

IDENTIFICATION

PRODUCT CODE: MAINDEC-15-D0AC-D (D)
PRODUCT NAME: HARDWARE INDEX REGISTER TEST
DATE REVISED: OCTOBER 1, 1970
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: ROBERT A. WHITTON

COPYRIGHT © 1970
DIGITAL EQUIPMENT
CORPORATION

20

PDP-15
MEMORY
INDEX REGISTER

1. ABSTRACT

THIS PROGRAM TESTS FOR PROPER OPERATION OF THE HARDWARE INDEX REGISTER AND LIMIT REGISTER AND THEIR ASSOCIATED INSTRUCTIONS. THE TESTS START BY VERIFYING THE SIMPLER OPERATIONS AND GRADUALLY BUILD IN COMPLEXITY TO TEST THE MORE COMPLICATED OPERATIONS OF THE REGISTERS. THE FIRST SECTION OF TESTS VERIFY OPERATION OF INDEXING AND THE ASSOCIATED INSTRUCTIONS. THE SECOND SECTION TESTS FOR CORRECT OPERATION OF THE INDEXING FEATURE IN RELATION TO EXTRA FIELDS OF MEMORY, IF AVAILABLE, INDEXED REFERENCES FROM FIELD 0 TO HIGHER FIELDS ARE FIRST TESTED. THEN A TEST PROGRAM IS RELOCATED TO EACH MEMORY FIELD AND CONTROL IS TRANSFERRED TO THAT FIELD TO TEST OPERATION OF INDEXED REFERENCES IN THAT FIELD AND IN THAT FIELD AND IN REFERENCE TO FIELD 0.

2. REQUIREMENTS

EQUIPMENT

PDP-15 EQUIPPED WITH FROM 4K TO 128K WORDS OF CORE.

STORAGE

THIS PROGRAM OCCUPIES CORE LOCATIONS 0-6000 OCTAL INCLUSIVE.

3. LOADING PROCEDURE

-
- A. PLACE TAPE IN READER
 - B. SET BANK MODE TO A 1
 - C. SET ADDRESS SWITCHES TO 17700
 - D. PRESS RESET, THEN READ-IN

4. STARTING PROCEDURE

- A. SELECT CONTROL SWITCH CONFIGURATION DESIRED FROM THE FOLLOWING TABLE. SWITCHES 1,2,3,4,5 MUST BE SET TO INDICATE THE AMOUNT OF CORE AVAILABLE. NORMAL POSITION OF THE SWITCHES ARE ALL ZEROS (DOWN) FOR A 4K MACHINE.

DATA SWITCHES	SET AS	MODE OF OPERATION
0	0	SUPPRESS HALTS
	1	HALT AT END OF CURRENT TEST
1,2,3,4,5		DENOTES ABSOLUTE AMOUNT OF CORE IN SYSTEM, REFER TO TABLE IN SECTION 8.
8	0	NORMAL ITERATIONS
	1	QUICK VERIFY (DO EACH TEST ONCE)
9	0	PRINT ERROR MESSAGE
	1	SUPPRESS ERROR PRINTOUT
10	0	LOOP ON ERROR
	1	PROCEED ON ERROR
11	0	TYPE "END" ON PASS COMPLETION
	1	SUPPRESS "END" MESSAGE

- B. SET BANK MODE TO A "1" AND ADDRESS SWITCHES TO 200
- C. PRESS RESET, THEN PRESS START
- D. PROGRAM WILL EITHER TYPE OUT MESSAGE TO PROCEED IF THE BANK MODE SWITCH IS OPERATIONAL OR AN ERROR WILL BE TYPED OUT.
- E. IN EITHER CASE IN ORDER TO CONTINUE PUT BANK MODE SWITCH TO A "0" AND PRESS RESET, THEN PRESS START. THE BANK MODE SWITCH WILL BE TESTED ONLY ONCE PER PROGRAM LOAD.

5. OPERATING PROCEDURE

PROGRAM AND/OR OPERATION ACTION

THE CONTROL SWITCHES MAY BE CHANGED AT ANY TIME WHILE THE PROGRAM IS RUNNING. DATA SWITCHES 1,2,3,4,5 INDICATING THE AMOUNT OF CORE AVAILABLE ARE ONLY SENSED AT THE BEGINNING OF THE PROGRAM. IF THESE SWITCHES ARE CHANGED WHILE THE PROGRAM IS RUNNING, THEY WILL BE IGNORED UNTIL THE START OF THE NEXT PASS.

6. ERRORS

ERROR HALTS AND DESCRIPTION

A. ERROR TYPEOUT FORMAT FOR NORMAL ERRORS

EXAMPLE:

ER000100 PASS=000001 AC=777000 L=0 XR=700700 LR=000002 BLK=0

THIS IS THE MAIN ERROR TYPEOUT FORMAT. THE LEADING ZEROS ARE NOT SUPPRESSED AND SHOULD BE IGNORED. THE MEANING OF EACH PART OF THIS MESSAGE IS AS FOLLOWS.

ER000100 IS THE TAG IN THE PROGRAM OF THE TEST FAILURE (ER100).

PASS=000001 IS THE NUMBER OF TIMES THROUGH THIS PARTICULAR TEST BEFORE FAILURE.

AC=777000 IS THE QUANTITY IN THE ACCUMULATOR AT THE TIME OF FAILURE.

L=0 IS THE STATE OF THE LINK AT THE TIME OF FAILURE.

XR=700700 IS THE QUANTITY IN THE INDEX REGISTER AT THE TIME OF FAILURE.

LR=000002 IS THE QUANTITY IN THE LIMIT REGISTER AT THE TIME OF FAILURE.

BLK=0 IS THE 32K BLOCK OF CORE CURRENTLY UNDER TEST

THE ERROR TAG IS USED IN CONJUNCTION WITH THE PROGRAM LISTING TO DETERMINE WHICH TEST IS FAILING AND WHAT IS BEING TESTED. THE PASS NUMBER MAY BE HELPFUL IN DETERMINING THE NATURE OF THE FAILURE, WHETHER IT IS A HARD FAILURE (WIRING ERROR OR DEAD CIRCUIT) OR A SOFT FAILURE (WIRING ERROR OR DEAD CIRCUIT) OR A SOFT FAILURE (LOOSE WIRE OR INTERMITTENT CIRCUIT). EACH TEST IS DONE 4096 TIMES; IF THE PASS NUMBER IS GREATER THAN 1 IT INDICATES A PROBABLE SOFT FAILURE.

(6 CONT'D)

B. ERROR TYPEOUT FOR ABNORMAL ERRORS

EXAMPLE:

CAL ADR=003750 TST ADR=000323 AC=707707 XR=037000 BLK=0

THIS ERROR FORMAT IS CAUSED BY EITHER PROGRAM SELF MODIFICATION OR WRONG ADDRESS CALCULATION AND TRANSFER OF CONTROL TO THIS WRONG ADDRESS. THE MEANING OF EACH PART IS AS FOLLOWS.

CAL ADR=003750 IS THE ADDRESS THAT WAS ERRONEOUSLY EXECUTED.

TST ADR=000323 IS THE ADDRESS OF THE CURRENT TEST WHEN CAL OCCURRED.

AC=707707 IS THE QUANTITY IN THE ACCUMULATOR AT THE TIME OF FAILURE.

XR=037000 IS THE QUANTITY IN THE INDEX REGISTER AT TIME OF CAL

BLK=0 IS THE 32K BLOCK OF CORE CURRENTLY UNDER TEST.

ERROR RECOVERY

THERE ARE TWO TYPES OF ERRORS THAT HAVE SPECIAL RECOVERY PROCEDURES. THE FIRST TYPE OF ERROR IS THE CAL. A CAL OR JMS 20 WILL FIRST TYPE OUT THE CAL ADDRESS, TEST ADDRESS AND THE CONTENTS OF THE AC AND THEN HALT. IF CONTINUE IS PRESSED THE PROGRAM WILL START THE FAILING TEST OVER AGAIN. THE SECOND TYPE OF ERROR IS WHEN THE COMPUTER HANGS ATTEMPTING TO REFERENCE A LOCATION IN NON EXISTENT MEMORY. THIS CAN BE VERIFIED BY EXAMINING THE OA WHICH SHOULD NOT CONTAIN AN ADDRESS GREATER THAN THE CORE AVAILABLE. IN THIS CASE WRITE DOWN THE CONTENTS OF THE MAJOR REGISTERS AND RESTART IN LOCATION 21 WHICH WILL CAUSE THE TEST ADDRESS AND CONTENTS OF THE ACCUMULATOR TO TYPE OUT AS IN CAL. PRESSING CONTINUE NOW WILL RESTART THE FAILING TEST.

7. RESTRICTIONS

THE BASIC INSTRUCTION TESTS AND EXTENDED MEMORY TESTS
MUST HAVE RUN SUCCESSFULLY.

8. MISCELLANEOUS

EXECUTION TIME

EXECUTION TIME IS DEPENDENT ON THE AMOUNT OF EXTRA MEMORY
AVAILABLE. APPROXIMATE RUN TIMES FOR ONE PASS IS AS FOLLOWS
FOR A MEMORY CYCLE TIME OF 800NS.

4K-26 SECONDS
8K-29 SECONDS
12K-33 SECONDS
16K-37 SECONDS
20K-41 SECONDS
24K-45 SECONDS
28K-49 SECONDS
32K-53 SECONDS

(8 CONT'D)

DATA SWITCH POSITIONS TO INDICATE CORE AVAILABLE

DATA SWITCHES	SET AS	TOTAL CORE PRESENT IN SYSTEM
1,2,3,4,5	00000	4K
	00001	8K
	00010	12K
	00011	16K
	00100	20K
	00101	24K
	00110	28K
	00111	32K
	01000	36K
	01001	40K
	01010	44K
	01011	48K
	01100	52K
	01101	56K
	01110	60K
	01111	64K
	10000	68K
	10001	72K
	10010	76K
	10011	80K
	10100	84K
	10101	88K
	10110	92K
	10111	96K
	11000	100K
	11001	104K
	11010	108K
	11011	112K
	11100	116K
	11101	120K
	11110	124K
	11111	128K

9.

PROGRAM DESCRIPTION

FOLLOWING IS A VERBAL FLOW CHART OF THE PROGRAM FLOW. THE A STEPS CONSTITUTE THE INITIALIZATION THAT OCCURS WHEN THE PROGRAM IS STARTED, THE B STEPS ARE CONCERNED WITH THE RUNNING OF EACH SEPARATE TEST, THE C STEPS EXPLAIN THE CAL TYPE ERROR HANDLING PROCEDURES AND THE D STEPS EXPLAIN THE NORMAL ERROR HANDLING PROCEDURES.

IF NOTHING IS LISTED IN THE DECISION BOX THE NEXT STEP WILL BE EXECUTED. IF A DECISION POINT IS REACHED EITHER THE NEXT STEP WILL BE EXECUTED OR CONTROL WILL BE TRANSFERRED DEPENDING ON WHERE THE YES OR NO DIRECTS THE PROGRAM.

STEP NO.	SUB NAME	DESCRIPTION OF STEP	YES	NO
A00	SENS	DETERMINE CORE AVAILABLE FROM AC SWITCHES 1-5 AND SET CORE INDICATORS		
A01	ZERFD	ZERO UNUSED CORE IN FIELD 0; INITIALIZE LOOP COUNTER		
A02		TRANSFER CONTROL TO B00		
B00		START OF TEST ROUTINE		
B01	REPEAT	INITIALIZE CAL HANDLER		
B02	REPEAT	GET LOOP COUNTER AND TEST FOR CORRECT TEST ADDRESSING		
B03	REPEAT	IS LOOP COUNTER 0? 0=LEGAL ENTRY		C00
B04	REPEAT	GET AND SAVE ADDRESS OF NEXT TEST		
B05	REPEAT	INITIALIZE LOOP COUNTER; CLEAR ERROR FLAG FOR THIS TEST		
B06	REPEAT	EXECUTE THIS TEST, DIS THIS TEST PASS?		C00
B07	DONE	TRAP BAD ADDRESSING, IS THIS TEST ADDRESS BAD? IS IT GREATER THAN NEXT TEST ADDRESS?	C00	
B08	DONE	IS ERROR FLAG SET?	B06	
B09	DONE	UPDATE LOOP COUNTER, IS COUNTER 0?		B06

(9 CONT'D)

STEP NO.	SUB NAME	DESCRIPTION	YES	NO
B10	DONE	IS AC0 SET (HALT AT END OF CURRENT TEST)?		B12
B11	DONE	HALT WITH TEST ADDRESS IN AC		
B12	DONE	UPDATE TEST ADDRESS AND TRANSFER CONTROL TO NEXT TEST STARTING AT B00.		
C00	TYPICAL	CAL HANDLER; SAVE AC		
C01	TYPICAL	IS AC1 (SUPPRESS ERROR TYPEOUT) SET?		C03
C02	TYPICAL	HALT WITH LAW 777(76077) IN AC		
C03	TYPICAL	PRINT CAL ADDRESS AND CONTENTS OF AC		
C04	TYPICAL	HALT HERE; GO TO C05 WHEN CONTINUE IS PRESSED		
C05	TYPICAL	TRANSFER CONTROL TO B06 TO TRY CURRENT TEST AGAIN		
D00	SETUP	SAVE AC AND LINK, IS BIT 9 (SUPPRESS ERROR TYPEOUT) SET?	D02	
D01	SETUP	PRINT THE FOLLOWING ON TTY: ERROR NUMBER, PASS NUMBER, CONTENTS OF AC, LINK, IR, LR		
D02	SETUP	IS BIT 0 (HALT AT END OF TEST) SET		D04
D03	SETUP	HALT HERE; GO TO D04 WHEN CONTINUE IS PRESSED		
D04	SETUP	IS BIT 10 (LOOP ON ERROR) SET?		
D05	SETUP	CLEAR LOOP COUNTER, TRANSFER CONTROL TO NEXT TEST STARTING AT B00		
D06	SETUP	SET LOOP ON ERROR FLAG; LOOP IN CURRENT TEST BE TRANSFERING CONTROL TO B26		

10. LISTING

```
/
/PDP-15 HARDWARE INDEX REGISTER TEST
/MAINDEC=15-00AC
/COPYRIGHT 1969,1970, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
/   BY ROBERT A. WHITTON
/   REV. OCT. 1, 1970
/
/AC SWITCH OPTIONS
/AC0=1 HALT AT END OF CURRENT TEST OR HALT IN ERROR LOOP
/AC1,2,3,4,5 NUMBER OF ADDITIONAL 4K MEMORY FIELDS AVAILABLE
/AC8=1 QUICK VERIFY (DO EACH TEST ONCE)
/AC9=1 SUPPRESS ERROR TYPEOUT
/AC10=1 PROCEED ON ERROR
/AC11=1 SUPPRESS END MESSAGE TYPEOUT
/INSTRUCTION EQUALITIES
/
707762 EPA=707762 /ASSERT PAGE MODE
740040 HALT=740040 /HALT
707741 SKP15=707741 /SKIP IF PROCESSOR IS A PDP-15
707761 SBA=707761 /SKIP IF IN BANK MODE
734000 CLAC=734000 /CLEAR AC
700406 TLS=700406 /LOAD TELEPRINTER BUFFER AND PRINT
700401 TSF=700401 /SKIP ON TELEPRINTER FLAG
700402 TCF=700402 /CLEAR TELEPRINTER
/
/
```

```

          .TITLE INDX15
          .ABS
          .LOC 200
00200
/
/TEST1=TEST PDP9/15 MODE SWITCH
/START PROGRAM IN PDP=15 MODE
/
00200 204223 T1.0 LAC FTITLE
00201 740200 SZA /HAS TITLE BEEN PRINTED?
00202 600206 JMP ,+4 /YES, SKIP IT
00203 104224 JMS TYPE /PRINT TITLE
00204 005566 MTITLE
00205 444223 ISZ FTITLE
00206 104403 JMS YCORE /DETERMINE HOW MUCH EXTRA MEMORY AVAILABLE
00207 104104 JMS SENS
00210 104342 JMS ZERF0
00211 104000 JMS REPEAT
00212 000250 T2.0
00213 204215 LAC BANKSW
00214 740200 SZA /HAS THIS TEST BEEN DONE?
00215 600224 JMP CYCLE /YES, SKIP IT
00216 707761 SBA
00217 604734 ER1 JMP H1 /ERROR, MACHINE NOT IN BANK MODE
00220 444215 ISZ BANKSW
00221 104224 JMS TYPE
00222 005603 MCONT
00223 740040 HALT
00224 707761 CYCLE SBA /SKIP IF IN BANK MODE
00225 741000 SKP
00226 604736 ER2 JMP H2 /ERROR, MACHINE IS NOT IN PAGE MODE
00227 707764 EBA /SET TO BANK MODE
00230 707761 SBA /SKIP IF IN BANK MODE
00231 604740 ER3 JMP H3 /ERROR, SHOULD BE IN BANK MODE
00232 707764 EBA /SET TO BANK MODE
00233 707761 SBA /SKIP IF IN BANK MODE
00234 604742 ER4 JMP H4 /ERROR, EPA COMPLIMENTS MODE FLOP
00235 707762 EPA /SET TO PAGE MODE
00236 707761 SBA /SKIP IF IN BANK MODE
00237 741000 SKP
00240 604744 ER5 JMP H5 /ERROR, SHOULD BE IN BANK MODE
00241 707762 EPA /SET TO PAGE MODE
00242 707761 SBA /SKIP IF IN BANK MODE
00243 741000 SKP
00244 604746 ER6 JMP H6 /ERROR, EBA COMPLIMENTS MODE FLOP
00245 707741 SKP15 /SKIP IF PROCESSOR IS A PDP=15
00246 604750 ER7 JMP H7 /ERROR, PROCESSOR NOT A PDP=15
00247 104023 JMS DONE
          .EJECT

```

```

/TEST 2=TEST AC,XR,LR DATA PATHS WITH ONES AND ZEROES
/
/TEST 2,0=TEST AC=LR DATA PATH WITH TRANSFER OF ONES
/
00250 104000 T2,0 JMS REPEAT
00251 000264 T2,1
00252 750001 CLC /ALL ONES IN AC
00253 722000 PAL /PLACE AC IN LR
00254 750000 CLA
00255 730000 PLA /PLACE LR IN AC
00256 545630 SAD (777777) /SKIP IF AC NOT ALL ONES
00257 741000 SKP
00260 604752 ER13 JMP H13 /ERROR, BAD (1) DATA TRANSFER
00261 750000 CLA /REINITIALIZE
00262 722000 PAL /PLACE ZEROES IN LR
00263 104023 JMS DONE

/TEST 2,1=TEST AC=LR DATA PATH WITH TRANSFER OF ZEROES
/
00264 104000 T2,1 JMS REPEAT
00265 000276 T2,2
00266 750000 CLA
00267 722000 PAL /PLACE AC IN LR
00270 730000 PLA /PLACE LR IN AC
00271 740200 SZA
00272 604754 ER14 JMP H14 /ERROR, BAD (0) DATA TRANSFER
00273 750001 CLC
00274 722000 PAL /REINITIALIZE LR WITH ONES
00275 104023 JMS DONE

/TEST 2,2=TEST AC=XR DATA PATH WITH TRANSFER OF ONES
/
00276 104000 T2,2 JMS REPEAT
00277 000312 T2,3
00300 750001 CLC /ALL ONES IN AC
00301 721000 PAX /PLACE AC IN XR
00302 750000 CLA
00303 724000 PXA /PLACE AC IN XR
00304 545630 SAD (777777) /SKIP IF AC NOT ALL ONES
00305 741000 SKP
00306 604756 ER15 JMP H15 /ERROR, BAD (1) DATA TRANSFER
00307 750000 CLA
00310 721000 PAX /REINITIALIZE XR WITH ZEROES
00311 104023 JMS DONE

/TEST 2,3=TEST AC=XR DATA PATH WITH TRANSFER OF ZEROES
/
00312 104000 T2,3 JMS REPEAT
00313 000324 T2,4
00314 750000 CLA
00315 721000 PAX /PLACE AC IN XR
00316 724000 PXA /PLACE XR IN AC
00317 740200 SZA
00320 604760 ER20 JMP H20 /ERROR, BAD (0) DATA TRANSFER
00321 750001 CLC
00322 721000 PAX /REINITIALIZE XR WITH ONES
00323 104023 JMS DONE
,EJECT

