613	15614	15615	15616	15617	15642

15643	15644	15645	15646	15647	15648	15649	15650	15651	15652	15653	15654	15655	15656	15657
15658	15659	15660	15661	15662	15663	15664	15665	15666	15667	15668	15669	15670	15671	15672
15693	15694	15695	15696	15697	15698	15699	15700	15701	15702	15703	15704	15705	15728	15729
15730	15731	15732	15733	15734	15735	15736	15737	15738	15739	15740	15741	15742	15743	15744
15745	15746	15747	15748	15749	15767	15768	15769	15770	15771	15772	15773	15774	15775	15776
15777	15778	15779	15780	15781	15782	15783	15784	15785	15786	15787	15788	15805	15806	15807
15808	15809	15810	15811	15812	15813	15814	15815	15816	15817	15818	15819	15820	15821	15822
15823	15824	15825	15826	15849	15850	15851	15852	15853	15854	15855	15856	15857	15858	15859
15860	15861	15862	15863	15864	15865	15866	15867	15868	15869	15870	15889	15890	15891	15892
15893	15894	15895	15896	15897	15898	15899	15900	15901	15902	15903	15904	15905	15906	15907
15908	15909	15910	15928	15929	15930	15931	15932	15933	15934	15935	15936	15937	15938	15939
15940	15941	15942	15943	15944	15945	15946	15947	15948	15949	15964	15965	15966	15967	15968
15969	15970	15971	15972	15973	15974	15975	15976	15977	15978	15979	15980	15981	15982	15983
15984	15985	16000	16001	16002	16003	16004	16005	16006	16007	16008	16009	16010	16011	16012
16013	16014	16015	16016	16017	16018	16019	16020	16021	16038	16039	16040	16041	16042	16043
16044	16045	16046	16047	16048	16049	16050	16051	16052	16053	16054	16055	16056	16057	16058
16059	16084	16085	16086	16087	16088	16089	16090	16091	16092	16093	16094	16095	16096	16097
16098	16099	16100	16101	16102	16103	16104	16105	16129	16130	16131	16132	16133	16134	16135
16136	16137	16138	16139	16140	16141	16142	16143	16144	16145	16146	16147	16148	16149	16150
16170	16171	16172	16173	16174	16175	16176	16177	16178	16179	16180	16181	16182	16183	16184
16185	16186	16187	16188	16189	16190	16191	16213	16214	16215	16216	16217	16218	16219	16220
16221	16222	16223	16224	16225	16226	16227	16228	16229	16230	16231	16232	16233	16234	16251
16252	16253	16254	16255	16256	16257	16258	16259	16260	16261	16262	16263	16264	16265	16266
16267	16268	16269	16270	16271	16272	16292	16293	16294	16295	16296	16297	16298	16299	16300
16301	16302	16303	16304	16305	16306	16307	16308	16309	16310	16311	16312	16313	16335	16336
16337	16338	16339	16340	16341	16342	16343	16344	16345	16346	16347	16348	16349	16350	16351
16352	16353	16354	16355	16356	16373	16374	16375	16376	16377	16378	16379	16380	16381	16382
16383	16384	16385	16386	16387	16388	16389	16390	16391	16392	16393	16394	16414	16415	16416
16417	16418	16419	16420	16421	16422	16423	16424	16425	16426	16427	16428	16429	16430	16431
16432	16433	16434	16435	16454	16455	16456	16457	16458	16459	16460	16461	16462	16463	16464
16465	16466	16467	16468	16469	16470	16471	16472	16473	16474	16475	16493	16494	16495	16496
16497	16498	16499	16500	16501	16502	16503	16504	16505	16506	16507	16508	16509	16510	16511
16512	16513	16514	16533	16534	16535	16536	16537	16538	16539	16540	16541	16542	16543	16544
16545	16546	16547	16548	16549	16550	16551	16552	16553	16554	16570	16571	16572	16573	16574
16575	16576	16577	16578	16579	16580	16581	16582	16583	16584	16585	16586	16587	16588	16589