```

```

/TEST 2,4=TEST LR TO XR DATA PATH WITH TRANSFER OF ONES
/
00324 104000 T2,4 JMS REPEAT
00325 200341 T2,5
00326 750001 CLC /ALL ONES IN AC
00327 722000 PAL /PLACE AC IN LR
00330 731000 PLX /PLACE LR IN XR
00331 750000 CLA
00332 724000 PXA /PLACE XR IN AC
00333 545630 SAD (777777) /SKIP IF AC NOT ALL ONES
00334 741000 SKP
00335 604762 ER21 JMP H21 /ERROR, BAD (1) DATA TRANSFER
00336 750000 CLA
00337 721000 PAX /REINITIALIZE XR WITH ZEROES
00340 104023 JMS DONE
/TEST 2,5=TEST LR TO XR DATA PATH WITH TRANSFER OF ZEROES
/
00341 104000 T2,5 JMS REPEAT
00342 000354 T2,6
00343 750000 CLA
00344 722000 PAL /PLACE AC IN LR
00345 731000 PLX /PLACE LR IN XR
00346 724000 PXA /PLACE XR IN AC
00347 740200 SEA
00350 604764 ER22 JMP H22 /ERROR, BAD (0) DATA TRANSFER
00351 750001 CLC
00352 721000 PAX /REINITIALIZE XR WITH ZEROES
00353 104023 JMS DONE
/TEST 2,6=TEST XR TO LR DATA PATH WITH TRANSFER OF ONES
/
00354 104000 T2,6 JMS REPEAT
00355 000371 T2,7
00356 750001 CLC /ALL ONES IN AC
00357 721000 PAX /PLACE AC IN XR
00360 726000 PXL /PLACE XR IN LR
00361 750000 CLA
00362 730000 PLA /PLACE LR IN AC
00363 545630 SAD (777777) /SKIP IF AC NOT ALL ONES
00364 741000 SKP
00365 604766 ER23 JMP H23 /ERROR, BAD (1) DATA TRANSFER
00366 750000 CLA
00367 722000 PAL
00370 104023 JMS DONE /REINITIALIZE LR WITH ZEROES
.EJECT

```

/TEST 2,7-TEST XR TO LR DATA PATH WITH TRANSFER OF ZEROES

```

/
00371 104002 T2,7 JMS REPEAT
00372 300404 T2,80
00373 750000 CLA
00374 721000 PAX /PLACE AC IN XR
00375 726000 PXL /PLACE XR IN LR
00376 730000 PLA /PLACE LR IN AC
00377 740200 SZA
00400 604770 ER24 JMP H24 /ERROR, BAD (0) DATA TRANSFER
00401 750001 CLC
00402 722000 PAL /REINITIALIZE LR WITH ONES
00403 104023 JMS DONE

```

/TEST 2,80 = TEST THAT AC TO LR DONE NOT AFFECT AC

```

/
00404 104000 T2,80 JMS REPEAT
00405 000415 T2,81
00406 750001 CLC /ALL ONES IN AC
00407 722000 PAL /PLACE AC IN LR
00410 545630 SAD (777777) /CHECK IF AC DIFFERENT
00411 741000 SKP
00412 604772 ER25 JMP H25 /ERROR, AC SHOULD CONTAIN 777777
00413 736000 CLLR
00414 104023 JMS DONE

```

/T2,81 = TEST THAT AC TO XR DOES NOT AFFECT AC

```

/
00415 104000 T2,81 JMS REPEAT
00416 000426 T2,82
00417 750001 CLC /ALL ONES IN AC
00420 721000 PAX /PLACE AC IN XR
00421 545630 SAD (777777) /CHECK IF AC DIFFERENT
00422 741000 SKP
00423 604774 ER30 JMP H30 /ERROR, AC SHOULD CONTAIN 777777
00424 735000 CLX
00425 104023 JMS DONE

```

/T2,82 = TEST THAT XR TO AC DOES NOT AFFECT XR

```

/
00426 104000 T2,82 JMS REPEAT
00427 000442 T2,83
00430 750001 CLC /ALL ONES IN AC
00431 721000 PAX /PLACE AC IN XR
00432 724000 PXA /PLACE XR IN AC
00433 750000 CLA /CLEAR AC AND PLACE THE
00434 724000 PXA /XR IN AC TO SEE IF AFFECTED
00435 545630 SAD (777777)
00436 741000 SKP
00437 604776 ER31 JMP H31 /ERROR, AC + XR SHOULD CONTAIN 777777
00440 735000 CLX
00441 104023 JMS DONE
,EJECT

```

```

/T2.83 = TEST THAT LR TO AC DOES NOT AFFECT LR
/
00442 104000 T2.83 JMS REPEAT
00443 000456 T2.84 CLC /ALL ONES IN AC
00444 750001 CLC /PLACE AC IN LR
00445 722000 PAL /PLACE LR IN AC
00446 730000 PLA /CLEAR AC AND PLACE THE
00447 750000 CLA /LR IN AC TO SEE IF AFFECTED
00450 730000 PLA
00451 545630 SAD (777777)
00452 741000 SKP
00453 605000 ER32 JMP H32 /ERROR, AC + LR SHOULD CONTAIN 777777
00454 736000 CLLR
00455 104023 JMS DONE

/T2.84 = TEST THAT XR TO LR DOES NOT AFFECT THE XR
/
00456 104000 T2.84 JMS REPEAT
00457 000473 T2.85 CLC /ALL ONES IN AC
00460 750001 CLC /PLACE AC IN XR
00461 721000 PAX /PLACE XR IN LR
00462 726000 PXL
00463 750000 CLA
00464 724000 PXA /GET CONTENTS OF XR IN AC
00465 545630 SAD (777777)
00466 741000 SKP
00467 605002 ER33 JMP H33 /ERROR, AC + XR SHOULD CONTAIN 777777
00470 736000 CLLR
00471 735000 CLX
00472 104023 JMS DONE

/T2.85 = TEST THAT LR TO XR DOES NOT AFFECT THE LR
/
00473 104000 T2.85 JMS REPEAT
00474 000510 T3.0 CLC /ALL ONES IN AC
00475 750001 CLC /PLACE AC IN LR
00476 722000 PAL /PLACE LR IN XR
00477 731000 PLX
00500 750000 CLA
00501 730000 PLA /GET CONTENTS OF LR IN AC
00502 545630 SAD (777777)
00503 741000 SKP
00504 605004 ER34 JMP H34 /ERROR, AC + LR SHOULD CONTAIN 777777
00505 736000 CLLR
00506 735000 CLX
00507 104023 JMS DONE
.EJECT

```

```

/
/TEST 3- TEST CLEAR XR,LR,AC
/
/TEST 3,0-TEST CLEAR AC WITH CLAC BUT NOT THE LINK
/
00510 104000 T3,0 JMS REPEAT
00511 000521 T3,1
00512 754003 CLA:CLL:CMH:CMH /SET AC + LINK TO ONES
00513 734000 CLAC /CLEAR AC
00514 740200 SZA
00515 605006 ER35 JMP H35 /ERROR, AC DID NOT CLEAR
00516 740400 SNL
00517 605010 ER40 JMP H40 /ERROR, CLAC CLEARED THE LINK
00520 104023 JMS DONE
/TEST3,1-TEST CLEAR XR
/
00521 104000 T3,1 JMS REPEAT
00522 000532 T3,2
00523 750001 CLC /SET AC TO ALL ONES
00524 721000 PAX /PLACE AC IN XR
00525 735000 CLX /CLEAR XR
00526 724000 PXA /PLACE XR IN AC
00527 740200 SZA
00530 605012 ER41 JMP H41 /ERROR, XR DID NOT CLEAR
00531 104023 JMS DONE
/TEST 3,2-TEST CLEAR LR
/
00532 104000 T3,2 JMS REPEAT
00533 000543 T4,0
00534 750001 CLC /SET AC TO ALL ONES
00535 722000 PAL /PLACE AC IN LR
00536 736000 CLLR /CLEAR LR
00537 730000 PLA /PLACE LR IN AC
00540 740200 SZA
00541 605014 ER42 JMP H42 /ERROR, LR DID NOT CLEAR
00542 104023 JMS DONE
,EJECT

```

```

/
/TEST40 TEST IF AC,XR,LR DATA PATHS AFFECT EACH OTHER
/
/TEST 4.0=TEST IF AC=LR DATA PATH AFFECTS ZEROES IN XR
/
00543 104000 T4.0 JMS REPEAT
00544 000560 T4.1
00545 730000 CLX /CLEAR XR
00546 750000 CLA /TRANSFER ONES + ZEROES BETWEEN AC AND LR
00547 722000 PAL
00550 730000 PLA
00551 750001 CLC
00552 722000 PAL
00553 730000 PLA
00554 724000 PXA
00555 740200 SZA /SKIP IF XR STILL ZERO
00556 605016 ER43 JMP H43 /ERROR, AC=LR TRANSFER AFFECTED XR
00557 104023 JMS DONE
/TEST4.1=TEST IF AC=LR DATA PATH AFFECTS ONES IN XR
/
00560 104000 T4.1 JMS REPEAT
00561 000576 T4.2
00562 750001 CLC /PLACE ONES IN XR
00563 721000 PAX
00564 722000 PAL /TRANSFER ONES + ZEROES BETWEEN AC + LR
00565 730000 PLA
00566 750000 CLA
00567 722000 PAL
00570 730000 PLA
00571 724000 PXA
00572 545630 SAD (777777) /SKIP IF XR HAS CHANGED
00573 741000 SKP /FROM ALL ONES CONDITION
00574 605020 ER44 JMP H44 /ERROR, AC=LR TRANSFER AFFECTED LR
00575 104023 JMS DONE
/TEST 4.2=TEST IF AC=XR DATA PATH AFFECTS ZEROES IN LR
/
00576 104000 T4.2 JMS REPEAT
00577 000613 T4.3
00600 736000 CLLR /CLEAR LR
00601 750000 CLA /TRANSFER ONES + ZEROES BETWEEN AC + XR
00602 721000 PAX
00603 724000 PXA
00604 750001 CLC
00605 721000 PAX
00606 724000 PXA
00607 730000 PLA /PLACE LR IN AC
00610 740200 SZA /SKIP IF LR STILL ZERO
00611 605022 ER45 JMP H45 /ERROR, AC=XR TRANSFER AFFECTED LR
00612 104023 JMS DONE
.EJECT

```

/TEST 4.3=TEST IF AC=XR DATA PATH AFFECTS ONES IN LR

```

/
T4,3   JMS      REPEAT
        T4,4
        CLC
        /PLACE ONES IN LR
        PAL
        /TRANSFER ONES + ZEROS BETWEEN AC + XR
        PAX
        PXA
        CLA
        PAX
        PXA
        PLA
        SAD      (777777) /SKIP IF XR HAS CHANGED
        SKP      /FROM ALL ONES CONDITION
ER50   JMP      H50 /ERROR, AC=XR TRANSFER AFFECTED LR
        JMS      DONE

```

/TEST 4.4=TEST IF LR=XR DATA PATH AFFECTS ZEROS IN AC

```

/
T4,4   JMS      REPEAT
        T4,5
        CLLR
        /CLEAR LR
        CLX
        /CLEAR XR
        CLC
        /PLACE ONES IN LR
        PAL
        /TRANSFER ONES BETWEEN LR=XR
        CLA
        PLX
        CLLR
        PXL
        SEA
        /SKIP IF AC STILL ZERO
ER51   JMP      H51
        JMS      DONE

```

/TEST 4.5=TEST IF LR=XR DATA PATH AFFECTS ONES IN AC

```

/
T4,5   JMS      REPEAT
        T5,0
        CLLR
        /CLEAR LR
        CLX
        /CLEAR XR
        CLC
        /PLACE ONES IN XR
        PAX
        /TRANSFER ONES BETWEEN XR=LR
        PXL
        CLX
        PLX
        SAD      (777777) /SKIP IS AC STILL ALL ONES
        SKP
ER52   JMP      H52 /ERROR, LR=XR DATA PATH AFFECTS AC
        JMS      DONE
        .EJECT

```

```

/
/TEST 5-TEST IF CLEAR AC,LR,XR AFFECT EACH OTHER
/
/TEST 5.0-TEST THAT CLEAR AC WITH CLAC DOES NOT AFFECT LR OR XR
/

```

```

00663 104000 T5.0 JMS REPEAT
00664 000702 T5.1
00665 205631 LAC (525252 /PLACE ALTERNATE ONES + ZEROES IN LR AND XR
00666 722000 PAL
00667 721000 PAX
00670 734000 CLAC
00671 730000 PLA
00672 545631 SAD (525252 /SKIP IF LR HAS CHANGED
00673 741000 SKP
00674 605032 ER53 JMP H53 /ERROR, LR AFFECTED BY CLAC
00675 724000 PXA
00676 545631 SAD (525252 /SKIP IF XR HAS CHANGED
00677 741000 SKP
00700 605034 ER54 JMP H54
00701 104023 JMS DONE

```

```

/TEST 5.1-TEST THAT CLR DOES NOT AFFECT AC OR XR
/

```

```

00702 104000 T5.1 JMS REPEAT
00703 000720 T5.2
00704 205631 LAC (525252 /PLACE ALTERNATES ONES + ZEROES IN XR AND AC
00705 721000 PAX
00706 736000 CLLR
00707 545631 SAD (525252 /SKIP IF AC AFFECTED BY CLR
00710 741000 SKP
00711 605036 ER55 JMP H55 /ERROR, AC CHANGED BY CLR
00712 750000 CLA
00713 724000 PXA
00714 545631 SAD (525252 /SKIP IF XR AFFECTED BY CLR
00715 741000 SKP
00716 605040 ER56 JMP H56 /ERROR, XR CHANGED BY CLR
00717 104023 JMS DONE

```

```

/TEST 5.2-TEST THAT CLX DOES NOT AFFECT AC OR LR
/

```

```

00720 104000 T5.2 JMS REPEAT
00721 000736 T5.3
00722 205631 LAC (525252 /PLACE ALTERNATE ONES + ZEROES IN LR AND AC
00723 722000 PAL
00724 739000 CLX
00725 545631 SAD (525252 /SKIP IF AC AFFECTED BY CLX
00726 741000 SKP
00727 605042 ER60 JMP H60 /ERROR, AC AFFECTED BY CLX
00730 750000 CLA
00731 730000 PLA
00732 545631 SAD (525252 /SKIP IF LR AFFECTED BY CLX
00733 741000 SKP
00734 605044 ER61 JMP H61 /ERROR, LR AFFECTED BY CLX
00735 104023 JMS DONE

```

```

.EJECT

```

```

/TEST 5,3=TEST THAT LR IS NOT PICKING UP BITS
/
00736 104000      T5,3  JMS      REPEAT
00737 000755      T5,4
00740 205632      LAC      (252525      /PICK UP PATTERN
00741 722000      PAL      /PLACE IN LR
00742 730000      PLA      /GET IT BACK
00743 545632      SAD      (252525      /DID IT CHANGE?
00744 741000      SKP      /NO, CONTINUE
00745 605046      ER62   JMP      H62        /YES, ERROR, PICKED UP BIT
00746 205631      LAC      (525252      /GET NEW PATTERN
00747 722000      PAL      /PLACE IN LR
00750 730000      PLA      /GET IT BACK
00751 545631      SAD      (525252      /DID IT CHANGE?
00752 741000      SKP
00753 600745      JMP      ER62        /YES, PICKED UP BITS
00754 104023      JMS      DONE

/TEST 5,4=TEST THAT XR IS NOT PICKING UP BITS
/
00755 104000      T5,4  JMS      REPEAT
00756 000774      T6,0
00757 205632      LAC      (252525      /GET PATTERN
00760 721000      PAX      /PLACE IN XR
00761 724000      PXA      /GET IT BACK
00762 545632      SAD      (252525      /DID IT CHANGE?
00763 741000      SKP      /NO, CONTINUE
00764 605050      ER63   JMP      H63        /YES, ERROR, PICKED UP BIT
00765 205631      LAC      (525252      /GET NEXT PATTERN
00766 721000      PAX      /PLACE IN XR
00767 724000      PXA      /GET IT BACK
00770 545631      SAD      (525252      /DID IT CHANGE?
00771 741000      SKP      /NO, CONTINUE
00772 600764      JMP      ER63        /YES, ERROR, PICKED UP BIT
00773 104023      JMS      DONE
      .EJECT

```

/TEST 6.0=TEST ADD TO AC + XR

/TEST 6.0=TEST ADD MAXIMUM AND MINIMUM POSITIVE QUANTITY TO AC WITH AAC
/TEST THAT ADD TO AC WITH AAC DOES NOT AFFECT LR OR XR

```

00774 104000
00775 001020
00776 735000
00777 736000
01000 750000
01001 723377
01002 545633
01003 741000
01004 605052
01005 723000
01006 545633
01007 741000
01010 605054
01011 724000
01012 740200
01013 605056
01014 730000
01015 740200
01016 605060
01017 104023

```

```

T6.0 JMS REPEAT
      T6.1
      CLX /CLEAR XR, LR, AC
      CLLR
      CLA
      AAC+377 /ADD 377 TO AC
      SAD (377) /SKIP IF AC IS NOT 377
      SKP
ER64 JMP H64 /ERROR, ADD 377 TO AC FAILED
      AAC+0
      SAD (377) /SKIP IF AC IS NOT 377
      SKP
ER65 JMP H65 /ERROR, ADD 0 TO AC FAILED
      PXA /PLACE XR IN AC
      SEA /SKIP IF XR STILL ZERO
ER66 JMP H66 /ERROR, XR PICKED UP BITS
      PLA /PLACE LR IN AC
      SEA /SKIP IF LR STILL ZERO
ER67 JMP H67 /ERROR, LR PICKED UP BIT DURING AAC
      JMS DONE

```

/TEST 6.1=TEST ADD MAXIMUM AND MINIMUM POSITIVE QUANTITY TO XR WITH AXR
/TEST THAT ADD TO XR WITH AXR DOES NOT AFFECT AC OR LR

```

01020 104000
01021 001045
01022 735000
01023 736000
01024 750000
01025 737377
01026 740200
01027 605062
01030 724000
01031 545633
01032 741000
01033 605064
01034 737000
01035 724000
01036 545633
01037 741000
01040 605066
01041 730000
01042 740200
01043 605070
01044 104023

```

```

T6.1 JMS REPEAT
      T6.2
      CLX /CLEAR LR, XR, AC
      CLLR
      CLA
      AXR+377 /ADD 377 TO XR
      SZA /SKIP IF AC STILL ZERO
ER70 JMP H70 /ERROR, AC PICKED UP BITS DURING AXR
      PXA /PLACE XR IN AC
      SAD (377)
      SKP
ER71 JMP H71
      AXR+0
      PXA /PLACE XR IN AC
      SAD (377)
      SKP
ER72 JMP H72 /ERROR, ADD 0 TO AC FAILED
      PLA /PLACE LR IN AC
      SEA /SKIP IF LR STILL ZERO
ER73 JMP H73 /ERROR, LR PICKED UP BITS DURING AXR
      JMS DONE
      .EJECT