16590	16591	16607	16608	16609	16610	16611	16612	16613	16614	16615	16616	16617	16618	16619
16620	16621	16622	16623	16624	16625	16626	16627	16628	16644	16645	16646	16647	16648	16649
16650	16651	16652	16653	16654	16655	16656	16657	16658	16659	16660	16661	16662	16663	16664
16665	16680	16681	16682	16683	16684	16685	16686	16687	16688	16689	16690	16691	16692	16693
16694	16695	16696	16697	16698	16699	16700	16701	16717	16718	16719	16720	16721	16722	16723
16724	16725	16726	16727	16728	16729	16730	16731	16732	16733	16734	16735	16736	16737	16738
16753	16754	16755	16756	16757	16758	16759	16760	16761	16762	16763	16764	16765	16766	16767
16768	16769	16770	16771	16772	16773	16774	16790	16791	16792	16793	16794	16795	16796	16797
16798	16799	16800	16801	16802	16803	16804	16805	16806	16807	16808	16809	16810	16811	16829
16830	16831	16832	16833	16834	16835	16836	16837	16838	16839	16840	16841	16842	16843	16844
16845	16846	16847	16848	16849	16850	16869	16870	16871	16872	16873	16874	16875	16876	16877
16878	16879	16880	16881	16882	16883	16884	16885	16886	16887	16888	16889	16890	16904	16905
16906	16907	16908	16909	16910	16911	16912	16913	16914	16915	16916	16917	16918	16919	16920
16921	16922	16923	16924	16925	16941	16942	16943	16944	16945	16946	16947	16948	16949	16950
16951	16952	16953	16954	16955	16956	16957	16958	16959	16960	16961	16962	16980	16981	16982
16983	16984	16985	16986	16987	16988	16989	16990	16991	16992	16993	16994	16995	16996	16997
16998	16999	17000	17001	17018	17019	17020	17021	17022	17023	17024	17025	17026	17027	17028
17029	17030	17031	17032	17033	17034	17035	17036	17037	17038	17039	17054	17055	17056	17057
17058	17059	17060	17061	17062	17063	17064	17065	17066	17067	17069	17070	17071	17072	17073

17074	17075	17076	17095	17096	17097	17098	17099	17100	17101	17102	17103	17104	17105	17106
17107	17108	17110	17111	17112	17113	17114	17115	17116	17117	17139	17140	17141	17142	17143
17144	17145	17146	17147	17148	17149	17150	17151	17152	17154	17155	17156	17157	17158	17159
17160	17161	17183	17184	17185	17186	17187	17188	17189	17190	17191	17192	17193	17194	17195
17196	17198	17199	17200	17201	17202	17203	17204	17205	17230	17231	17232	17233	17234	17235
17236	17237	17238	17239	17240	17241	17242	17243	17245	17246	17247	17248	17249	17250	17251
17252	17271	17272	17273	17274	17275	17276	17277	17278	17279	17280	17281	17282	17283	17284
17286	17287	17288	17289	17290	17291	17292	17293	17312	17313	17314	17315	17316	17317	17318
17319	17320	17321	17322	17323	17324	17325	17327	17328	17329	17330	17331	17332	17333	17334
17355	17356	17357	17358	17359	17360	17361	17362	17363	17364	17365	17366	17367	17368	17370
17371	17372	17373	17374	17375	17376	17377	17398	17399	17400	17401	17402	17403	17404	17405
17406	17407	17408	17409	17410	17411	17413	17414	17415	17416	17417	17418	17419	17420	17445
17446	17447	17448	17449	17450	17451	17452	17453	17454	17455	17456	17457	17458	17460	17461
17462	17463	17464	17465	17466	17467	17485	17486	17487	17488	17489	17490	17491	17492	17493
17494	17495	17496	17497	17498	17500	17501	17502	17503	17504	17505	17506	17507	17526	17527
17528	17529	17530	17531	17532	17533	17534	17535	17536	17537	17538	17539	17541	17542	17543
17544	17545	17546	17547	17548	17569	17570	17571	17572	17573	17574	17575	17576	17577	17578