```

```

/TEST 6,2 = TEST ADD MAXIMUM AND MINIMUM NEGATIVE QUANTITY TO AC WITH AAC
/
01045 104000 T6,2 JMS REPEAT
01046 001061 T6,3
01047 754002 CLA:CLL
01050 723777 AAC+777 /ADD 777777 TO AC (-1)
01051 545630 SAD (777777) /SKIP IF AC IS NOT 777777
01052 741000 SKP
01053 605072 ER74 JMP H74 /ERROR, ADD -1 TO AC FAILED
01054 723400 AAC+400 /ADD 777400 TO AC (-256 DECIMAL)
01055 545634 SAD (777377)
01056 741000 SKP
01057 605074 ER75 JMP H75 /ERROR, ADD -256 TO AC FAILED
01060 104023 JMS DONE

/TEST 6,3 = TEST ADD MAXIMUM AND MINIMUM NEGATIVE QUANTITY TO XR WITH AXR
/
01061 104000 T6,3 JMS REPEAT
01062 001077 T6,4
01063 735000 CLX
01064 737777 AXR+777
01065 724000 PXA /PLACE XR IN AC
01066 545630 SAD (777777) /SKIP IF AC IS NOT 777777
01067 741000 SKP
01070 605076 ER76 JMP H76 /ERROR, ADD -1 TO XR FAILED
01071 737400 AXR+400
01072 724000 PXA
01073 545634 SAD (777377)
01074 741000 SKP
01075 605100 ER77 JMP H77 /ERROR, ADD -256 TO XR FAILED
01076 104023 JMS DONE

/TEST 6,4 = TEST UP COUNT FOR INDEX REGISTER WITH AXR+1
/
01077 104320 T6,4 JMS INIT10 /INITIALIZE LOOP COUNTER FOR 1 LOOPS
01100 104000 JMS REPEAT
01101 001122 T6,5
01102 735000 CLX /CLEAR INDEX REGISTER
01103 141120 DEM K6,4 /CLEAR MEMORY CELL UPCOUNT
01104 737001 L6,4 AXR+1 /ADD 1 TO XR
01105 441120 ISZ K6,4 /ADD 1 TO MEMORY CELL UPCOUNT
01106 741000 SKP
01107 601114 JMP ER100+1 /MEMORY CELL UPCOUNT OVERFLOWED
01110 724000 PXA /GET CONTENTS OF XR
01111 541120 SAD K6,4 /IS THE SAME AS MEMORY CELL?
01112 601104 L6,4 /YES, CONTINUE TO LOOP
01113 605102 ER100 JMP H100 /ERROR, XR FAILED TO INCREMENT
01114 724000 PXA /GET CONTENTS OF XR
01115 740200 SZA /DID IT OVERFLOW?
01116 601113 JMP ,=3 /NO, ERROR
01117 741000 SKP /YES, EXIT
01120 000000 K6,4 0 /MEMORY UPCOUNT CELL
01121 104023 JMS DONE
,EJECT

```

/TEST 6,5 = TEST UPCOUNT FOR AC WITH AAC+1

```

/
01122 104000 T6,5 JMS REPEAT
01123 001142 T6,6
01124 754000 CLAICLL /CLEAR AC AND LINK
01125 141140 DEM K6,5 /CLEAR MEMORY CELL UPCOUNT
01126 723001 L6,5 AAC+1 /ADD 1 TO AC
01127 441140 ISZ K6,5 /ADD 1 TO MEMORY CELL UPCOUNT
01130 741000 SKP
01131 601135 JMP ER101+1 /MEMORY CELL UPCOUNT OVERFLOWED
01132 541140 SAD K6,5 /IS AC SAME AS MEMORY CELL UPCOUNT?
01133 601126 JMP L6,5 /YES, CONTINUE TO LOOP
01134 605104 ER101 JMP H101 /ERROR, AC FAILED TO INCREMENT
01135 740200 SEA /DID AC OVERFLOW?
01136 601134 JMP ,=2 /NO, ERROR
01137 741000 SKP /YES, EXIT
01140 000000 K6,5 0 /MEMORY CELL UPCOUNT
01141 104023 JMS DONE

```

/TEST 6,6 = TEST DOWN COUNT FOR INDEX REGISTER WITH AXR+777

```

/
01142 104000 T6,6 JMS REPEAT
01143 001167 T6,7
01144 750001 CLAICMA
01145 721000 PAX /PUT 777777 INTO XR AND
01146 041165 DAC K6,6 /MEMORY DOWN COUNT CELL
01147 737777 L6,6 AXR+777 /ADD -1 TO XR
01150 201165 LAC K6,6
01151 345630 TAD (=1 /ADD -1 TO MEMORY CELL DOWN COUNT
01152 041165 DAC K6,6
01153 741200 SNA /DID CELL REACH 0?
01154 601161 JMP ER102+1 /YES, TEST 0 AND EXIT
01155 724000 PXA /GET CONTENTS OF INDEX REGISTER
01156 541165 SAD K6,6 /IS IT SAME AS MEMORY CELL COUNT?
01157 601147 JMP L6,6 /YES, CONTINUE TO LOOP
01160 605106 ER102 JMP H102 /ERROR, XR FAILED TO DECREMENT
01161 724000 PXA /GET CONTENTS OF XR
01162 740200 SEA /IS IT 0?
01163 601160 JMP ,=3 /NO, EXIT, DID NOT DECREMENT TO 0
01164 741000 SKP
01165 000000 K6,6 0 /MEMORY DOWN COUNT CELL
01166 104023 JMS DONE
.EJECT

```

/TEST 6,7 = TEST DOWN COUNT FOR AC WITH AAC+777

```

/
01167 104000 T6,7 JMS REPEAT
01170 001215 T6,8
01171 750001 CLAICMA
01172 041212 DAC K6,7 /PUT 777777 IN AC AND DOWN COUNT CELL
01173 723777 L6,7 AAC+777 /ADD -1 TO AC
01174 041213 DAC C6,7 /TEMPORARLY SAVE AC
01175 201212 LAC K6,7 /GET DOWNCOUNT MEMORY CELL
01176 345630 TAD (-1 /ADD -1
01177 041212 DAC K6,7 /PUT BACK INTO MEMORY CELL
01200 201213 LAC C6,7 /GET SAVED AC
01201 741200 SNA /IS IT 0 YET?
01202 601206 JMP ER103+1 /YES, TEST DOWNCOUNT CELL
01203 541212 SAD K6,7 /NO, IS IT THE SAME AS DOWNCOUNT CELL
01204 601173 JMP L6,7 /YES, CONTINUE TO LOOP
01205 605110 ER103 JMP H103 /AC FAILED TO DECREMENT
01206 201212 LAC K6,7 /IS TEST CELL 0
01207 740200 SEA
01210 601205 JMP .+3 /NO, ERROR
01211 601214 JMP .+3 /YES EXIT
01212 000000 K6,7 0 /MEMORY DOWNCOUNT CELL
01213 000000 C6,7 0 /TEMPORARY STORAGE FOR AC
01214 104023 JMS DONE

```

/TEST 6,8 = TEST OF INDEX REGISTER INSTRUCTION AXR+RANDOM NUMBER
/WITH THE RANDOM NUMBER BEING IN THE RANGE OF 0-777

```

/
01215 104313 T6,8 JMS INITN /RESTORE LOOP COUNTER TO NORMAL
01216 104000 JMS REPEAT
01217 001247 T7,0
01220 724000 PXA
01221 041244 DAC K6,8 /SAVE XR IN SUM MEMORY CELL
01222 104325 JMS RANDOM
01223 505635 AND (777 /RESTRICT SIZE OF RANDOM NUMBER
01224 041245 DAC V6,8 /SAVE IN TEMPORARY AREA
01225 345636 TAD (AXR /FORM AXR+RANDOM NUMBER
01226 041237 DAC X6,8 /PLACE IN EXECUTABLE LOCATION
01227 201245 LAC V6,8
01230 505637 AND (400 /TEST IF RANDOM NUMBER NEGATIVE
01231 741200 SNA /IT IS NEGATIVE, FORM 18 BIT WORD
01232 601234 JMP .+2 /NOT NEGATIVE CONTINUE
01233 205640 LAC (777000
01234 341245 TAD V6,8
01235 341244 TAD K6,8 /UPDATE SUM MEMORY CELL
01236 041244 DAC K6,8
01237 740040 X6,8 XX /MODIFIED TO AXR+RANDOM NUMBER
01240 724000 PXA /GET XR
01241 541244 SAD K6,8 /IS IT THE SAME AS SUM MEMORY CELL
01242 601246 JMP .+4 /YES, CONTINUE
01243 605112 ER104 JMP H104
01244 000000 K6,8 0 /SUM MEMORY CELL
01245 000002 V6,8 0 /TEMPORARY STORAGE AREA
01246 104023 JMS DONE
.EJECT

```

24

```

/TEST 7 = TEST AXS
/TEST 7.0 = TEST AXS WITH EQUAL XR AND LR
/
01247 104000 T7,0 JMS REPEAT
01250 001265 T7,1
01251 736000 CLLR /CLEAR LIMIT REGISTER
01252 754001 CLA:CLLICMA
01253 721000 PAX /PLACE ALL ONES IN INDEX REGISTER
01254 750000 CLA
01255 725001 AXS+1
01256 605114 ER110 JMP H110 /ERROR, DID NOT SKIP WHEN LR*XR=0
01257 750001 CLC /PLACE ALL ONES IN LR AND XR
01260 722000 PAL
01261 721000 PAX
01262 725000 AXS+0 /ADD 0 TO XR AND SKIP IF >=LR
01263 605116 ER111 JMP H111 /ERROR, DID NOT SKIP WHEN LR*XR=777777
01264 104023 JMS DONE

/TEST 7.1 = TEST AXS FOR SKIP CONDITION WITH LR=0 AND XR
/HAS A ONE FLOATED THROUGH IT
/
01265 104000 T7,1 JMS REPEAT
01266 001304 T7,2
01267 736000 CLLR /CLEAR LR
01270 754030 CLA:CLLIAC /PUT A ONE IN AC
01271 721000 PAX /PLACE IT IN XR
01272 725000 AXS+0 /ADD 0 TO XR, SKIP IF IT IS = > LR
01273 605120 ER112 JMP H112
01274 740010 RAL
01275 740100 SMA /SKIP ON ACC=1, TEST NOT COMPLETE
01276 601271 JMP ER112=2 /LOOP THROUGH TEST WITH ONE IN NEXT POSITION
01277 721000 PAX /ACC=1, STORE 400000 IN XR
01300 725000 AXS+0 /ADD 0 TO XR AND SKIP IF=> LR
01301 741000 SKP
01302 605122 ER113 JMP H113 /ERROR, SKIPPED WHEN LR=0 AND XR=NEGATIVE
01303 104023 JMS DONE

/TEST 7.2 = TEST AXS FOR NON-SKIP CONDITION WHEN XR=0 AND
/LR HAS A ONE FLOATED THROUGH IT
/
01304 104000 T7,2 JMS REPEAT
01305 001324 T7,3
01306 735000 CLX /CLEAR XR
01307 754030 CLLICLA:IAC /CLEAR AC, LINK AND PUT ONE IN AC
01310 722000 PAL /PLACE AC IN LR
01311 725000 AXS+0 /ADD 0 TO XR, SKIP IF = > LR
01312 741000 SKP
01313 605124 ER114 JMP H114 /ERROR, SKIPPED WHEN LR = > XR
01314 740010 RAL /ROTATE BIT AND TEST IF IT IS ACC
01315 740100 SMA
01316 601310 JMP ER114=3 /NO, LOOP THROUGH TEST AGAIN
01317 740030 IAC
01320 722000 PAL /ACC=1, STORE 400001 IN LR
01321 725002 AXS+0
01322 605126 ER115 JMP H115 /ERROR, DID NOT SKIP WHEN XR=0 AND LR=MINUS
01323 104023 JMS DONE
.EJECT

```

/TEST 7,3 = TEST AXS FOR NON-SKIP WITH RIPPLE CARRY IN XR

```

/
01324 104000  T7,3  JMS      REPEAT
01325 001334      T8,0
01326 736000      CLLR
01327 754001      CLAI:CLLICMA  /CLEAR LIMIT REGISTER
01330 721000      PAX          /ALL ONES IN AC
01331 725001      AXS+1      /ADD 1 TO SET XR TO 0
01332 605130  ER116  JMP      H110    /ERROR, SKIPPED WHEN XR=0, LR=0
01333 104023      JMS      DONE

```

/TEST 8 = TEST COMBINATIONS OF INDEX REGISTER INSTRUCTIONS

/TEST 8,0 = FIRST COMBINATION INSTRUCTION EXECUTION TEST USING SHIFTS

```

/
01334 104000  T8,0  JMS      REPEAT
01335 001370      T8,1
01336 205631      LAC      (525252)
01337 722000      PAL
01340 205632      LAC      (252525)
01341 721000      PAX          /SHIFT DATA AROUND PATHS
01342 730000      PLA
01343 726000      PXL
01344 721000      PAX
01345 724000      PXA
01346 731000      PLX
01347 722000      PAL          /SHIFT DATA AND CLEAR REGISTERS
01350 730000      PLA
01351 736000      CLLR
01352 722000      PAL
01353 724000      PXA
01354 735000      CLX
01355 721000      PAX
01356 734000      CLAC
01357 730000      PLA
01360 545631      SAD      (525252)
01361 741000      SKP
01362 605132  ER117  JMP      H117    /ERROR, BAD DATA
01363 724000      PXA
01364 545632      SAD      (252525)
01365 741000      SKP
01366 605134  ER120  JMP      H120    /ERROR, BAD DATA
01367 104023      JMS      DONE
                .EJECT

```

/TEST 8.1 = SECOND COMBINATION INSTRUCTION TEST USING ADD, SHIFTS
/AND MACHINE MODE CHANGES ALWAYS EXECUTING INSTRUCTIONS IN PAGE MODE

01370	104000	T8.1	JMS	REPEAT	
01371	001422		T8.2		
01372	750001		CLC		/PLACE ALL ONES IN XR+LR
01373	722000		PAL		
01374	721000		PAX		
01375	707764		EBA		/BANK MODE TO PAGE MODE
01376	707762		EPA		
01377	735000		CLX		
01400	707764		EBA		/BANK MODE TO PAGE MODE
01401	707762		EPA		
01402	736000		CLLR		/CLEAR LR
01403	730000		PLA		
01404	707764		EBA		
01405	707762		EPA		
01406	737100		AXR+100		/ADD 100 TO XR, SHOULD NOW CONTAIN 100
01407	723100		AAC+100		/ADD 100 TO AC, SHOULD NOW CONTAIN 100
01410	722000		PAL		
01411	730000		PLA		
01412	545641		SAD	(100)	
01413	741000		SKP		
01414	605136	ER121	JMP	H121	
01415	724000		PXA		
01416	545641		SAD	(100)	
01417	741000		SKP		
01420	605140	ER122	JMP	H122	
01421	104023		JMS	DONE	

/TEST 8.2 = THIRD COMBINATION INSTRUCTION TEST USING ADDS, SKIPS,
/MACHINE MODE CHANGES AND SHIFTS

01422	104000	T8.2	JMS	REPEAT
01423	001444		T8.21	
01424	735000		CLX	
01425	205633		LAC	(377)
01426	722000		PAL	
01427	734000		CLAC	
01430	707764		EBA	
01431	707762		EPA	
01432	734000		CLAC	
01433	725377		AXS+377	
01434	605142	ER124	JMP	H124
01435	707764		EBA	
01436	707762		EPA	
01437	730000		PLA	
01440	545633		SAD	(377)
01441	741000		SKP	
01442	605144	ER125	JMP	H125
01443	104023		JMS	DONE
			.EJECT	

/TEST 8,21=FOURTH COMBINATION INSTRUCTION TEST USING CHAIN
/OF XCT,X

01444	104000	T8,21	JMS	REPEAT
01445	001477		T8,3	
01446	750030		CLAI IAC	
01447	721000		PAX	
01450	750000		CLA	
01451	411454		XCT	T8A,X
01452	723001		AAC+1	
01453	601455		JMP	,+2
01454	000000	T8A	0	
01455	411460		XCT	T8B,X
01456	723001		AAC+1	
01457	601461		JMP	,+2
01460	000000	T8B	0	
01461	411464		XCT	T8C,X
01462	723001		AAC+1	
01463	601465		JMP	,+2
01464	000000	T8C	0	
01465	411470		XCT	T8D,X
01466	723001		AAC+1	
01467	601471		JMP	,+2
01470	000000	T8D	0	
01471	723001		AAC+1	
01472	723001		AAC+1	
01473	545642		SAD	(12
01474	741000		SKP	
01475	605146	ER126	JMP	H120
01476	104023		JMS	DONE
			,EJECT	

/TEST 8,3 = TEST INTER-REGISTER TRANSFER OF RANDOM NUMBERS
/BETWEEN AC AND LR

```

/
01477 104000 T8,3 JMS REPEAT
01500 301512 T8,4
01501 104325 JMS RANDOM /GET RANDOM NUMBER
01502 722000 PAL /PUT IN LR
01503 730000 PLA /GET IT BACK
01504 544341 SAD RANB /DID IT CHANGE?
01505 741000 SKP /NO, CONTINUE
01506 605150 ER130 JMP H130 /YES, ERROR
01507 104023 JMS DONE

```

/TEST 8,4 = TEST INTER-REGISTER TRANSFER OF RANDOM NUMBERS
/BETWEEN AC AND XR

```

/
01510 104000 T8,4 JMS REPEAT
01511 001521 T8,5
01512 104325 JMS RANDOM /GET RANDOM NUMBER
01513 721000 PAX /PUT IN XR
01514 724000 PXA /GET IT BACK
01515 544341 SAD RANB /DID IT CHANGE?
01516 741000 SKP /NO, CONTINUE
01517 605152 ER131 JMP H131 /YES, ERROR
01520 104023 JMS DONE

```

/TEST 8,5 = TEST INTER-REGISTER TRANSFER OF RANDOM NUMBERS
/BETWEEN AC TO LR TO XR TO AC

```

/
01521 104000 T8,5 JMS REPEAT
01522 001533 T8,6
01523 104325 JMS RANDOM /GET RANDOM NUMBER
01524 721000 PAX /MOVE AROUND
01525 726000 PXL
01526 730000 PLA
01527 544341 SAD RANB /DID IT CHANGE?
01530 741000 SKP /NO, CONTINUE
01531 605154 ER132 JMP H132 /YES, ERROR
01532 104023 JMS DONE

```

/TEST 8,6 = TEST INTER-REGISTER TRANSFER OF RANDOM NUMBERS
/BEWTWEEN AC TO XR TO LR TO AC

```

/
01533 104000 T8,6 JMS REPEAT
01534 001545 T9,0
01535 104325 JMS RANDOM /GET RANDOM NUMBER
01536 722000 PAL /MOVE AROUND
01537 731000 PLX
01540 724000 PXA
01541 544341 SAD RANB /DID IT CHANGE?
01542 741000 SKP /NO, CONTINUE
01543 605156 ER133 JMP H133 /YES, ERROR
01544 104023 JMS DONE
,EJECT

```

```

/
/TEST9=TEST DIRECT INDEXED ADDRESSING
/
/TEST9,0=TEST LAC AND DAC INDEXED POSITIVELY
/

```

01545	104000	T9,0	JMS	REPEAT	
01546	001564		T9,01		
01547	143722		DEM	PUT+2	
01550	205643		LAC	(2	/PLACE 2 IN INDEX REGISTER
01551	721000		PAX		
01552	213715		LAC	GET,X	/GET CONTENTS OF GET+2
01553	543717		SAD	GET+2	/SKIP IF NOT CONTENTS OF GET+2
01554	741000		SKP		/GOOD, CONTINUE
01555	605160	ER134	JMP	H134	/ERROR, INDEX BIT 5 FAILED WITH LAC
01556	053720		DAC	PUT,X	/DEPOSIT IN PUT+2
01557	203722		LAC	PUT+2	/GET CONTENTS OF PUT+2
01560	543717		SAD	GET+2	/SKIP IF NOT CONTENTS OF GET+2
01561	741000		SKP		
01562	605162	ER135	JMP	H135	/ERROR, INDEX BIT 5 FAILED WITH DAC
01563	104023		JMS	DONE	
					/TEST 9,01=TEST LAC AND DAC INDEXED NEGATIVELY
					/
01564	104000	T9,01	JMS	REPEAT	
01565	001603		T9,1		
01566	143722		DEM	PUT+2	
01567	205644		LAC	(=2	/PUT =2 IN INDEX REGISTER
01570	721000		PAX		
01571	213721		LAC	GET+4,X	/GET CONTENTS OF GET+2
01572	543717		SAD	GET+2	/SKIP IF NOT CONTENTS OF GET+2
01573	741000		SKP		
01574	605164	ER136	JMP	H136	/ERROR, WRONG EFFECTIVE ADDRESS CALCULATED
01575	053724		DAC	PUT+4,X	/DEPOSIT IN PUT+2
01576	203722		LAC	PUT+2	
01577	543717		SAD	GET+2	/SKIP IF NOT CONTENTS OF GET+2
01600	741000		SKP		
01601	605166	ER137	JMP	H137	/ERROR, WRONG EFFECTIVE ADDRESS CALCULATED
01602	104023		JMS	DONE	
					.EJECT

/TEST9,1-TEST XOR AND DZM INDEXED POSITIVELY

```

/
01603 104000 T9,1 JMS REPEAT
01604 001625 T9,11
01605 205645 LAC (1 /PLACE 1 IN XR
01606 721000 PAX
01607 750001 CLC /PLACE ALL ONES IN PUT+1
01610 043721 DAC PUT+1
01611 153720 DZM PUT,X /DEPOSIT ZERO IN PUT+1
01612 203721 LAC PUT+1 /GET CONTENTS OF PUT+1
01613 740200 SEA /SKIP IF IT GOT ZEROED
01614 605170 ER140 JMP H140 /ERROR, INDEX BIT 5 FAILED WITH DZM
01615 205631 LAC (525252
01616 251622 XOR K9,1,X /XOR WITH CONTENTS OF K9,1+1
01617 545630 SAD (777777
01620 741000 SKP
01621 605172 ER141 JMP H141 /ERROR, INDEX BIT 5 FAILED WITH XOR
01622 741000 K9,1 SKP
01623 252525 JMS
01624 104023 JMS DONE

```

/TEST 9,11-TEST XOR AND DZM INDEXED NEGATIVELY

```

/
01625 104000 T9,11 JMS REPEAT
01626 001647 T9,2
01627 205630 LAC (=1 /PLACE -1 IN INDEX REGISTER
01630 721000 PAX
01631 750001 CLC /PLACE ALL ONE IN PUT+1
01632 043721 DAC PUT+1
01633 153722 DZM PUT+2,X /DEPOSIT ZERO IN PUT+1
01634 203721 LAC PUT+1
01635 740200 SEA /SKIP IF PUT+1 CLEARED
01636 605174 ER142 JMP H142 /ERROR, WRONG EFFECTIVE ADDRESS CALCULATED
01637 205631 LAC (525252
01640 251646 XOR K9,11+2,X /XOR WITH CONTENTS OF K9,11+1
01641 545630 SAD (777777
01642 741000 SKP
01643 605176 ER143 JMP H143 /ERROR, WRONG EFFECTIVE ADDRESS CALCULATED
01644 741000 K9,11 SKP
01645 252525 JMS
01646 104023 JMS DONE

```

/TEST9,2-TEST JMS AND JMP INDEXED

```

/
01647 104000 T9,2 JMS REPEAT
01650 001662 T9,3
01651 205643 LAC (2 /PLACE 2 IN XR
01652 721000 PAX
01653 113725 JMS SUB1,X /JMS TO SUB1+2
01654 605200 ER144 JMP H144 /ERROR, INDEX BIT 5 FAILED WITH JMS
01655 205645 LAC (1 /PLACE 1 IN XR
01656 721000 PAX
01657 613732 JMP SUB2,X /JMP SUB2+1
01660 605202 ER145 JMP H145 /ERROR, INDEX BIT 5 FAILED WITH JMS
01661 104023 JMS DONE

```

.EJECT

/TEST9,3=TEST SAD AND XCT INDEXED

```

/
T9,3   JMS      REPEAT
01662  104000
01663  001700   T9,4
01664  736000   CLLR           /CLEAR LIMIT REGISTER
01665  205643   LAC      (2    /PLACE 2 IN XR
01666  721000   PAX
01667  203717   LAC      GET+2 /GET CONTENTS OF GET+2
01670  553715   SAD      GET,X
01671  741000   SKP
01672  605204   ER146  JMP      H146 /ERROR, INDEX BIT 5 FAILED WITH SAD
01673  205645   LAC      (1    /PLACE 1 IN XR
01674  721000   PAX
01675  413734   XCT      SUB3,X
01676  605206   ER147  JMP      H147 /ERROR, INDEX BIT 5 FAILED WITH XCT
01677  104023   JMS      DONE

```

/TEST9,4=TEST REMAINDER OF MEMORY REFERENCE INSTRUCTIONS

/INDEXED ADD,TAD,ISE,AND

```

/
T9,4   JMS      REPEAT
01700  104000
01701  001732   T9,5
01702  143720   DEM      PUT    /ZERO AREA FOR ISE TEST
01703  143722   DEM      PUT+2
01704  205645   LAC      (1    /PLACE 1 IN XR
01705  721000   PAX
01706  750000   CLA           /CLEAR AC AND ADD TOTE+1
01707  313736   ADD      TOTE,X
01710  543737   SAD      TOTE+1
01711  741000   SKP
01712  605210   ER150  JMP      H150 /ERROR, ADD INDEXED FAILED
01713  737001   AXR+1
01714  353736   TAD      TOTE,X
01715  545646   SAD      (770
01716  741000   SKP
01717  605212   ER151  JMP      H151 /ERROR, TAD INDEXED FAILED
01720  453720   ISE      PUT,X  /ISE PUT+2
01721  203722   LAC      PUT+2
01722  545645   SAD      (1
01723  741000   SKP
01724  605214   ER152  JMP      H152 /ERROR, ISE INDEXED FAILED
01725  513720   AND      PUT,X  /AND AC WITH PUT+2
01726  545645   SAD      (1
01727  741000   SKP
01730  605216   ER153  JMP      H153 /ERRO, AND INDEXED FAILED
01731  104023   JMS      DONE
          .EJECT

```

/TEST9,5=TEST THAT CAL AND CAL* INDEXED DO NOT INDEX

```

01732 104000
01733 001756
01734 205647
01735 040021
01736 205650
01737 040022
01740 205651
01741 040023
01742 205645
01743 721000
01744 000000
01745 741000
01746 605220
01747 205652
01750 721000
01751 205653
01752 040020
01753 030000
01754 605222
01755 104023

```

```

/
T9,5 JMS REPEAT
      T10,0
      LAC (JMP* 20 /INITIALIZE RETURN
      DAC 21
      LAC (ISE 21
      DAC 22
      LAC (JMP* 21
      DAC 23
      LAC (1
      PAX
      CAL 0,X /TRY TO CAL INDEXED
      SKP
ER154 JMP H154 /ERROR, CAL INDEXED WORKED
      LAC (3 /PLACE 3 IN XR
      PAX
      LAC (SUB4
      DAC 20
      CAL* 0,X
ER155 JMP H155 /ERROR, CAL* INDEXED WORKED
      JMS DONE

```

/TEST10=TEST INDIRECT INDEXED ADDRESSING

/TEST10,0=TEST LAC* AND DAC* INDEXED

```

01756 104000
01757 002002
01760 143747
01761 203715
01762 740031
01763 343746
01764 721000
01765 750000
01766 233715
01767 543746
01770 741000
01771 605224
01772 737001
01773 203715
01774 073715
01775 203747
01776 543715
01777 741000
02000 605226
02001 104023

```

```

/
T10,0 JMS REPEAT
      T10,1
      DEM IGET*1
      LAC GET
      CMA: IAC
      TAD IGET /CALCULATE AND SAVE ADDRESS DIFFERENCE
      PAX /BETWEEN GET AND IGET
      CLA
      LAC* GET,X /GET CONTENTS OF IGET
      SAD IGET
      SKP
ER160 JMP H160 /ERROR, LAC* INDEXED FAILED
      AXR*1
      LAC GET
      DAC* GET,X /DEPOSIT AT IGET*1
      LAC IGET*1 /GET RESULTS OF GET OPERATION
      SAD GET
      SKP
ER161 JMP H161
      JMS DONE
      .EJECT

```

/TEST10,1=TEST JMP* AND JMS* INDEXED

```

/
02002 104000 T10,1 JMS REPEAT
02003 002017 T10,2 JMS REPEAT
02004 205654 LAC (JMP ER162 /INITIALIZE ERROR RETURN
02005 043721 DAC PUT+1
02006 205655 LAC (5 /PLACE 5 IN XR
02007 721000 PAX
02010 133750 JMS* JGET,X /JMS TO SUB1+2
02011 605230 ER162 JMP H162 /ERROR, JMS* INDEXED FAILED
02012 205645 LAC (1
02013 721000 PAX
02014 633751 JMP* JSUB2,X
02015 605232 ER163 JMP H163 /ERROR, JMP* INDEXED FAILED
02016 104023 JMS DONE

```

/TEST10,2=TEST SAD* AND XCT* INDEXED

```

/
02017 104000 T10,2 JMS REPEAT
02020 002034 T10,3 JMS REPEAT
02021 205643 LAC (2 /PLACE 2 IN XR
02022 721000 PAX
02023 203717 LAC GET+2 /PUT QUANTITY GET+2 IN AC
02024 573754 SAD* SGET,X /CHECK IF SAD INDIRECT INDEXED OK
02025 741000 SKP
02026 605234 ER164 JMP H164 /ERROR, SAD* INDEXED FAILED
02027 433755 XCT* SGETX,X
02030 543715 SAD GET
02031 741000 SKP
02032 605236 ER165 JMP H165 /ERROR, XCT* INDEXED FAILED
02033 104023 JMS DONE
      .EJECT

```

/

/TEST 10,3 TEST LAC*A,X WHERE A IS AN INCREASING POSITIVE VALUE

/TO MAX OF 7777 AN X IS AN INCREASING NEGATIVE VALUE SO THAT

/AN EFFECTIVE ADDRESS OF 0 IS ALWAYS CALCULATED

/

02034	104320	T10,3	JMS	INIT10	
02035	104000		JMS	REPEAT	
02036	002061		T11,0		
02037	142057		DEM	V10,3	/CLEAR VARIABLE STORAGE+XR
02040	735000		CLX		
02041	205631		LAC	(525252	/SET CONSTANT IN LOC 0
02042	040000		DAC	0	
02043	232057	L10,3	LAC*	V10,3,X	/ADDRESS LOC 0
02044	545631		SAD	(525252	
02045	741000		SKP		
02046	605240	ER166	JMP	H166	/LOC 0 NOT ADDRESSED; IF FAILURE
02047	202057		LAC	V10,3	/PLACE JMP ,=3 IN ER166 TO SEE
02050	740030		IAC		/LOOP AND INSURE LOOP IS NOT INCREMENTED
02051	545656		SAD	(100000	/END, YET
02052	602060		JMP	V10,3+1	/YES, EXIT
02053	042057		DAC	V10,3	/NO, SAVE
02054	740031		CHAI IAC		
02055	721000		PAX		/PUT COMPLIMENT IN XR
02056	602043		JMP	L10,3	
02057	000000	V10,3	0		
02060	104023		JMS	DONE	

/

/TEST11=TEST INDIRECT INDEX AUTO INCREMENTED ADDRESSING

/

/TEST11,0=TEST LAC* 10,DAC* 10

/

02061	104313	T11,0	JMS	INITN	/RESTORE NORMAL LOOP COUNT
02062	104000		JMS	REPEAT	
02063	002105		T11,1		
02064	203760		LAC	DATA	/INITIALIZE 10
02065	040010		DAC	10	
02066	205645		LAC	(1	/PUT 1 IN XR
02067	721000		PAX		
02070	230010		LAC*	10,X	/LOAD GET+2 INTO AC
02071	543762		SAD	DATA+2	
02072	741000		SKP		
02073	605242	ER170	JMP	H170	/ERROR,
02074	205657		LAC	(STORE	
02075	040010		DAC	10	
02076	205631		LAC	(525252	
02077	070010		DAC*	10,X	/DEPOSIT IN PUT+2
02100	203765		LAC	STORE+2	
02101	545631		SAD	(525252	
02102	741000		SKP		
02103	605244	ER171	JMP	H171	/ERROR
02104	104023		JMS	DONE	

,EJECT

```

/TEST11,1=TEST SAD* 11,XCT* 11
/
02105 104000 T11,1 JMS REPEAT
02106 002126 T11,2
02107 205645 LAC (1 /PUT A ONE IN INDEX REGISTER
02110 721000 PAX
02111 203760 LAC DATA /PUT ADDRESS OF LOC DATA IN AUTO INDEX
02112 040011 DAC 11
02113 203762 LAC DATA*2 /GET CONTENTS OF DATA*2 AND COMPARE
02114 590011 SAD* 11,X /WITH SELF VIA AUTO INDEX INDEXED
02115 741000 SKP
02116 605246 ER172 JMP H172
02117 750000 CLA
02120 737004 AXR*4 /ADD 4 TO XR
02121 430011 XCT* 11,X /XCT LOCATION JXCT
02122 543762 SAD DATA*2
02123 741000 SKP
02124 605250 ER173 JMP H173
02125 104023 JMS DONE

/TEST11,2=TEST JMP* 17,JMS* 17 INDEXED
/
02126 104000 T11,2 JMS REPEAT
02127 002143 T11,3
02130 205645 LAC (1
02131 721000 PAX
02132 203770 LAC JSUB3 /PUT ADDRESS JSUB3*1 IN LOC.17
02133 040017 DAC 17
02134 630017 JMP* 17,X /JMP TO JSUB*
02135 605252 ER174 JMP H174
02136 737001 AXR*1 /ADD 1 TO XR, NEW TOTAL 2
02137 440017 ISE 17 /ADD 1 TO 17 TO MAKE IT JSUB3*3
02140 130017 JMS* 17,X
02141 605254 ER175 JMP H175
02142 104023 JMS DONE

/TEST 11,3 = TEST LAC*XR AFTER MODE CHANGE
/
02143 104000 T11,3 JMS REPEAT
02144 002157 T12,0
02145 750000 CLA
02146 723002 AAC*2
02147 721000 PAX /PLACE QUANTITY 2 INXR
02150 707764 EBA /TOGGLE MODE
02151 707762 EPA
02152 213715 LAC GET,X /USE INDEXED ADDRESSING
02153 543717 SAD GET*2
02154 741000 SKP
02155 605256 ER176 JMP H176
02156 104023 JMS DONE
.EJECT

```

```

/
/TEST 12 = TEST FOR INDEXED LAC, DAC ACCESS TO ALL FIELDS
/IN ALL BLOCKS, ONE BLOCK AT A TIME
/TEST XCT TO F1,F3,F5,F7 AND LAC* TO F0,F2,F4,F6 INDEXED
/
/TEST 12,0=TEST ACCESS TO FIELD 0
/

```

02157	104000	T12,0	JMS	REPEAT	
02160	002213		T12,1		
02161	404166		XCT	TEPW	/DOES THIS TEST EXIST?
02162	602536		JMP	T13,01	/NON-EX MEM, GO TO NEXT GROUP OF TESTS
02163	205660		LAC	(NOP	/PLACE NOP IN TEST LOC IN F0,00
02164	047000		DAC	TSTLOC	
02165	204156		LAC	F0A	/PLACE PC BITS FOR F0 IN XR
02166	344201		TAD	BLOCK	/ADD BLOCK BITS
02167	721000		PAX		
02170	760000		LAW	0	/DEPOSIT LAW 0
02171	344203		TAD	BLKCON	
02172	044204		DAC	TEMKON	
02173	057000		DAC	TSTLOC,X	
02174	217000		LAC	TSTLOC,X	
02175	544204		SAD	TEMKON	
02176	741000		SKP		
02177	605260	ER200	JMP	H200	/EITHER LAC OR DAC TO F0 FAILED
02200	204156		LAC	F0A	
02201	344201		TAD	BLOCK	
02202	721000		PAX		
02203	204156		LAC	F0A	
02204	345661		TAD	(7000	
02205	044216		DAC	POINT1	
02206	234216		LAC*	POINT1,X	
02207	544204		SAD	TEMKON	
02210	741000		SKP		
02211	605262	ER201	JMP	H201	/LAC*,X TO TSTLOC TO F0 FAILED
02212	104023		JMS	DONE	
			.EJECT		

/
/TEST 12,1 = TEST ACCESS TO FIELD 1
/

02213	104000	T12.1	JMS	REPEAT	
02214	002247		T12,2		
02215	404167		XCT	TEF1	/DOES THIS FIELD EXIST?
02216	602536		JMP	T13:01	/NON-EX MEM, GO TO NEXT GROUP OF TESTS
02217	760000		LAW	0	/PLACE LAW 0 IN TEST LOC IN F0
02220	047000		DAC	TSTLOC	
02221	204157		LAC	F1A	/PLACE PC BITS FOR F1 IN XR
02222	344201		TAD	BLOCK	/ADD BLOCK BITS
02223	721000		PAX		
02224	760001		LAW	1	/DEPOSIT LAW 1 IN TEST + CONSTANT
02225	344203		TAD	BLKCON	
02226	044204		DAC	TEMKON	
02227	057000		DAC	TSTLOC,X	/LOCATION OF F1
02230	207000		LAC	TSTLOC	/LAC TEST LOC IN F0
02231	545662		SAD	(LAW	
02232	741000		SKP		
02233	605264	ER202	JMP	H202	/ERROR, DAC OCCURRED IN F0
02234	217000		LAC	TSTLOC,X	/LAC TEST LOC IN F1
02235	544204		SAD	TEMKON	
02236	741000		SKP		
02237	605266	ER203	JMP	H203	/ERROR, EITHER LAC OR DAC TO F1 FAILED
02240	750000		CLA		
02241	417000		XCT	TSTLOC,X	/XCT TEST LOC IN F1
02242	544204		SAD	TEMKON	
02243	741000		SKP		
02244	605270	ER204	JMP	H204	/ERROR, XCT INDEXED TO F1 FAILED
02245	103416		JMS	TESTF0	/TEST IF LOC IN F0 MODIFIED
02246	104023		JMS	DONE	
			.	EJECT	

/TEST 12,2 - TEST ACCESS TO FIELD 2

```

/
T12,2  JMS      REPEAT
02247  104000
02250  002310      T12,3
02251  404170      XCT      TEF2      /DOES THIS FIELD EXIST?
02252  602536      JMP      T13,01    /NON-EX MEM, GO TO NEXT GROUP OF TESTS
02253  760000      LAW      0
02254  047000      DAC      TSTLOC
02255  204160      LAC      F2A      /PLACE PC BITS FOR F2 IN XR
02256  344201      TAD      BLOCK    /ADD BLOCK BITS
02257  721000      PAX
02260  760002      LAW      2
02261  344203      TAD      BLKCON
02262  044204      DAC      TEMKON
02263  057000      DAC      TSTLOC,X  /DAC IN TEST LOCATION OF F2
02264  207000      LAC      TSTLOC    /GET TEST LOC IN F0
02265  545662      SAD      (LAW
02266  741000      SKP
02267  605272      ER205   JMP      H205      /ERROR, DAC OCCURRED IN F0
02270  217000      LAC      TSTLOC,X  /GET TEST LOC IN F2
02271  544204      SAD      TEMKON
02272  741000      SKP
02273  605272      ER206   JMP      H205      /ERROR, LAC OR DAC INDEXED TO F2 FAILED
02274  204157      LAC      F1A
02275  344201      TAD      BLOCK
02276  721000      PAX
02277  204157      LAC      F1A
02300  345661      TAD      (7000
02301  044216      DAC      POINT1
02302  234216      LAC*    POINT1,X
02303  544204      SAD      TEMKON
02304  741000      SKP
02305  605276      ER207   JMP      H207      /ERROR, LAC* INDEXED TO LOC TO F2 FAILED
02306  103401      JMS     TESTF1    /TEST IF LOC IN F1 MODIFIED
02307  104023      JMS     DONE
          .EJECT

```

/TEST 12,3 - TEST ACCESS TO FIELD 3

/

02310	104000	T12,3	JMS	REPEAT	
02311	002344		T12,4		
02312	404171		XCT	TEST	/DOES THIS FIELD EXIST?
02313	602536		JMP	T13,01	/NO, NON-EX MEM, GO TO NEXT TEST
02314	760000		LAW	0	
02315	047000		DAC	TSTLOC	
02316	204161		LAC	F3A	/PLACE PC BITS FOR F3 IN XR
02317	344201		TAD	BLOCK	/ADD BLOCK BITS
02320	721000		PAX		
02321	760003		LAW	3	
02322	344203		TAD	BLKCON	
02323	044204		DAC	TEMKON	
02324	057000		DAC	TSTLOC,X	/DAC IN TEST LOC IN F3
02325	207000		LAC	TSTLOC	/TEST LOC IN F0
02326	545662		SAD	(LAW	
02327	741000		SKP		
02330	605300	ER210	JMP	H210	/ERROR, DAC OCCURRED IN F0
02331	217000		LAC	TSTLOC,X	/GET CONTENTS OF TEST LOC IN F3
02332	544204		SAD	TEMKON	
02333	741000		SKP		
02334	605302	ER211	JMP	H211	/ERROR, LAC OR DAC INDEXED INTO F3 FAILED
02335	750000		CLA		
02336	417000		XCT	TSTLOC,X	/XCT TEST LOC IN F3
02337	544204		SAD	TEMKON	
02340	741000		SKP		
02341	605304	ER212	JMP	H212	/ERROR, XCT INDEXED TO F3 FAILED
02342	103364		JMS	TESTF2	/TEST IF LOC IN F2,F1 MODIFIED
02343	104023		JMS	DONE	
			.EJECT		