17579	17580	17581	17582	17584	17585	17586	17587	17588	17589	17590	17591	17613	17614	17615
17616	17617	17618	17619	17620	17621	17622	17623	17624	17625	17626	17628	17629	17630	17631
17632	17633	17634	17635	17657	17658	17659	17660	17661	17662	17663	17664	17665	17666	17667
17668	17669	17670	17672	17673	17674	17675	17676	17677	17678	17679	17704	17705	17706	17707
17708	17709	17710	17711	17712	17713	17714	17715	17716	17717	17719	17720	17721	17722	17723
17724	17725	17726	17745	17746	17747	17748	17749	17750	17751	17752	17753	17754	17755	17756
17757	17758	17760	17761	17762	17763	17764	17765	17766	17767	17786	17787	17788	17789	17790
17791	17792	17793	17794	17795	17796	17797	17798	17799	17801	17802	17803	17804	17805	17806
17807	17808	17831	17832	17833	17834	17835	17836	17837	17838	17839	17840	17841	17842	17843
17844	17846	17847	17848	17849	17850	17851	17852	17853	17874	17875	17876	17877	17878	17879
17880	17881	17882	17883	17884	17885	17886	17887	17889	17890	17891	17892	17893	17894	17895
17896	17921	17922	17923	17924	17925	17926	17927	17928	17929	17930	17931	17932	17933	17934
17936	17937	17938	17939	17940	17941	17942	17943	17962	17963	17964	17965	17966	17967	17968
17969	17970	17971	17972	17973	17974	17975	17976	17977	17978	17979	17980	17981	17982	17983
18002	18003	18004	18005	18006	18007	18008	18009	18010	18011	18012	18013	18014	18015	18016
18017	18018	18019	18020	18021	18022	18023	18044	18045	18046	18047	18048	18049	18050	18051
18052	18053	18054	18055	18056	18057	18058	18059	18060	18061	18062	18063	18064	18065	18086
18087	18088	18089	18090	18091	18092	18093	18094	18095	18096	18097	18098	18099	18100	18101
18102	18103	18104	18105	18106	18107	18130	18131	18132	18133	18134	18135	18136	18137	18138
18139	18140	18141	18142	18143	18144	18145	18146	18147	18148	18149	18150	18151	18169	18170
18171	18172	18173	18174	18175	18176	18177	18178	18179	18180	18181	18182	18183	18184	18185
18186	18187	18188	18189	18190	18211	18212	18213	18214	18215	18216	18217	18218	18219	18220
18221	18222	18223	18224	18225	18226	18227	18228	18229	18230	18231	18232	18232	18256	18257
18259	18260	18261	18262	18263	18264	18265	18266	18267	18268	18269	18270	18271	18272	18273
18274	18275	18276	18277	18301	18302	18303	18304	18305	18306	18307	18308	18309	18310	18311
18312	18313	18314	18315	18316	18317	18318	18319	18320	18321	18322	18348	18349	18350	18351
18352	18353	18354	18355	18356	18357	18358	18359	18360	18361	18362	18363	18364	18365	18366
18367	18368	18369	18387	18388	18389	18390	18391	18392	18393	18394	18395	18396	18397	18398
18399	18400	18401	18402	18403	18404	18405	18406	18407	18408	18435	18436	18437	18438	18439
18440	18441	18442	18443	18444	18445	18446	18447	18448	18449	18450	18451	18452	18453	18454
18455	18456	18474	18475	18476	18477	18478	18479	18480	18481	18482	18483	18484	18485	18486
18487	18488	18489	18490	18491	18492	18493	18494	18495	18513	18514	18515	18516	18517	18518
18519	18520	18521	18522	18523	18524	18525	18526	18527	18528	18529	18530	18531	18532	18533
18534	18561	18562	18563	18564	18565	18566	18567	18568	18569	18570	18571	18572	18573	18574
18575	18576	18577	18578	18579	18580	18581	18582	18597	18598	18599	18600	18601	18602	18603
18604	18605	18606	18607	18608	18609	18610	18611	18612	18613	18614	18615	18616	18617	18618