/TEST 12,6 = TEST ACCESS TO FIELD 6

```

/
02441 104000 T12.6 JMS REPEAT
02442 002502 T12,7
02443 404174 XCT TEF0 /DOES THIS FIELD EXIST?
02444 602536 JMP T13,01 /NO, GO TO NEXT TEST
02445 760000 LAW 0
02446 047000 DAC TSTLOC
02447 204164 LAC F6A /YES, PLACE PC BITS FOR F6 IN XR
02450 344201 TAD BLOCK /ADD BLOCK BITS
02451 721000 PAX
02452 760006 LAW 6
02453 344203 TAD BLKCON
02454 044204 DAC TEMKON
02455 057000 DAC TSTLOC,X /DAC IN TEST LOC IN F6
02456 207000 LAC TSTLOC /TEST LOC IN F0
02457 545662 SAD CLAW
02460 741000 SKP
02461 605322 ER221 JMP H221 /ERROR, DAC OCCURRED IN F0
02462 217000 LAC TSTLOC,X /GET CONTENTS OF TEST LOC IN F6
02463 544204 SAD TEMKON
02464 741000 SKP
02465 605324 ER222 JMP H222 /ERROR, LAC OR DAC INDEXED TO F6 FAILED
02466 204161 LAC F3A
02467 344201 TAD BLOCK
02470 721000 PAX
02471 204161 LAC F3A
02472 345661 TAD (7000
02473 044216 DAC POINT1
02474 234216 LAC POINT1,X
02475 544204 SAD TEMKON
02476 741000 SKP
02477 605326 ER223 JMP H223 /ERROR, XCT INDEXED TO F6 FAILED
02500 103315 JMS TESTF5 /TEST IF LOC IN F5,F4,F3,F2,F1 MODIFIED
02501 104023 JMS DONE
      .EJECT

```

/TEST 12,7 - TEST ACCESS TO FIELD 7

```

/
02502 104000 T12,7 JMS REPEAT
02503 002536 T13,01
02504 404175 XCT TEST /DOES THIS FIELD EXIST?
02505 602536 JMP T13,01 /NO, GO TO NEXT TEST
02506 760000 LAW 0
02507 047000 DAC TSTLOC
02510 204165 LAC F7A /YES, PLACE PC BITS FOR F7 IN XR
02511 344201 TAD BLOCK /ADD BLOCK BITS
02512 721000 PAX
02513 760007 LAW 7
02514 344203 TAD BLKCON
02515 044204 DAC TEMKON
02516 057000 DAC TSTLOC,X /DAC IN TEST LOC IN F7
02517 207000 LAC TSTLOC /TEST LOC IN F0
02520 545662 SAD (LAW
02521 741000 SKP
02522 605332 ER224 JMP H224 /ERROR, DAC OCCURRED IN F0
02523 217000 LAC TSTLOC,X /GET CONTENTS OF TEST LOC IN F7
02524 544204 SAD TEMKON
02525 741000 SKP
02526 605332 ER225 JMP H225 /ERROR, LAC OR DAC INDEXED TO F7 FAILED
02527 750000 CLA
02530 417000 XCT TSTLOC,X /XCT TEST LOCATION IN F7
02531 544204 SAD TEMKON
02532 741000 SKP
02533 605334 ER226 JMP H226 /ERROR, XCT INDEXED TO F7 FAILED
02534 103300 JMS TESTF6 /TEST IF LOC IN F6,F5,F4,F3,F2,F1 MODIFIED
02535 104023 JMS DONE
,EJECT

```

/
 /TEST 13 = RELOCATE TEST PROGRAM IN UPPER MEMORY PAGE IF IT EXISTS
 /AND TRANSFER CONTROL TO THIS PROGRAM TO VERIFY INDEXED OPERATION

/TEST 13,01 - TEST F0 FOR LAC+DAC+ AUTO INDEX INDEXED TO F0

02536	144211	T13,01	OEM	COUNT	
02537	204202		LAC	BLKNO	/ARE WE TESTING BLOCK 0?
02540	741202		SNA		/NO, PERFORM THESE TESTS
02541	602612		JMP	T13,11	/YES, SKP THESE TESTS
02542	104000		JMS	REPEAT	
02543	002561			T13,02	
02544	404166		XCT	TEF0	
02545	603435		JMP	TSTOP	
02546	104364		JMS	ZERMEM	/ZERO ALL MEMORY EXCEPT F0 ONCE
02547	204156		LAC	F0A	/GET ADDRESS OF BLOCK + FIELD
02550	344201		TAD	BLOCK	
02551	103502		JMS	MOVE1	/YES, MOVE TEST PROGRAM TO THIS FIELD
02552	760001		LAW	1	/TEST CONSTANT
02553	344203		TAD	BLKCON	
02554	103665		JMS	MINIT	/INITIALIZE CONSTANTS
02555	117000		JMS	TSTLOC,X	/XCT TEST PROGRAM IN RESIDENT FIELD
02556	605336	ER227	JMP	H227	/LAC+ AUTO INDEX INDEXED TO F0 FAILED
02557	605340	ER230	JMP	H230	/DAC+ AUTO INDEX INDEXED TO F0 FAILED
02560	104023		JMS	DONE	

/TEST 13,02 - TEST F0 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

02561	104000	T13,02	JMS	REPEAT	/CAL FROM HERE INDICATES JMS+XR FAILED
02562	002576		T13,03		
02563	204156		LAC	F0A	
02564	344201		TAD	BLOCK	
02565	103514		JMS	MOVE2	/YES, MOVE TEST PROGRAM TO THIS FIELD
02566	760002		LAW	2	/TEST CONSTANT
02567	344203		TAD	BLKCON	
02570	103665		JMS	MINIT	/INITIALIZE CONSTANTS
02571	117000		JMS	TSTLOC,X	/XCT TEST PROGRAM IN RESIDENT FIELD
02572	605342	ER231	JMP	H231	/LAC INDEXED IN RESIDENT FIELD FAILED
02573	605344	ER232	JMP	H232	/DAC INDEXED IN RESIDENT FIELD FAILED
02574	605346	ER233	JMP	H233	/SAD INDEXED IN RESIDENT FIELD FAILED
02575	104023		JMS	DONE	

/TEST 13,03 - TEST F0 FOR DAC AND TAD INDEXED TO F0

02576	104000	T13,03	JMS	REPEAT	
02577	002612		T13,11		
02600	204156		LAC	F0A	
02601	344201		TAD	BLOCK	
02602	103526		JMS	MOVE3	/YES, MOVE TEST PROGRAM TO THIS FIELD
02603	760003		LAW	3	/TEST CONSTANT
02604	344203		TAD	BLKCON	
02605	103665		JMS	MINIT	/INITIALIZE CONSTANTS
02606	117000		JMS	TSTLOC,X	/XCT TEST PROGRAM IN RESIDENT FIELD
02607	605350	ER234	JMP	H234	/DAC INDEXED TO F0 FAILED
02610	605352	ER235	JMP	H235	/TAD INDEXED TO F0 FAILED
02611	104023		JMS	DONE	

,EJECT

/TEST 13,11 = TEST F1 FOR LAC+DAC+ AUTO INDEX INDEXED TO F0

```

/
02612 144211 T13,11 DZM COUNT
02613 104000 JMS REPEAT
02614 022632 T13,12
02615 404167 XCT TEF1 /DOES THIS FIELD EXIST?
02616 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02617 104364 JMS ZERMEM /ZERO ALL MEMORY EXCEPT F0 ONCE
02620 204157 LAC F1A /GET ADDRESS OF BLOCK + FIELD
02621 344201 TAD BLOCK
02622 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
02623 760101 LAW 101 /TEST CONSTANT
02624 344203 TAD BLKCON
02625 103665 JMS MINIT /INITIALIZE CONSTANTS
02626 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02627 605354 ER236 JMP H236 /LAC+ AUTO INDEX INDEXED TO F0 FAILED
02630 605356 ER237 JMP H237 /DAC+ AUTO INDEX INDEXED TO F0 FAILED
02631 104023 JMS DONE

```

/TEST 13,12 = TEST F1 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
02632 104000 T13,12 JMS REPEAT /CAL FROM HERE INDICATES JMS+XR FAILED
02633 022651 T13,13
02634 404167 XCT TEF1 /DOES THIS FIELD EXIST?
02635 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02636 204157 LAC F1A
02637 344201 TAD BLOCK
02640 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
02641 760102 LAW 102 /TEST CONSTANT
02642 344203 TAD BLKCON
02643 103665 JMS MINIT /INITIALIZE CONSTANTS
02644 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02645 605360 ER240 JMP H240 /LAC INDEXED IN RESIDENT FIELD FAILED
02646 605362 ER241 JMP H241 /DAC INDEXED IN RESIDENT FIELD FAILED
02647 605364 ER242 JMP H242 /SAD INDEXED IN RESIDENT FIELD FAILED
02650 104023 JMS DONE

```

/TEST 13,13 = TEST F1 FOR DAC AND TAD INDEXED TO F0

```

/
02651 104000 T13,13 JMS REPEAT
02652 022667 T13,21
02653 404167 XCT TEF1 /DOES THIS FIELD EXIST?
02654 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02655 204157 LAC F1A
02656 344201 TAD BLOCK
02657 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
02660 760103 LAW 103 /TEST CONSTANT
02661 344203 TAD BLKCON
02662 103665 JMS MINIT /INITIALIZE CONSTANTS
02663 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02664 605366 ER243 JMP H243 /DAC INDEXED TO F0 FAILED
02665 625373 ER244 JMP H244 /TAD INDEXED TO F0 FAILED
02666 104023 JMS DONE
,EJECT

```

/TEST 13,21 - TEST F2 FOR LAC+DAC+ AUTO INDEX INDEXED TO F0

```

/
02667 104023 T13,21 JMS REPEAT
02670 102706 T13,22
02671 404170 XCT TEF2 /DOES THIS FIELD EXIST?
02672 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02673 104364 JMS ZERMEM /ZERO ALL MEMORY EXCEPT F0 ONCE
02674 204160 LAC F2A
02675 344201 TAD BLOCK
02676 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
02677 760201 LAW 201 /TEST CONSTANT
02700 344203 TAD BLKCON
02701 103665 JMS MINIT /INITIALIZE CONSTANTS
02702 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02703 605372 ER245 JMP H245 /LAC+ AUTO INDEX INDEXED TO F0 FAILED
02704 605374 ER246 JMP H246 /DAC+ AUTO INDEX INDEXED TO F0 FAILED
02705 104023 JMS DONE

```

/TEST 13,22 - TEST F2 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
02706 104000 T13,22 JMS REPEAT /CAL FROM HERE INDICATES JMS+XR FAILED
02707 202725 T13,23
02710 404170 XCT TEF2 /DOES THIS FIELD EXIST?
02711 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02712 204160 LAC F2A
02713 344201 TAD BLOCK
02714 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
02715 760202 LAW 202 /TEST CONSTANT
02716 344203 TAD BLKCON
02717 103665 JMS MINIT /INITIALIZE CONSTANTS
02720 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02721 605376 ER250 JMP H250 /LAC INDEXED IN RESIDENT FIELD FAILED
02722 605400 ER251 JMP H251 /DAC INDEXED IN RESIDENT FIELD FAILED
02723 605402 ER252 JMP H252 /SAD INDEXED IN RESIDENT FIELD FAILED
02724 104023 JMS DONE

```

/TEST 13,23 - TEST F2 FOR DAC AND TAD INDEXED TO F0

```

/
02725 104000 T13,23 JMS REPEAT
02726 202743 T13,31
02727 404170 XCT TEF2 /DOES THIS FIELD EXIST?
02730 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02731 204160 LAC F2A
02732 344201 TAD BLOCK
02733 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
02734 760203 LAW 203 /TEST CONSTANT
02735 344203 TAD BLKCON
02736 103665 JMS MINIT /INITIALIZE CONSTANTS
02737 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02740 605404 ER253 JMP H253 /DAC INDEXED TO F0 FAILED
02741 605406 ER254 JMP H254 /TAD INDEXED TO F0 FAILED
02742 104023 JMS DONE

```

.EJECT

/TEST 13,31 - TEST F3 FOR LAC*DAC* AUTO INDEX INDEXED TO F0

```

/
02743 104000 T13,31 JMS REPEAT
02744 002762 T13,32
02745 404171 XCT TEFS /DOES THIS FIELD EXIST?
02746 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02747 104364 JMS ZERMEN /ZERO ALL MEMORY EXCEPT F0 ONCE
02750 204161 LAC F3A
02751 344201 TAD BLOCK
02752 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
02753 760301 LAW 301 /TEST CONSTANT
02754 344203 TAD BLKCON
02755 103665 JMS MINIT /INITIALIZE CONSTANTS
02756 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02757 605410 ER255 JMP H255 /LAC* AUTO INDEX INDEXED TO F0 FAILED
02760 605412 ER260 JMP H260 /DAC* AUTO INDEX INDEXED TO F0 FAILED
02761 104023 JMS DONE

```

/TEST 13,32 - TEST F3 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
02762 104000 T13,32 JMS REPEAT /CAL FROM HERE INDICATES JMS*XR FAILED
02763 003001 T13,33
02764 404171 XCT TEFS /DOES THIS FIELD EXIST?
02765 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
02766 204161 LAC F3A
02767 344201 TAD BLOCK
02770 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
02771 760302 LAW 302 /TEST CONSTANT
02772 344203 TAD BLKCON
02773 103665 JMS MINIT /INITIALIZE CONSTANTS
02774 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
02775 605414 ER261 JMP H261 /LAC INDEXED IN RESIDENT FIELD FAILED
02776 605416 ER262 JMP H262 /DAC INDEXED IN RESIDENT FIELD FAILED
02777 605420 ER263 JMP H263 /SAD INDEXED IN RESIDENT FIELD FAILED
03000 104023 JMS DONE

```

/TEST 13,33 - TEST F3 FOR DAC AND TAD INDEXED TO F0

```

/
03001 104000 T13,33 JMS REPEAT
03002 003017 T13,41
03003 404171 XCT TEFS /DOES THIS FIELD EXIST?
03004 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03005 204161 LAC F3A
03006 344201 TAD BLOCK
03007 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
03010 760303 LAW 303 /TEST CONSTANT
03011 344203 TAD BLKCON
03012 103665 JMS MINIT /INITIALIZE CONSTANTS
03013 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03014 605422 ER264 JMP H264 /DAC INDEXED TO F0 FAILED
03015 605424 ER265 JMP H265 /TAD INDEXED TO F0 FAILED
03016 104023 JMS DONE
,EJECT

```

/TEST 13,41 = TEST F4 FOR LAC+DAC+ AUTO INDEX INDEXED TO F0

```

/
23017 104000 T13,41 JMS REPEAT
23020 003036 T13,42
23021 404172 XCT TEF4 /DOES THIS FIELD EXIST?
23022 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23023 104364 JMS ZERMEM /ZERO ALL MEMORY EXCEPT F0 ONCE
23024 204162 LAC F4A
23025 344201 TAD BLOCK
23026 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
23027 760401 LAW 401 /TEST CONSTANT
23030 344203 TAD BLKCON
23031 103665 JMS MINIT /INITIALIZE CONSTANTS
23032 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23033 605426 ER270 JMP H270 /LAC+ AUTO INDEX INDEXED TO F0 FAILED
23034 605430 ER271 JMP H271 /DAC+ AUTO INDEX INDEXED TO F0 FAILED
23035 104023 JMS DONE

```

/TEST 13,42 = TEST F4 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
23036 104000 T13,42 JMS REPEAT /CAL FROM HERE INDICATES JMS+XR FAILED
23037 003055 T13,43
23040 404172 XCT TEF4 /DOES THIS FIELD EXIST?
23041 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23042 204162 LAC F4A
23043 344201 TAD BLOCK
23044 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
23045 760402 LAW 402 /TEST CONSTANT
23046 344203 TAD BLKCON
23047 103665 JMS MINIT /INITIALIZE CONSTANTS
23050 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23051 605432 ER272 JMP H272 /LAC INDEXED IN RESIDENT FIELD FAILED
23052 605434 ER273 JMP H273 /DAC INDEXED IN RESIDENT FIELD FAILED
23053 605436 ER274 JMP H274 /SAD INDEXED IN RESIDENT FIELD FAILED
23054 104023 JMS DONE

```

/TEST 13,43 = TEST F4 FOR DAC AND TAD INDEXED TO F0

```

/
23055 104000 T13,43 JMS REPEAT
23056 003073 T13,51
23057 404172 XCT TEF4 /DOES THIS FIELD EXIST?
23060 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23061 204162 LAC F4A
23062 344201 TAD BLOCK
23063 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
23064 760403 LAW 403 /TEST CONSTANT
23065 344203 TAD BLKCON
23066 103665 JMS MINIT /INITIALIZE CONSTANTS
23067 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23070 605442 ER275 JMP H275 /DAC INDEXED TO F0 FAILED
23071 605442 ER300 JMP H300 /TAD INDEXED TO F0 FAILED
23072 104023 JMS DONE

```

,EJECT

/TEST 13,51 - TEST F5 FOR LAC+DAC+ AUTO INDEX INDEXED TO F0

```

/
03073 124027 T13,51 JMS REPEAT
03074 223112 T13,52
03075 404173 XCT TEF5 /DOES THIS FIELD EXIST?
03076 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03077 124364 JMS ZERMEM /ZERO ALL MEMORY EXCEPT F0 ONCE
03100 204163 LAC F5A
03101 344201 TAD BLOCK
03102 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
03103 760501 LAW 501 /TEST CONSTANT
03104 344203 TAD BLKCON
03105 123665 JMS MINIT /INITIALIZE CONSTANTS
03106 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03107 605444 ER301 JMP H301 /LAC+ AUTO INDEX INDEXED TO F0 FAILED
03110 605446 ER302 JMP H302 /DAC+ AUTO INDEX INDEXED TO F0 FAILED
23111 104023 JMS DONE

```

/TEST 13,52 - TEST F5 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
03112 104000 T13,52 JMS REPEAT /CAL FROM HERE INDICATES JMS+XR FAILED
03113 203131 T13,53
03114 404173 XCT TEF5 /DOES THIS FIELD EXIST?
03115 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03116 204163 LAC F5A
03117 344201 TAD BLOCK
03120 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
03121 760502 LAW 502 /TEST CONSTANT
03122 344203 TAD BLKCON
03123 103665 JMS MINIT /INITIALIZE CONSTANTS
03124 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03125 605452 ER303 JMP H303 /LAC INDEXED IN RESIDENT FIELD FAILED
03126 605452 ER304 JMP H304 /DAC INDEXED IN RESIDENT FIELD FAILED
03127 605454 ER305 JMP H305 /SAD INDEXED IN RESIDENT FIELD FAILED
03130 104023 JMS DONE

```

/TEST 13,53 - TEST F5 FOR DAC AND TAD INDEXED TO F0

```

/
03131 104002 T13,53 JMS REPEAT
03132 203147 T13,61
03133 404173 XCT TEF5 /DOES THIS FIELD EXIST?
03134 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03135 204163 LAC F5A
03136 344201 TAD BLOCK
03137 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
03140 760503 LAW 503 /TEST CONSTANT
03141 344223 TAD BLKCON
03142 103665 JMS MINIT /INITIALIZE CONSTANTS
03143 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03144 605456 ER310 JMP H310 /DAC INDEXED TO F0 FAILED
03145 605467 ER311 JMP H311 /TAD INDEXED TO F0 FAILED
23146 104023 JMS DONE
,EJECT

```

/TEST 13,61 = TEST F6 FOR LAC*DAC* AUTO INDEX INDEXED TO F0

```

/
23147 104003 T13,61 JMS REPEAT
23150 203166 T13,62
23151 404174 XCT TEF6 /DOES THIS FIELD EXIST?
23152 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23153 104364 JMS ZERMEN /ZERO ALL MEMORY EXCEPT F0 ONCE
23154 204164 LAC F6A
23155 344201 TAD BLOCK
23156 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
23157 760601 LAW 601 /TEST CONSTANT
23160 344203 TAD BLKCON
23161 103665 JMS MINIT /INITIALIZE CONSTANTS
23162 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23163 605462 ER312 JMP H312 /LAC* AUTO INDEX INDEXED TO F0 FAILED
23164 605464 ER313 JMP H313 /DAC* AUTO INDEX INDEXED TO F0 FAILED
23165 104023 JMS DONE

```

/TEST 13,62 = TEST F FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
23166 104002 T13,62 JMS REPEAT /CAL FROM HERE INDICATES JMS*XR FAILED
23167 203205 T13,63
23170 404174 XCT TEF6 /DOES THIS FIELD EXIST?
23171 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23172 204164 LAC F6A
23173 344201 TAD BLOCK
23174 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
23175 760602 LAW 602 /TEST CONSTANT
23176 344203 TAD BLKCON
23177 103665 JMS MINIT /INITIALIZE CONSTANTS
23200 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23201 605466 ER314 JMP H314 /LAC INDEXED IN RESIDENT FIELD FAILED
23202 605470 ER315 JMP H315 /DAC INDEXED IN RESIDENT FIELD FAILED
23203 605472 ER320 JMP H320 /SAD INDEXED IN RESIDENT FIELD FAILED
23204 104023 JMS DONE

```

/TEST 13,63 = TEST F6 FOR DAC AND TAD INDEXED TO F0

```

/
23205 104000 T13,63 JMS REPEAT
23206 203223 T13,71
23207 404174 XCT TEF6 /DOES THIS FIELD EXIST?
23210 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
23211 204164 LAC F6A
23212 344201 TAD BLOCK
23213 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
23214 760603 LAW 603 /TEST CONSTANT
23215 344203 TAD BLKCON
23216 103665 JMS MINIT /INITIALIZE CONSTANTS
23217 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
23220 605474 ER321 JMP H321 /DAC INDEXED TO F0 FAILED
23221 605476 ER322 JMP H322 /TAD INDEXED TO F0 FAILED
23222 104023 JMS DONE

```

.EJECT

/TEST 13,71 = TEST F7 FOR LAC+DAC* AUTO INDEX INDEXED TO F0

```

/
03223 104000 T13,71 JMS REPEAT
03224 003242 T13,72
03225 404175 XCT TEF7 /DOES THIS FIELD EXIST?
03226 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03227 104364 JMS ZERMEM /ZERO ALL MEMORY EXCEPT F0 ONCE
03230 204165 LAC F7A
03231 344201 TAD BLOCK
03232 103502 JMS MOVE1 /YES, MOVE TEST PROGRAM TO THIS FIELD
03233 760701 LAW 701 /TEST CONSTANT
03234 344203 TAD BLKCON
03235 103665 JMS MINIT /INITIALIZE CONSTANTS
03236 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03237 605500 ER323 JMP H323 /LAC* AUTO INDEX INDEXED TO F0 FAILED
03240 605502 ER324 JMP H324 /DAC* AUTO INDEX INDEXED TO F0 FAILED
03241 104023 JMS DONE

```

/TEST 13,72 = TEST F7 FOR LAC,DAC,SAD AND JMS INDEXED IN RESIDENT FIELD

```

/
03242 104000 T13,72 JMS REPEAT /CAL FROM HERE INDICATES JMS+XR FAILED
03243 003261 T13,73
03244 404175 XCT TEF7 /DOES THIS FIELD EXIST?
03245 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03246 204165 LAC F7A
03247 344201 TAD BLOCK
03250 103514 JMS MOVE2 /YES, MOVE TEST PROGRAM TO THIS FIELD
03251 760702 LAW 702 /TEST CONSTANT
03252 344203 TAD BLKCON
03253 103665 JMS MINIT /INITIALIZE CONSTANTS
03254 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03255 605504 ER325 JMP H325 /LAC INDEXED IN RESIDENT FIELD FAILED
03256 605506 ER330 JMP H330 /DAC INDEXED IN RESIDENT FIELD FAILED
03257 605510 ER331 JMP H331 /SAD INDEXED IN RESIDENT FIELD FAILED
03260 104023 JMS DONE

```

/TEST 13,73 = TEST F7 FOR DAC AND TAD INDEXED TO F0

```

/
03261 104000 T13,73 JMS REPEAT
03262 003435 TSTOP
03263 404175 XCT TEF7 /DOES THIS FIELD EXIST?
03264 603435 JMP TSTOP /NO, GO TO NEXT GROUP OF TESTS
03265 204165 LAC F7A
03266 344201 TAD BLOCK
03267 103526 JMS MOVE3 /YES, MOVE TEST PROGRAM TO THIS FIELD
03270 760703 LAW 703 /TEST CONSTANT
03271 344203 TAD BLKCON
03272 103665 JMS MINIT /INITIALIZE CONSTANTS
03273 117000 JMS TSTLOC,X /XCT TEST PROGRAM IN RESIDENT FIELD
03274 605512 ER332 JMP H332 /DAC INDEXED TO F0 FAILED
03275 605514 ER333 JMP H333 /TAD INDEXED TO F0 FAILED
03276 104023 JMS DONE
03277 603435 JMP TSTOP

```

.EJECT

52

/TEST SUBROUTINES TO CHECK FOR ILLEGAL MEMORY ADDRESSING
 /TEST FIELDS TO SEE IF TEST LOCATION MODIFIED

03300	000000	TESTF6	0		
03301	203300	LAC	,=1		
03302	043434	DAC	TRET	/SAVE FOR RETURN	
03303	204164	LAC	F6A	/PLACE FIELD ADDRESS IN XR	
03304	344201	TAD	BLOCK		
03305	721000	PAX			
03306	760006	LAW	6		
03307	344203	TAD	BLKCON		
03310	044204	DAC	TEMKON		
03311	217000	LAC	TSTLOC,X		
03312	544204	SAD	TEMKON		
03313	603320	JMP	,+5		
03314	605516	ER334	JMP	H334	/ERROR TEST LOC IN F6 MODIFIED
03315	000000	TESTF5	0		
03316	203315	LAC	,=1		
03317	043434	DAC	TRET	/SAVE FOR RETURN	
03320	204163	LAC	F5A	/PLACE FIELD ADDRESS IN XR	
03321	344201	TAD	BLOCK		
03322	721000	PAX			
03323	760005	LAW	5		
03324	344203	TAD	BLKCON		
03325	044204	DAC	TEMKON		
03326	217000	LAC	TSTLOC,X		
03327	544204	SAD	TEMKON		
03330	603335	JMP	,+5		
03331	605520	ER335	JMP	H335	/ERROR, TEST LOC IN F5 MODIFIED
03332	000000	TESTF4	0		
03333	203332	LAC	,=1		
03334	043434	DAC	TRET	/SAVE FOR RETURN	
03335	204162	LAC	F4A	/PLACE FIELD ADDRESS IN XR	
03336	344201	TAD	BLOCK		
03337	721000	PAX			
03340	760004	LAW	4		
03341	344203	TAD	BLKCON		
03342	044204	DAC	TEMKON		
03343	217000	LAC	TSTLOC,X		
03344	544204	SAD	TEMKON		
03345	603352	JMP	,+5		
03346	605522	ER340	JMP	H340	/ERROR, TEST LOC IN F4 MODIFIED
03347	000000	TESTF3	0		
03350	203347	LAC	,=1		
03351	043434	DAC	TRET	/SAVE FOR RETURN	
03352	204161	LAC	F3A	/PLACE FIELD ADDRESS IN XR	
03353	344201	TAD	BLOCK		
03354	721000	PAX			
03355	760003	LAW	3		
03356	344203	TAD	BLKCON		
03357	044204	DAC	TEMKON		
03360	217000	LAC	TSTLOC,X		
03361	544204	SAD	TEMKON		
03362	603367	JMP	,+5		
03363	605524	ER341	JMP	H341	/ERROR, TEST LOC IN F3 MODIFIED
			,EJECT		

03364	000000	TESTF2	0		
03365	203364		LAC	,=1	
03366	043434		DAC	TRET	/SAVE FOR RETURN
03367	204160		LAC	F2A	/PLACE FIELD ADDRESS IN XR
03370	344201		TAD	BLOCK	
03371	721000		PAX		
03372	760002		LAW	2	
03373	344203		TAD	BLKCON	
03374	044204		DAC	TEMKON	
03375	217000		LAC	TSTLOC,X	
03376	544204		SAD	TEMKON	
03377	603404		JMP	,+5	
03400	605526	ER342	JMP	H342	/ERROR, TEST LOC IN F2 MODIFIED
03401	000000	TESTF1	0		
03402	203401		LAC	,=1	
03403	043434		DAC	TRET	/SAVE FOR RETURN
03404	204157		LAC	F1A	/PLACE FIELD ADDRESS IN XR
03405	344201		TAD	BLOCK	
03406	721000		PAX		
03407	760001		LAW	1	
03410	344203		TAD	BLKCON	
03411	044204		DAC	TEMKON	
03412	217000		LAC	TSTLOC,X	
03413	544204		SAD	TEMKON	
03414	603421		JMP	,+5	
03415	605530	ER343	JMP	H343	/ERROR, TEST LOC IN F1 MODIFIED
03416	000000	TESTF0	0		
03417	203416		LAC	,=1	
03420	043434		DAC	TRET	/SAVE FOR RETURN
03421	204156		LAC	F0A	/PLACE FIELD ADDRESS IN XR
03422	344201		TAD	BLOCK	
03423	721000		PAX		
03424	760000		LAW	0	
03425	344203		TAD	BLKCON	
03426	044204		DAC	TEMKON	
03427	217000		LAC	TSTLOC,X	
03430	544204		SAD	TEMKON	
03431	741000		SKP		
03432	605532	ER344	JMP	H344	/ERROR, TEST LOC IN F0 MODIFIED
03433	623434		JMP	TRET	
03434	000000	TRET	0		
				.EJECT	

/PRINT END MESSAGE IF DESIRED AND RESTART

03435	204202	TSTOP	LAC	BLKNO	/HAS THE LAST BLOCK BEEN TESTED?
03436	545652		SAD	(3	
03437	603462		JMP	TST01	/YES, EXIT
03440	204201		LAC	BLOCK	/NO, IS CORE AVAILABLE FOR NEXT BLOCK
03441	740031		CMA! IAC		
03442	344177		TAD	CBLK	
03443	741200		SNA		
03444	603462		JMP	TST01	/NO, EXIT
03445	444202		ISZ	BLKNO	/YES, UPDATE BLKNO
03446	204203		LAC	BLKCON	
03447	345663		TAD	(10	
03450	044203		DAC	BLKCON	/UPDATE BLOCK CONSTANT
03451	204201		LAC	BLOCK	
03452	345656		TAD	(100000	
03453	044201		DAC	BLOCK	/UPDATE BLOCK PC BITS
03454	144211		DZM	COUNT	
03455	204200		LAC	CBLKX	
03456	345664		TAD	(-100000	/ADJUST FOR CORE
03457	044200		DAC	CBLKX	/ALREADY TESTED
03460	104104		JMS	SENS	/SET CORE SWITCHES
03461	602157		JMP	T12,0	/DO EXTENDED MEM TEST FOR NEXT BLOCK
03462	750004	TST01	LAS		
03463	505665		AND	(1000	/MASK FOR BIT 8 QUICK VERIFY
03464	740200		SEA		
03465	603475		JMP	TS02	/QV=1 PRINT END MESSAGE
03466	750004		LAS		/QV=0 CHECK BIT 11
03467	505641		AND	(100	
03470	740200		SEA		
03471	600200		JMP	T1,0	/BIT 11=1, RESTART WITHOUT PRINTING.
03472	104224		JMS	TYPE	
03473	205534		MEND		
03474	600200		JMP	T1,0	
03475	104224	TS02	JMS	TYPE	/PRINT END AND QUICK VERIFY MESSAGE
03476	005534		MEND		
03477	104224		JMS	TYPE	
03500	005541		MOVF		
03501	600200		JMP	T1,0	
			EJECT		

/SUBROUTINE TO MOVE TEST PROGRAMS FOR INDEXED OPERATION IN
/EXTENDED MEMORY TO OTHER FIELDS FROM FIELD 2

```

/
03502 000000 MOVE1 2
03503 721000 PAX
03504 044205 DAC FIELD /PLACE FIELD BITS IN XR
03505 345666 TAD (ENDT1=TSTP1 /SAVE IN FIELD ALSO
03506 722000 PAL /ADD THE NUMBER OF WORDS IN TEST PROGRAM
03507 205667 LAC (TSTP1=1 /ADD TSTLOC AND SAVE IN LR
03510 103540 JMS MOVES
03511 204205 LAC FIELD /RESTORE FIELD BITS IN XR
03512 721000 PAX
03513 623502 JMP* MOVE1
03514 000000 MOVE2 0
03515 721000 PAX
03516 044205 DAC FIELD
03517 345670 TAD (ENDT2=TSTP2
03520 722000 PAL
03521 205671 LAC (TSTP2=1
03522 103540 JMS MOVES
03523 204205 LAC FIELD
03524 721000 PAX
03525 623514 JMP* MOVE2
03526 000000 MOVE3 0
03527 721000 PAX
03530 044205 DAC FIELD
03531 345672 TAD (ENDT3=TSTP3
03532 722000 PAL
03533 205673 LAC (TSTP3=1
03534 103540 JMS MOVES
03535 204205 LAC FIELD
03536 721000 PAX
03537 623526 JMP* MOVE3
03540 000000 MOVES 0
03541 040010 DAC 10
03542 220010 LAC* 10 /GET WORD
03543 057000 DAC TSTLOC,X /DEPOSIT IN NEW FIELD STARTING AT 7222
03544 725001 AXS+1 /ADD 1 TO XR, SKIP IF TRANSFER COMPLETE
03545 603542 JMP ,=3 /GET NEXT WORD
03546 623540 JMP* MOVES /TRANSFER COMPLETE, EXIT
.EJECT

```

/RELOCATABLE TEST ROUTINES TO VERIFY INDEXED OPERATIONS IN
 /UPPER MEMORY FIELDS. THESE ROUTINES ARE MOVED TO UPPER MEMORY
 /AND PROGRAM CONTROL IS TRANSFERED TO THEM.

/TSTP1=FIRST TEST ROUTINE FOR INDEXED OPERATION IN EX=MEM
 /TESTS FOR LAC AND DAC INDIRECT AUTO INDEX INDEXED.

03547	003547	TSTP1	TSTP1	
03550	140010		DZM	10
03551	140017		DZM	17
03552	140100		DZM	100
03553	750030		CLAIAC	
03554	740030		IAC	/PUT 2 IN XR
03555	721000		PAX	
03556	230017		LAC*	17,X
03557	740001		CMA	/GET CONTENTS OF LOC 13 IN F0
03560	070016		DAC*	16,X
03561	540050		SAD	50
03562	741000		SKP	/SAVE COMPLIMENT IN LOC 14 IN F0
03563	607023		JMP	TSTLOC*23
03564	447000		ISZ	TSTLOC
03565	230010		LAC*	10,X
03566	540050		SAD	50
03567	741000		SKP	/WAS LOC 17 IN F0 ADDRESSED CORRECTLY?
03570	607023		JMP	TSTLOC*23
03571	447000		ISZ	TSTLOC
03572	730000		PLA	/YES, CONTINUE TESTING
03573	740031		CMAIAC	/NO, EXIT
03574	721000		PAX	/SETUP FOR GOOD EXIT
03575	637000		JMP*	TSTLOC,X
03576	000000	ENDT1	0	/GOOD EXIT
			.EJECT	

/SECOND RELOCATABLE TEST ROUTINE FOR INDEXED OPERATION IN EX=MEM
 /TESTS FOR LAC, DAC, SAD AND JMS INDEXED IN RESIDENT FIELD

```

03577 003577 TSTP2 TSTP2
03600 750030 CLA:IAC /PLACE 2 IN XR
03601 740030 IAC
03602 721000 PAX
03603 140012 DEM 12
03604 140053 DZM 53
03605 140054 DZM 54
03606 140055 DZM 55
03607 210776 LAC 776,X /GET CONTENTS OF LOC 1000 IN RESIDENT FIELD
03610 050010 DAC 10,X /DAC IN LOC 12 IN RESIDENT FIELD
03611 541000 SAD 1000 /DID LAC+XR SUCCEED?
03612 741000 SKP /YES, CONTINUE
03613 607032 JMP TSTLOC+32 /NO, ERROR EXIT
03614 447000 ISZ TSTLOC /SETUP FOR GOOD EXIT
03615 200012 LAC 12
03616 541000 SAD 1000 /DID DAC+XR SUCCEED?
03617 741000 SKP /YES, CONTINUE
03620 607032 JMP TSTLOC+32 /NO, ERROR EXIT
03621 447000 ISZ TSTLOC /SETUP FOR GOOD EXIT
03622 550776 SAD 776,X /TRY TO SAD+XR
03623 741000 SKP /GOOD, CONTINUE
03624 627000 JMP* TSTLOC /BAD, ERROR EXIT
03625 447000 ISZ TSTLOC /SETUP FOR GOOD EXIT
03626 110053 JMS 53,X /ATTEMPT A JMS+XR
03627 000000 CAL /IF CAL FROM HERE JMP* INDEXED
03630 000000 CAL /IN RESIDENT FIELD FAILED
03631 730000 PLA /PLACE COMP OF PC BITS 1+2 IN XR
03632 740031 CHA:IAC
03633 721000 PAX
03634 637000 JMP* TSTLOC,X
03635 000000 ENDT2 0
      .EJECT

```

/THIRD RELOCATABLE TEST ROUTINE FOR INDEXED OPERATION IN EX=MEM
 /TESTS FOR DAC AND TAD INDEXED TO F0

```

03636 003636 TSTP3 TSTP3
03637 724000 PXA /GET FIELD BITS, FORM TWOS COMPLIMENT
03640 740031 CMAIAC
03641 721000 PAX
03642 201000 LAC 1000 /GET FIELD TEST CONSTANT
03643 250100 DAC 100,X /DEPOSIT IN LOC 100 IN F0
03644 210100 LAC 100,X /GET CONTENTS OF LOC 100 IN F0
03645 541000 SAD 1000 /TEST IF DAC INDEXED TO F0 WORKED
03646 741000 SKP /YES, CONTINUE
03647 607022 JMP TSTLOC+22 /NO, EXIT
03650 447000 ISZ TSTLOC /SETUP FOR GOOD EXIT
03651 750000 CLA
03652 140100 DZH 100 /CLEAR LOC 100 IN RESIDENT FIELD
03653 350100 TAD 100,X /TAD LOC 100 FROM F0
03654 541000 SAD 1000 /DID TAD INDEXED TO F0 WORK
03655 741000 SKP
03656 607022 JMP TSTLOC+22 /NO, EXIT
03657 447000 ISZ TSTLOC /SETUP FOR GOOD EXIT
03660 730000 PLA /PLACE COMP OF PC BITS 1+2 IN XR
03661 740031 CMAIAC
03662 721000 PAX
03663 637000 JMP* TSTLOC,X /EXIT
03664 000000 ENDS 0
,EJECT

```

/SUBROUTINE TO INITIALIZE CONSTANTS AND OPERATORS FOR EX=MEM TEST

```

/
03665 000000  MINIT 0
03666 051000  DAC 1000,X /DEPOSIT TEST CONSTANT IN LOC 1000 IN RESIDENT FIELD
03667 140014  D2M 14
03670 140100  D2M 100
03671 205674  LAC (575634
03672 040013  DAC 13
03673 205675  LAC (11
03674 040016  DAC 16
03675 040010  DAC 10
03676 205663  LAC (10
03677 040017  DAC 17
03700 150776  D2M 776,X
03701 205641  LAC (100
03702 050040  DAC 40,X
03703 205676  LAC (202143
03704 050050  DAC 50,X
03705 205677  LAC (14
03706 050051  DAC 51,X
03707 205700  LAC (JMP* 55,X
03710 050056  DAC 56,X
03711 724000  PXA /PUT XR IN LR
03712 505701  AND (300000
03713 722000  PAL /ONLY BLOCK BITS 1+2
03714 623665  JMP* MINIT

/
/REFERENCE AREA FOR CORRECT INDEXING TESTS
/
/USED FOR LAC,XOR,SAD
03715 003715  GET GET /PUT PROGRAM COUNTER IN OWN LOCATION
03716 003716  GET+1
03717 003717  GET+2
/USED FOR DAC,D2M,ISZ,AND
03720 000000  PUT 0 /IF DAC INDEXED PLACED QUANTITY HERE IT
03721 000000  0
03722 000000  0
03723 000000  0
03724 000000  0
/USE FOR JMS
03725 000000  SUB1 0
03726 623725  JMP* SUB1
03727 000000  SUB1X 0 /JMS HERE IF INDEXED CORRECTLY
03730 443727  ISZ SUB1X
03731 623727  JMP* SUB1X
,EJECT