18636	18637	18638	18639	18640	18641	18642	18643	18644	18645	18646	18647	18648	18649	18651

18652	18653	18654	18655	18656	18657	18658	18679	18680	18681	18682	18683	18684	18685	18686
18687	18688	18689	18690	18691	18692	18694	18695	18696	18697	18698	18699	18700	18701	18723
18724	18725	18726	18727	18728	18729	18730	18731	18732	18733	18734	18735	18736	18738	18739
18740	18741	18742	18743	18744	18745	18767	18768	18769	18770	18771	18772	18773	18774	18775
18776	18777	18778	18779	18780	18782	18783	18784	18785	18786	18787	18788	18789	18814	18815
18816	18817	18818	18819	18820	18821	18822	18823	18824	18825	18826	18827	18829	18830	18831
18832	18833	18834	18835	18836	18854	18855	18856	18857	18858	18859	18860	18861	18862	18863
18864	18865	18866	18867	18869	18870	18871	18872	18873	18874	18875	18876	18894	18895	18896
18897	18898	18899	18900	18901	18902	18903	18904	18905	18906	18907	18909	18910	18911	18912
18913	18914	18915	18916	18935	18936	18937	18938	18939	18940	18941	18942	18943	18944	18945
18946	18947	18948	18950	18951	18952	18953	18954	18955	18956	18957	18976	18977	18978	18979
18980	18981	18982	18983	18984	18985	18986	18987	18988	18989	18991	18992	18993	18994	18995
18996	18997	18998	19017	19018	19019	19020	19021	19022	19023	19024	19025	19026	19027	19028
19029	19030	19032	19033	19034	19035	19036	19037	19038	19039	19062	19063	19064	19065	19066
19067	19068	19069	19070	19071	19072	19073	19074	19075	19077	19078	19079	19080	19081	19082
19083	19084	19106	19107	19108	19109	19110	19111	19112	19113	19114	19115	19116	19117	19118
19119	19121	19122	19123	19124	19125	19126	19127	19128	19153	19154	19155	19156	19157	19158
19159	19160	19161	19162	19163	19164	19165	19166	19168	19169	19170	19171	19172	19173	19174
19175	19193	19194	19195	19196	19197	19198	19199	19200	19201	19202	19203	19204	19205	19206
19208	19209	19210	19211	19212	19213	19214	19215	19233	19234	19235	19236	19237	19238	19239
19240	19241	19242	19243	19244	19245	19246	19248	19249	19250	19251	19252	19253	19254	19255
19275	19276	19277	19287	19288	19291	19292	19293	19294	19295	19296	19297	19298	19300	19301
19302	19303	19304	19305	19306	19307	19308	19309	19311	19312	19313	19314	19315	19317	19318
19319	19320	19321	19323	19324	19325	19326	19328	19329	19330	19332	19333	19334	19335	19336
19337	19339	19340	19341	19343	19344	19345	19346	19347	19348	19350	19351	19352	19353	19354
19356	19357	19358	19360	19361	19362	19364	19365	19366	19368	19369	19370	19371	19372	19373
19374	19375	19377	19378	19379	19380	19382	19383	19384	19386	19387	19388	19391	19392	19394
19395	19396	19397	19398	19399	19400	19401	19404	19405	19406	19407	19408	19409	19410	19411
19414	19415	19416	19417	19418	19419	19420	19421	19424	19425	19426	19427	19428	19429	19430
19431	19434	19435	19436	19437	19440	19441	19442	19444	19445	19446	19447	19449	19450	19451
19454	19509	19526	19581	19589	19597	19598	19599	19600	19601	19604	19605	19606	19607	19608
19609	19610	19613	19614	19615	19616	19617	19618	19619	19620	19621	19624	19625	19626	19627
19629	19630	19631	19632	19633	19634	19635	19638	19639	19640	19641	19642	19643	19646	19647
19648	19649	19650	19651	19652	19655	19656	19657	19658	19659	19660	19661	19664	19665	19666