```

03732	601660	/USED FOR JMP		
03733	601661	SUB2	JMP	ER145
			JMP	ER145+1
		/USE FOR XCT		
03734	601676	SUB3	JMP	ER147
03735	601677		JMP	ER147+1
		/USED FOR ADD,TAD		
03736	000007	TOTE	7	
03737	000070		70	
03740	200702		700	
		/USED FOR CAL*		
03741	000002	SUB4	0	/CAL HERE IF GOOD
03742	443741		ISE	SUB4
03743	623741		JMP*	SUB4
03744	000000	SUB4X	0	/CAL HERE IF BAD
03745	623744		JMP*	SUB4X
		/USED FOR LAC*,DAC*		
03746	003746	IGET	IGET	
03747	000000		0	
		/USED FOR JMS*,JMP*		
03750	003722	JGET	SUB1=3	
03751	003752	JSUB2	JSUB2+1	
03752	602015		JMP	ER163
03753	602016		JMP	ER163+1
		/USED FOR SAD*,XCT*		
03754	003715	SGET	GET	
03755	003755	SGETX	SGETX	
03756	003717		GET+2	
03757	203715		LAC	GET
		/USED FOR SAD* 11,XCT* 11,LAC* 10,DAC* 10		
03760	003760	DATA	DATA	
03761	003761		DATA+1	
03762	003762		DATA+2	
03763	000000	STORE	0	
03764	000000		0	
03765	000000		0	
03766	000000		0	
03767	203762		LAC	DATA+2
		/USED FOR JMP* 12,JMS* 12		
03770	003771	JSUB3	JSUB3+1	
03771	602135		JMP	ER174
03772	602135		JMP	ER174
03773	602136		JMP	ER174+1
03774	000000		0	
03775	623774		JMP*	=1
03776	000000		0	
03777	602142		JMP	ER175+1
			.EJECT	

```

/
/DIAGNOSTIC SUBROUTINES
/REPEAT SUBROUTINE = WILL INITIALIZE LOOP COUNTER AND STORE THE
/ADDRESS OF THE FIRST INSTRUCTION OF THE PRESENT TEST IN REPEAT,
/STORE ADDRESS OF NEXT TEST IN NXTADR
/IN CASE OF A CAL AND HALT IN LOC 21, REPEAT WILL CONTAIN THE
/ADDRESS OF THE FIRST INSTRUCTION OF THE TEST THAT CAUSED THE CAL,
/

```

```

04000 000000 REPEAT 0
04001 144206 DZM FPR1 /CLEAR ERROR PRINTED FLAG
04002 204000 LAC REPEAT /SAVE TEST ADDRESS
04003 505702 AND (77777)
04004 345630 TAD (=1
04005 044214 DAC SAVIT
04006 205703 LAC (JMP TYPICAL /INITIALIZE LOC.21 FOR ILLEGAL CAL
04007 040021 DAC 21
04010 204211 LAC COUNT /WAS THIS ENTRANCE LEGAL?
04011 740200 SZA
04012 000000 CAL /NO, BAD TRANSFER OF THE PROGRAM CONTROL
04013 224000 LAC* REPEAT /YES, CONTINUE
04014 044207 DAC NXTADR /GET AND SAVE ADDRESS OF NEXT TEST
04015 444000 ISZ REPEAT
04016 204217 LAC ITIN /PICK UP LOOP INTINATION
04017 740001 CMA /NEGATE AND STORE
04020 044211 DAC COUNT
04021 144212 DZM LOOP /CLEAR LOCK ON ERROR FLAG
04022 624000 JMP* REPEAT

```

```

/DONE SUBROUTINE = TEST LOOP FLAG, IF SET DO NOT UPDATE
/COUNTER BUT LOOP CONTINUOUSLY, IF FLAG IS CLEAR UPDATE COUNTER
/AND TEST FOR COMPLETION, IF COMPLETE CONTINUE ON TO NEXT TEST,
/TEST FOR HALT BIT AC0, IF A 1 HALT AT END OF CURRENT TEST,
/

```

```

04023 000000 DONE 0
04024 204023 LAC =1 /FORM COMPLEMENT OF DONE ADDRESS
04025 505704 AND (177777)
04026 740031 CMA: IAC /TO TRAP BAD ADDRESSING,
04027 344207 TAD NXTADR /IS NXTADR > PRESENT TEST ADDRESS
04030 741100 SPA
04031 000000 CAL /NO, BAD TRANSFER OF THE PROGRAM CONTROL
04032 204212 LAC LOOP
04033 741200 SNA /SKIP IF LOCK ON ERROR FLAG SET
04034 444211 ISZ COUNT /UPDATE COUNTER
04035 624000 JMP* REPEAT /GO THROUGH TEST AGAIN
04036 750004 LAS
04037 751100 SPA: CLA /IS BIT 0 SET?
04040 741000 SKP /YES, HLT WITH TEST ADR IN AC
04041 624023 JMP* DONE /NO, EXIT
04042 204214 LAC SAVIT
04043 740042 HALT
04044 624023 JMP* DONE
.EJECT

```

```

/
/SETUP SUBROUTINE = TEST AC SWITCHES FOR SUPPRESS ERROR TYPEOUT
/AND LOOP ON ERROR, IF SUPPRESS ERROR TYPEOUT BIT SET, DO NOT
/PRINT, IF PROCEED ON ERROR BIT IS NOT SET, SET LOOP FLAG
/AND LOOP TEST CONTINUOUSLY IF ANY ERROR ENCOUNTERED,
/IF PROCEED ON ERROR IS SET, PRINT IF APPLICABLE AND CONTINUE ON TO NEXT
/TEST,
/

```

04045	000000	SETUP	0		
04046	044210		DAC	SAVAC	/SAVE AC AND LINK
04047	750000		CLA		
04050	740010		RAL		
04051	044213		DAC	LINK	
04052	204206		LAC	FPR1	/HAS THIS ERROR BEEN PRINTED?
04053	740200		SZA		
04054	604061		JMP	CK0	/YES, DON'T DO IT
04055	750004		LAS		/TEST FOR BIT 9, SUPPRESS ERROR TYPEOUT
04056	505637		AND	(400	
04057	741200		SNA		
04060	104460		JMS	PRINT	
04061	750004	CK0	LAS		
04062	741100		SPA		/IS BIT 0 SET?
04063	741000		SKP		/YES, HALT WITH TEST ADDRESS IN AC
04064	604067		JMP	,+3	/NO, CONTINUE
04065	204214		LAC	SAVIT	
04066	740040		HALT		/YES, STOP HERE
04067	750004		LAS		/TEST FOR BIT 10, PROCEED ON ERROR
04070	505705		AND	(200	
04071	741200		SNA		/IS PROCEED ON ERROR FLAG SET?
04072	604075		JMP	,+3	
04073	144211		DEM	COUNT	/CLEAR LOOP COUNTER
04074	624207		JMP*	NXTADR	/YES, CONTINUE ON TO NEXT TEST
04075	204212		LAC	LOOP	/HAS LOOP FLAG BEEN SET YET?
04076	741200		SNA		
04077	444212		ISZ	LOOP	/SET LOOP ON ERROR FLAG
04100	204213		LAC	LINK	/RESTORE AC AND LINK
04101	740020		RAR		
04102	204210		LAC	SAVAC	
04103	624000		JMP*	REPEAT	
			.	EJECT	

/TEST FOR EXTRA MEMORY BY SENSING AC SWITCHES BITS 3-5
 /AND SETTING LOC TEF1-TEF7 WITH A SKP IF NON-EXISTANT
 /EACH BLOCK OF MEMORY IS DONE ONE AT A TIME

```

/
74104 000000 SENS Z
04105 205660 LAC (NOP /INITIALIZE ALL FLAGS
04106 044167 DAC TEF1
04107 044170 DAC TEF2
04110 044171 DAC TEF3
04111 044172 DAC TEF4
04112 044173 DAC TEF5
04113 044174 DAC TEF6
04114 044175 DAC TEF7
04115 205706 LAC (SKP
04116 044166 DAC TEF0
04117 204176 LAC CFLD
04120 344200 TAD CBLKX
04121 741200 SNA
04122 624104 JMP SENS /ONLY 4K CONTINUE
04123 544157 SAD F1A
04124 604153 JMP EM8K /8K
04125 544160 SAD F2A
04126 604151 JMP EM12K /12K
04127 544161 SAD F3A
04130 604147 JMP EM16K /16K
04131 544162 SAD F4A
04132 604145 JMP EM20K /20K
04133 544163 SAD F5A
04134 604143 JMP EM24K /24K
04135 544164 SAD F6A
04136 604141 JMP EM28K /28K
04137 205706 EM32K LAC (SKP /32K OR GREATER
04140 044175 DAC TEF7
04141 205706 EM28K LAC (SKP
04142 044174 DAC TEF6
04143 205706 EM24K LAC (SKP
04144 044173 DAC TEF5
04145 205706 EM20K LAC (SKP
04146 044172 DAC TEF4
04147 205706 EM16K LAC (SKP
04150 044171 DAC TEF3
04151 205706 EM12K LAC (SKP
04152 044170 DAC TEF2
04153 205706 EM8K LAC (SKP
04154 044167 DAC TEF1
04155 624104 JMP SENS
      .EJECT

```

/EXTENDED INDEXED MEMORY TEST PARAMETERS

	007000	/	
	000000	YSTLOC=7000	/TEST LOCATION IN MEMORY
04156	000000	F0A	0
04157	010000	F1A	10000
04160	020000	F2A	20000
04161	030000	F3A	30000
04162	040000	F4A	40000
04163	050000	F5A	50000
04164	060000	F6A	60000
04165	070000	F7A	70000

/EXISTENT FIELD INDICATOR STORAGE

	000000	/	
04166	000000	TEF0	0
04167	000000	TEF1	0
04170	000000	TEF2	0
04171	000000	TEF3	0
04172	000000	TEF4	0
04173	000000	TEF5	0
04174	000000	TEF6	0
04175	000000	TEF7	0
04176	000000	CFLD	0
04177	000000	CBLK	0
04200	000000	CBLKX	0
04201	000000	BLOCK	0
04202	000000	BLKNO	0
04203	000000	BLKCON	0
04204	000000	TEMKON	0
04205	000000	FIELD	0
04206	000000	FPR1	0
04207	000000	NXTADR	0
04210	000000	SAVAC	0
04211	000000	COUNT	0
04212	000000	LOOP	0
04213	000000	LINK	0
04214	000000	SAVIT	0
04215	000000	BANKSW	0
04216	000000	POINT1	0
04217	007777	ITIN	7777
04220	000000	TEMP	0
04221	000000	TEMP	0
04222	000000	FTYPE	0
04223	000000	FTITLE	0

,EJECT

/PC BITS 3-5 OF MAX CORE AVAILABLE
 /PC BITS 1+2 OF MAX CORE AVAILABLE
 /PC BITS FOR SENS
 /PC BITS 1+2 OF BLOCK UNDER TEST
 /BLOCK NUMBER FOR PRINTOUT
 /BLOCK CONSTANT FOR TESTING
 /TEMPORARY TEST CONSTANT

/ERROR ALREADY PRINTED FLAG
 /ADDRESS OF NEXT TEST
 /STORE CONTENTS OF AC
 /STORE NUMBER OF TIMES TO DO TEST
 /SCOPE LOOP FLAG, 1 = SCOPE LOOP

/CURRENT TEST ADDRESS
 /BANK MODE SWITCH TEST FLAG
 /TEMP POINTER FOR INDIRECT INDEXING
 /NUMBER OF TIMES EACH TEST IS DONE
 /TEMPORARY STORAGE MESSAGE
 /TEMPORARY STORAGE
 /TEMPORARY STORAGE MESSAGE
 /TITLE FLAG, 1=PRINTED

```

/
/SIXBIT UNPACKING ROUTINE
/*CARRIAGE RETURN, #=LINE FEED; MESSAGE MUST BE TERMINATED
/ BY 0? ENTER WITH JMS TYPE FOLLOWED BY START OF MESSAGE ADDRESS
/

```

04224	000000	TYPE	0		
04225	224224		LAC*	TYPE	/GET AND STORE STRING ADDRESS
04226	044220		DAC	TEMP	
04227	144222		DZM	FTYPE	/CLEAR TYPE FLAG
04230	444224		ISZ	TYPE	/SET FOR GOOD EXIT
04231	224220	TSC1	LAC*	TEMP	/GET DATA WORD
04232	742010		RTL		
04233	742010		RTL		
04234	742010		RTL		
04235	740010		RAL		
04236	104250		JMS	TSC2	/TYPE FIRST CHARACTER
04237	224220		LAC*	TEMP	
04240	742020		RTR		
04241	742020		RTR		
04242	742020		RTR		
04243	104250		JMS	TSC2	/TYPE SECOND CHARACTER
04244	224220		LAC*	TEMP	
04245	104250		JMS	TSC2	/TYPE THIRD CHARACTER
04246	444220		ISZ	TEMP	/UPDATE DATA ADDRESS
04247	604231		JMP	TSC1	
04250	000000	TSC2	0		
04251	505707		AND	(77	/MASK EXCESS BITS
04252	044221		DAC	TEMP	
04253	204222		LAC	FTYPE	/IS SPECIAL CHARACTER FLAG SET?
04254	740200		SZA		
04255	604265		JMP	TYPSP	/YES
04256	204221		LAC	TEMP	/NO, GET CHARACTER
04257	741200		SNA		/IS IT 0?
04260	604263		JMP	,+3	/YES, SPECIAL CHARACTER
04261	104274	TYPAT	JMS	TYOUT	/NO, GO PRINT CHARACTER
04262	624250		JMP*	TSC2	/EXIT TSC2
04263	444222		ISZ	FTYPE	/SET SPECIAL CHARACTER FLAG
04264	624250		JMP*	TSC2	/EXIT TSC2
04265	144222	TYPSP	DZM	FTYPE	/CLEAR SPECIAL CHARACTER FLAG
04266	204221		LAC	TEMP	
04267	741200		SNA		/IS IT 0?
04270	624250		JMP*	TSC2	/YES IGNORE IT, EXIT TSC2
04271	545707		SAD	(77	/IS IT 77?
04272	624224		JMP*	TYPE	/YES EXIT TYPE SUBROUTINE
04273	604261		JMP	TYPAT	/NO

```

,EJECT

```

```

04274 000000 TTOUT 0
04275 545710 SAO (45 /IS IT 45?
04276 604307 JMP PRINTB /YES, TYPE IT
04277 545711 SAO (43 /IS IT 43?
04300 604311 JMP PRINTC /YES, TYPE IT
04301 345712 TAD (=40 /NO
04302 741100 SPA
04303 345641 TAD (100
04304 345713 TAD (240

/
PRINTA JMS OTY /TYPE CHARACTER
04306 624274 JMP* TTOUT
04307 760215 PRINTB LAW 215
04310 604305 JMP PRINTA /TYPE CARRIAGE RETURN
04311 760212 PRINTC LAW 212
04312 604305 JMP PRINTA /TYPE LINE FEED

/
/INITN=INITIALIZE LOOP COUNTER FOR NORMAL ITERATIONS
/
INITN 0
04313 000000 INITN 0
04314 144211 DEM COUNT
04315 205714 LAC (7777
04316 044217 DAC ITIN
04317 624313 JMP* INITN

/
/INIT10=INITIALIZE LOOP COUNTER FOR 1 LOOPS
/
INIT10 0
04320 000000 INIT10 0
04321 144211 DEM COUNT
04322 205645 LAC (1
04323 044217 DAC ITIN
04324 624320 JMP* INIT10

/
/RANDOM NUMBER GENERATOR=EXIT WITH NUMBER IN AC AND RANB
/
RANDOM 0
04325 000000 RANDOM 0
04326 750000 CLA
04327 344340 TAD RANA
04330 344341 TAD RANB
04331 044340 DAC RANA
04332 750000 CLA
04333 740010 RAL
04334 344340 TAD RANA
04335 344341 TAD RANB
04336 044341 DAC RANB
04337 624325 JMP* RANDOM
04340 731420 RANA
04341 015612 RANB
.EJECT

```

/
/ZERO UNUSED PORTIONS OF FIELD 0
/

04342	000000	ZERF2	Z		
04343	144211		DZM	COUNT	/INITIALIZE TEST AND LOOP COUNTER
04344	144363		DZM	POINT	/ZERO LOCATIONS 0-177
04345	164363		DZM*	POINT	/ZERO LOCATION
04346	444363		ISZ	POINT	/UPDATE POINTER
04347	204363		LAC	POINT	/GET POINT
04350	545705		SAD	(200	/HAS LOC. 177 BEEN ZEROED YET?
04351	741000		SKP		/YES, CONTINUE
04352	604345		JMP	ZERF0*3	/NO, LOOP
04353	205627		LAC	SIZE	/ZERO FROM TOP OF PROGRAM TO 7777
04354	044363		DAC	POINT	/SAVE IT
04355	444363		ISZ	POINT	/UPDATE POINTER
04356	164363		DZM*	POINT	/ZERO LOCATION
04357	204363		LAC	POINT	/GET POINTER
04360	545714		SAD	(7777	/HAS LOC. 7777 BEEN ZEROED YET?
04361	624342		JMP*	ZERF0	/YES, EXIT
04362	604355		JMP	,=5	/NO, LOOP
04363	000000		POINT	0	/POINTER FOR CLEARING OPERATION

/

/ZERO ALL MEMORY EXCEPT FOR UPDATE REPEAT SO THAT EXTENDED
/MEMORY IS ONLY ZEROED ONCE EACH EXTENDED MEMORY TEST AND THAT TEST
/PROGRAM IS ONLY MOVED ONCE

04364	000000	ZERMEM	0		
04365	444000		ISZ	REPEAT	/UPDATE REPEAT
04366	444000		ISZ	REPEAT	
04367	444000		ISZ	REPEAT	
04370	205627		LAC	SIZE	
04371	040015		DAC	15	
04372	750004		LAS		
04373	505715		AND	(370000	/GET FIELD BITS
04374	345714		TAD	(7777	
04375	044363		DAC	POINT	/FORM LAST ADDRESS
04376	160015		DZM*	15	/ZERO LOCATION
04377	200015		LAC	15	/WAS IT THE LAST ONE?
04400	544363		SAD	POINT	
04401	624364		JMP*	ZERMEM	/YES, EXIT
04402	604376		JMP	,=4	/NO, LOOP
			,EJECT		

68

```

/
/TCORE INITIALIZE CORE CONSTANTS AND TEST SWITCHES FOR AVAILABLE CORE
/

```

```

04403 000000 TCORE 0
04404 144200 DZM CBLKX
04405 144177 DZM CBLK
04406 144201 DZM BLOCK
04407 144202 DZM BLKNO
04410 144203 DZM BLKCON
04411 144176 DZM CFLD
04412 750004 LAS
04413 505716 AND (70000 /PC BITS 3-5 IN CFLD
04414 044176 DAC CFLD
04415 750004 LAS
04416 505701 AND (300000 /PC BITS 1-2 IN CBLK
04417 044177 DAC CBLK
04420 044200 DAC CBLKX
04421 624403 JMP* TCORE

```

```

/
/OCTAL PRINT - NO ZERO SUPPRESSION
/

```

```

04422 000000 OCTP 0
04423 044456 DAC XOCT1 /SAVE QUANTITY
04424 742030 SWHA /SWAP HALVES OF AC
04425 044457 DAC XOCT2 /SAVE SWAPPED AC
04426 104436 JMS OCTP3 /PRINT HI ORDER THREE
04427 204456 LAC XOCT1 /GET UNSWAPPED QUANTITY
04430 044457 DAC XOCT2 /SAVE UNSWAPPED AC
04431 104436 JMS OCTP3 /PRINT LO ORDER THREE
04432 760240 LAW 240 /SPACE
04433 104575 JMS OTY /PRINT IT
04434 104575 JMS OTY
04435 624422 JMP* OCTP /EXIT
04436 000000 OCTP3 0
04437 742020 RTR /SHIFT 6 RIGHT
04440 742020 RTR
04441 742020 RTR
04442 104450 JMS OCTSUB /PRINT 1ST CHARACTER
04443 740020 RAR /SHIFT 4 RIGHT
04444 742020 RTR
04445 104450 JMS OCTSUB /PRINT 2ND CHARACTER
04446 104450 JMS OCTSUB /PRINT 3RD CHARACTER
04447 624436 JMP* OCTP3
04450 000000 OCTSUB 0
04451 505717 AND (7
04452 345720 TAD (260
04453 104575 JMS OTY
04454 204457 LAC XOCT2
04455 624450 JMP* OCTSUB
04456 000000 XOCT1 0
04457 000000 XOCT2 0
.EJECT