19667	19668	19669	19672	19673	19674	19675	19676	19679	19680	19681	19682	19683	19686	19687
19688	19689	19690	19691	19692	19693	19696	19697	19698	19700	19716	19728	19766	19795	19807
19837	19838	19839	19840	19841	19843	19844	19846	19847	19849	19850	19851	19852	19853	19854
19855	19857	19858	19859	19860	19861	19862	19864	19865	19866	19867	19868	19869	19870	19873
19874	19875	19876	19877	19878	19881	19882	19883	19884	19885	19886	19887	19890	19891	19892
19893	19894	19895	19896	19897	19900	19901	19902	19903	19904	19905	19906	19909	19910	19911
19912	19913	19914	19915	19916	19919	19920	19921	19922	19923	19924	19925	19926	19929	19930
19931	19932	19933	19934	19935	19938	19939	19940	19941	19942	19943	19946	19947	19948	19949
19950	19951	19952	19955	19956	19957	19959	19960	19961	19962	19963	19964	19965	19968	19969
19970	19971	19972	19973	19976	19977	19978	19979	19980	19981	19982	19983	19985	19986	19987
19989	19997	20009	20010	20014	20015	20016	20017	20018	20019	20020	20021	20022	20023	20026
20027	20028	20029	20030	20031	20032	20033	20035	20036	20037	20038	20039	20040	20043	20044
20045	20046	20047	20048	20049	20050	20052	20053	20054	20055	20056	20057	20058	20059	20061
20062	20063	20064	20065	20066	20069	20079	20097	20098	20100	20101	20103	20104	20106	20107
20108	20109	20110	20111	20114	20115	20117	20118	20120	20121	20123	20124	20125	20126	20127
20128	20129	20130	20131	20133	20134	20135	20136	20138	20139	20140	20141	20142	20143	20144
20146	20147	20148	20149	20151	20152	20153	20154	20155	20157	20158	20159	20161	20162	20163
20165	20166	20167	20170	20171	20173	20174	20176	20177	20178	20179	20180	20181	20182	20183
20184	20185	20186	20187	20188	20189	20190	20191	20194	20202	20219	20220	20223	20224	20227
20228	20231	20232	20233	20234	20235	20236	20239	20240	20243	20244	20247	20248	20251	20252

20253	20254	20255	20256	20257	20258	20259	20262	20263	20264	20265	20267	20268	20271	20272
20275	20276	20277	20279	20281	20282	20283	20284	20285	20286	20287	20288	20289	20290	20291
20295	20296	20297	20298	20299	20300	20301	20302	20303	20304	20305	20308	20309	20311	20312
20313	20314	20315	20317	20318	20320	20321	20323	20324	20326	20327	20328	20329	20330	20331
20332	20333	20335	20336	20337	20338	20339	20342	20343	20346	20347	20348	20349	20350	20354
20355	20356	20357	20358	20359	20362	20363	20364	20365	20366	20367	20370	20380	20409	20410
20411	20412	20413	20415	20416	20418	20419	20421	20422	20423	20424	20425	20428	20439	20484
20485	20488	20489	20492	20493	20496	20497	20500	20501	20502	20503	20504	20505	20508	20509
20510	20511	20512	20513	20514	20515	20516	20517	20518	20519	20522	20523	20525	20526	20527
20528	20530	20531	20532	20533	20534	20535	20537	20538	20539	20540	20543	20544	20546	20547
20548	20549	20550	20551	20552	20553	20554	20555	20556	20557	20558	20559	20560	20563	20564
20566	20567	20568	20569	20570	20571	20572	20573	20574	20575	20576	20579	20580	20581	20582
20584	20585	20586	20587	20590	20591	20592	20593	20595	20596	20597	20598	20600	20601	20603
20604	20606	20607	20608	20609	20612	20688	20689	20690	20692	20918	20919	20920	20921	20941
20942	20943	20944	20945	20946	20948	20983	20987	20988						

ERRORS DETECTED: 0

\*CZKEEB,CZKEEB/CRF/NL:TOC=CZKEEB.SML,CZKEEB.P11  
 RUN-TIME: 60 83 13 SECONDS  
 CORE USED: 34K