```

/PRINT ERROR NUMBER, CONTENTS OF AC, LINK, XR, LR

/PRINT

04460	000000			
04461	204206	LAC	FPR1	
04462	741200	SNA		/HAS ERROR PRINTED ONCE, FLAG SET?
04463	444206	ISZ	FPR1	/NO, SET IT
04464	104603	JMS	CRLF	
04465	760305	LAW	305	/E
04466	104575	JMS	OTY	
04467	760322	LAW	322	/R
04470	104575	JMS	OTY	
04471	224045	LAC*	SETUP	/GET ERROR NUMBER
04472	505635	AND	(777	/MASK OUT LAW
04473	104422	JMS	OCTP	/PRINT OCTAL EQUIVALENT OF AC
04474	760320	LAW	320	/P
04475	104575	JMS	OTY	
04476	760301	LAW	301	/A
04477	104575	JMS	OTY	
04500	760323	LAW	323	/S,S
04501	104575	JMS	OTY	
04502	104575	JMS	OTY	
04503	760275	LAW	275	/#
04504	104575	JMS	OTY	
04505	204211	LAC	COUNT	/GET LOOP COUNTER AND CALCULATE
04506	740030	IAC		/NUMBER OF PASSES COMPLETED
04507	344217	TAD	ITIN	/AND PRINT IT.
04510	740030	IAC		
04511	104422	JMS	OCTP	
04512	760301	LAW	301	/A
04513	104575	JMS	OTY	
04514	760303	LAW	303	/C
04515	104575	JMS	OTY	
04516	760275	LAW	275	/#
04517	104575	JMS	OTY	
04520	204210	LAC	SAVAC	/GET CONTENTS OF AC
04521	104422	JMS	OCTP	
04522	760314	LAW	314	/L
04523	104575	JMS	OTY	
04524	760275	LAW	275	/#
04525	104575	JMS	OTY	
04526	204213	LAC	LINK	/GET CONTENTS OF LINK
04527	740200	SEA		
04530	604534	JMP	,+4	
04531	760260	LAW	260	/PRINT 0
04532	104575	JMS	OTY	
04533	604536	JMP	,+3	
04534	760261	LAW	261	/PRINT 1
04535	104575	JMS	OTY	
04536	760240	LAW	240	/SPACE,SPACE
04537	104575	JMS	OTY	
04541	104575	JMS	OTY	

.EJECT

04541	760330	LAW	330	/X
04542	104575	JMS	OTY	
04543	760322	LAW	322	/R
04544	104575	JMS	OTY	
04545	760275	LAW	275	/=
04546	104575	JMS	OTY	
04547	724000	PXA		/GET CONTENTS OF XR
04550	104422	JMS	OCTP	
04551	760314	LAW	314	/L
04552	104575	JMS	OTY	
04553	760322	LAW	322	/R
04554	104575	JMS	OTY	
04555	760275	LAW	275	/=
04556	104575	JMS	OTY	
04557	730000	PLA		/GET CONTENTS OF LR
04560	104422	JMS	OCTP	
04561	760302	LAW	302	/B
04562	104575	JMS	OTY	
04563	760314	LAW	314	/L
04564	104575	JMS	OTY	
04565	760313	LAW	313	/K
04566	104575	JMS	OTY	
04567	760275	LAW	275	/=
04570	104575	JMS	OTY	
04571	204202	LAC	BLKNO	/PRINT CORE BLOCK
04572	345720	TAD	(260	
04573	104575	JMS	OTY	
04574	624460	JMP*	PRINT	
		/OUTPUT CHARACTER ON TELETYPE		
		/		
04575	000000	OTY	0	
04576	700406		TLS	
04577	700401		TSP	
04600	604577	JMP	=1	
04601	700402		TCF	
04602	624575	JMP*	OTY	
04603	000000	CRLF	0	/PRINT CARRIAGE RETURN + LINE FEED
04604	760215	LAW	215	
04605	104575	JMS	OTY	
04606	760212	LAW	212	
04607	104575	JMS	OTY	
04610	624603	JMP*	CRLF	
		,EJECT		

/TYPE WHEN CAL OCCURS=TYPE WHERE CALLED FROM AND ADDRESS OF PRESENT TEST
 /AND CONTENTS OF AC, IF NO PRINTOUT DESIRED
 /HALT WITH LAW 777 IN AC

24611	744218	TYPICAL	DAC	SAVAC	/SAVE AC
24612	750024		LAS		/TEST FOR BIT 9, SUPPRESS ERROR TYPROUT
24613	505637		AND	(400	
24614	741200		SNA		/IS BIT 9 SET?
24615	604623		JMP	,+6	/NO, CONTINUE WITH PRINTOUT
24616	760777		LAW	777	/YES, HALT WITH LAW 777 IN AC
24617	740040		HALT		
24620	204218		LAC	SAVAC	/DISPLAY CONTENTS OF AC WHEN CONTINUE PRESSED
24621	740040		HALT		
24622	604621		JMP	,=1	
24623	104603		JMS	CRLF	/PRINT CR & LF
24624	760303		LAW	303	/C
24625	104575		JMS	0TY	
24626	760301		LAW	301	/A
24627	104575		JMS	0TY	
24630	760314		LAW	314	/L
24631	104575		JMS	0TY	
24632	760240		LAW	240	/SPACE
24633	104575		JMS	0TY	
24634	760301		LAW	301	/A
24635	104575		JMS	0TY	
24636	760304		LAW	304	/D
24637	104575		JMS	0TY	
24640	760322		LAW	322	/R
24641	104575		JMS	0TY	
24642	760275		LAW	275	/B
24643	104575		JMS	0TY	
24644	200020		LAC	20	/PRINT CONTENTS OF LCC20(=1)
24645	740200		SEA		
24646	345630		TAD	(=1	
24647	505702		AND	(77777	/MASK NON ADDRESS BITS
24650	104422		JMS	OCTP	
24651	760324		LAW	324	/T
24652	104575		JMS	0TY	
24653	760305		LAW	305	/E
24654	104575		JMS	0TY	
24655	760323		LAW	323	/S
24656	104575		JMS	0TY	
24657	760324		LAW	324	/T
24660	104575		JMS	0TY	
24661	760240		LAW	240	/SPACE
24662	104575		JMS	0TY	
24663	760301		LAW	301	/A
24664	104575		JMS	0TY	
24665	760324		LAW	304	/D
24666	104575		JMS	0TY	

,EJECT

2

24667	760322	LAW	322	/R
24670	104575	JMS	0TY	
24671	760275	LAW	275	/S
24672	104575	JMS	0TY	
24673	204214	LAC	SAVIT	
24674	104422	JMS	0CTP	
24675	760301	LAW	301	/A
24676	104575	JMS	0TY	
24677	760303	LAW	303	/C
24727	104575	JMS	0TY	
24701	760275	LAW	275	/S
24702	104575	JMS	0TY	
24703	204210	LAC	SAVAC	/GET CONTENTS OF AC
24704	104422	JMS	0CTP	/PRINT CONTENT OF AC
24705	760330	LAW	330	/X
24706	104575	JMS	0TY	
24707	760322	LAW	322	/R
24710	104575	JMS	0TY	
24711	760275	LAW	275	/S
24712	104575	JMS	0TY	
24713	724000	PXA		
24714	104422	JMS	0CTP	
24715	760302	LAW	302	/B
24716	104575	JMS	0TY	
24717	760314	LAW	314	/L
24720	104575	JMS	0TY	
24721	760313	LAW	313	/K
24722	104575	JMS	0TY	
24723	760275	LAW	275	/S
24724	104575	JMS	0TY	
24725	204202	LAC	BLKNC	
24726	345720	TAD	(260	
24727	104575	JMS	0TY	
24730	740040	HALT		
24731	104342	JMS	ZERFP	
24732	104364	JMS	ZERMEM	
24733	624214	JMP*	SAVIT	/TRY TEST AGAIN WITH INITIALIZATION
		.EJECT		

/
 /DIAGNOSTIC ERROR HALT SECTION/
 /WHEN AN ERROR IS ENCOUNTERED A JUMP TO THIS SECTION
 /WILL BE EXECUTED AND A HALT WILL OCCUR WITH A LAW XX
 /IN THE AC THE XX REFERRING TO THE ERROR NUMBER. PRESSING
 /CONTINUE WILL RETRIEVE THE CONTENTS OF THE AC AT THE TIME
 /OF ERROR. PRESSING CONTINUE AGAIN WILL CYCLE THE
 /TEST THAT FAILED PERMANENTLY. THE PROGRAM CANNOT BE
 /CONTINUED EXCEPT BY RESTARTING AT LOC 200 OR AT SOME
 /OTHER LOCATION THAT STARTS THE NEXT TEST.

/
 /ERROR 1-11 ARE ASSOCIATED WITH TEST 1.0 9/15 MODE SWITCH

04734	104045	H1	JMS	SETUP
04735	760001		LAW	1
04736	104045	H2	JMS	SETUP
04737	760002		LAW	2
04740	104045	H3	JMS	SETUP
04741	760003		LAW	3
04742	104045	H4	JMS	SETUP
04743	760004		LAW	4
04744	104045	H5	JMS	SETUP
04745	760005		LAW	5
04746	104045	H6	JMS	SETUP
04747	760006		LAW	6
04750	104045	H7	JMS	SETUP
04751	760007		LAW	7

/ERRORS 13-34 ARE ASSOCIATED WITH TEST 2 DATA PATHS TRANSFERS

04752	104045	H13	JMS	SETUP
04753	760013		LAW	13
04754	104045	H14	JMS	SETUP
04755	760014		LAW	14
04756	104045	H15	JMS	SETUP
04757	760015		LAW	15
04760	104045	H20	JMS	SETUP
04761	760020		LAW	20
04762	104045	H21	JMS	SETUP
04763	760021		LAW	21
04764	104045	H22	JMS	SETUP
04765	760022		LAW	22
04766	104045	H23	JMS	SETUP
04767	760023		LAW	23
04770	104045	H24	JMS	SETUP
04771	760024		LAW	24
04772	104045	H25	JMS	SETUP
04773	760025		LAW	25
04774	104045	H30	JMS	SETUP
04775	760030		LAW	30
04776	104045	H31	JMS	SETUP
04777	760031		LAW	31
05000	104045	H32	JMS	SETUP
05001	760032		LAW	32
05002	104045	H33	JMS	SETUP
05003	760033		LAW	33
05004	104045	H34	JMS	SETUP

25205	760034	LAW	34
		/ERRORS 35 - 42 ARE ASSOCIATED WITH TEST 3 CLEAR INSTRUCTIONS	
		/	
25206	104045	H39	JMS SETUP
25207	760035	LAW	35
25210	104045	H40	JMS SETUP
25211	760040	LAW	40
25212	104045	H41	JMS SETUP
25213	760041	LAW	41
25214	104045	H42	JMS SETUP
25215	760042	LAW	42
		/ERRORS 43 - 52 ARE ASSOCIATED WITH TEST 4 DATA PATH INTERACTION	
		/	
25216	104045	H43	JMS SETUP
25217	760043	LAW	43
25220	104045	H44	JMS SETUP
25221	760044	LAW	44
25222	104045	H45	JMS SETUP
25223	760045	LAW	45
25224	104045	H52	JMS SETUP
25225	760050	LAW	50
25226	104045	H51	JMS SETUP
25227	760051	LAW	51
25230	104045	H52	JMS SETUP
25231	760052	LAW	52
		/ERRORS 53 - 61 ARE ASSOCIATED WITH TEST 5 CLEAR INTERACTION	
		/	
25232	104045	H53	JMS SETUP
25233	760053	LAW	53
25234	104045	H54	JMS SETUP
25235	760054	LAW	54
25236	104045	H55	JMS SETUP
25237	760055	LAW	55
25240	104045	H56	JMS SETUP
25241	760056	LAW	56
25242	104045	H60	JMS SETUP
25243	760060	LAW	60
25244	104045	H61	JMS SETUP
25245	760061	LAW	61
		/ERRORS 62-63 ARE ASSOCIATED WITH BIT PICKUP TEST	
		/	
25246	104045	H62	JMS SETUP
25247	760062	LAW	62
25250	104045	H63	JMS SETUP
25251	760063	LAW	63
		/ERRORS 62 - 104 ARE ASSOCIATED WITH TEST 6 ADD TO AC AND XR	
		/	
25252	104045	H64	JMS SETUP
25253	760064	LAW	64
25254	104045	H65	JMS SETUP
25255	760065	LAW	65
25256	104045	H66	JMS SETUP
25257	760066	LAW	66
25260	104045	H67	JMS SETUP
25261	760067	LAW	67

05062	104045	H70	JMS	SETUP
05063	760070		LAW	70
05064	104045	H71	JMS	SETUP
05065	760071		LAW	71
05066	104045	H72	JMS	SETUP
05067	760072		LAW	72
05070	104045	H73	JMS	SETUP
05071	760073		LAW	73
05072	104045	H74	JMS	SETUP
05073	760074		LAW	74
05074	104045	H75	JMS	SETUP
05075	760075		LAW	75
05076	104045	H76	JMS	SETUP
05077	760076		LAW	76
05100	104045	H77	JMS	SETUP
05101	760077		LAW	77
05102	104045	H100	JMS	SETUP
05103	760100		LAW	100
05104	104045	H101	JMS	SETUP
05105	760101		LAW	101
05106	104045	H102	JMS	SETUP
05107	760102		LAW	102
05110	104045	H103	JMS	SETUP
05111	760103		LAW	103
05112	104045	H104	JMS	SETUP
05113	760104		LAW	104

/ERRORS 110 = 116 ARE ASSOCIATED WITH TEST 7 ADD TO XR AND SKIP

05114	104045	H110	JMS	SETUP
05115	760110		LAW	110
05116	104045	H111	JMS	SETUP
05117	760111		LAW	111
05120	104045	H112	JMS	SETUP
05121	760112		LAW	112
05122	104045	H113	JMS	SETUP
05123	760113		LAW	113
05124	104045	H114	JMS	SETUP
05125	760114		LAW	114
05126	104045	H115	JMS	SETUP
05127	760115		LAW	115
05130	104045	H116	JMS	SETUP
05131	760116		LAW	116

/ERRORS 117 = 133 ARE ASSOCIATED WITH TEST 8 INSTRUCTION EXERCISER

05132	104045	H117	JMS	SETUP
05133	760117		LAW	117
05134	104045	H120	JMS	SETUP
05135	760120		LAW	120
05136	104045	H121	JMS	SETUP
05137	760121		LAW	121
05140	104045	H122	JMS	SETUP
05141	760122		LAW	122
05142	104045	H124	JMS	SETUP
05143	760124		LAW	124
05144	104045	H125	JMS	SETUP

05145	760125		LAW	125
05146	104045	H126	JMS	SETUP
05147	760126		LAW	126
05150	104045	H130	JMS	SETUP
05151	760130		LAW	130
05152	104045	H131	JMS	SETUP
05153	760131		LAW	131
05154	104045	H132	JMS	SETUP
05155	760132		LAW	132
05156	104045	H133	JMS	SETUP
05157	760133		LAW	133

/ERRORS 134-155 ARE ASSOCIATED WITH TEST 9 DIRECT INDEXED ADDRESSING

05160	104045	H134	JMS	SETUP
05161	760134		LAW	134
05162	104045	H135	JMS	SETUP
05163	760135		LAW	135
05164	104045	H136	JMS	SETUP
05165	760136		LAW	136
05166	104045	H137	JMS	SETUP
05167	760137		LAW	137
05170	104045	H140	JMS	SETUP
05171	760140		LAW	140
05172	104045	H141	JMS	SETUP
05173	760141		LAW	141
05174	104045	H142	JMS	SETUP
05175	760142		LAW	142
05176	104045	H143	JMS	SETUP
05177	760143		LAW	143
05200	104045	H144	JMS	SETUP
05201	760144		LAW	144
05202	104045	H145	JMS	SETUP
05203	760145		LAW	145
05204	104045	H146	JMS	SETUP
05205	760146		LAW	146
05206	104045	H147	JMS	SETUP
05207	760147		LAW	147
05210	104045	H150	JMS	SETUP
05211	760150		LAW	150
05212	104045	H151	JMS	SETUP
05213	760151		LAW	151
05214	104045	H152	JMS	SETUP
05215	760152		LAW	152
05216	104045	H153	JMS	SETUP
05217	760153		LAW	153
05220	104045	H154	JMS	SETUP
05221	760154		LAW	154
05222	104045	H155	JMS	SETUP
05223	760155		LAW	155

/ERRORS 160-165 ARE ASSOCIATED WITH TEST 10 INDIRECT INDEXED ADDRESSING

05224	104045	H160	JMS	SETUP
05225	760160		LAW	160
05226	104045	H161	JMS	SETUP
05227	760161		LAW	161

05230	124045	H162	JMS	SETUP
05231	760162		LAW	162
05232	104045	H163	JMS	SETUP
05233	760163		LAW	163
05234	104045	H164	JMS	SETUP
05235	760164		LAW	164
05236	104045	H165	JMS	SETUP
05237	760165		LAW	165
05240	104045	H166	JMS	SETUP
05241	760166		LAW	166

/ERRORS 170-175 ARE ASSOCIATED WITH TEST 11-INDIRECT
/INDEXED AUTO INCREMENTED ADDRESS

05242	104045	H170	JMS	SETUP
05243	760170		LAW	170
05244	104045	H171	JMS	SETUP
05245	760171		LAW	171
05246	104045	H172	JMS	SETUP
05247	760172		LAW	172
05250	104045	H173	JMS	SETUP
05251	760173		LAW	173
05252	104045	H174	JMS	SETUP
05253	760174		LAW	174
05254	104045	H175	JMS	SETUP
05255	760175		LAW	175
05256	104045	H176	JMS	SETUP
05257	760176		LAW	176

/ERRORS 20-232 ARE ASSOCIATED WITH TEST 12-REFERENCE INSTRUCTIONS
/INDEXED TO OTHER FIELDS

05260	104045	H200	JMS	SETUP
05261	760200		LAW	200
05262	104045	H201	JMS	SETUP
05263	760201		LAW	201
05264	104045	H202	JMS	SETUP
05265	760202		LAW	202
05266	104045	H203	JMS	SETUP
05267	760203		LAW	203
05270	104045	H204	JMS	SETUP
05271	760204		LAW	204
05272	104045	H205	JMS	SETUP
05273	760205		LAW	205
05274	104045	H206	JMS	SETUP
05275	760206		LAW	206
05276	104045	H207	JMS	SETUP
05277	760207		LAW	207
05300	104045	H210	JMS	SETUP
05301	760210		LAW	210
05302	104045	H211	JMS	SETUP
05303	760211		LAW	211
05304	104045	H212	JMS	SETUP
05305	760212		LAW	212
05306	104045	H213	JMS	SETUP
05307	760213		LAW	213
05310	104045	H214	JMS	SETUP

25311	760214		LAW	214
25312	104045	H215	JMS	SETUP
25313	760215		LAW	215
25314	104045	H216	JMS	SETUP
25315	760216		LAW	216
25316	104045	H217	JMS	SETUP
25317	760217		LAW	217
25320	104045	H220	JMS	SETUP
25321	760220		LAW	220
25322	104045	H221	JMS	SETUP
25323	760221		LAW	221
25324	104045	H222	JMS	SETUP
25325	760222		LAW	222
25326	104045	H223	JMS	SETUP
25327	760223		LAW	223
25330	104045	H224	JMS	SETUP
25331	760224		LAW	224
25332	104045	H225	JMS	SETUP
25333	760225		LAW	225
25334	104045	H226	JMS	SETUP
25335	760226		LAW	226
25336	104045	H227	JMS	SETUP
25337	760227		LAW	227
25340	104045	H230	JMS	SETUP
25341	760230		LAW	230
25342	104045	H231	JMS	SETUP
25343	760231		LAW	231
25344	104045	H232	JMS	SETUP
25345	760232		LAW	232

/ERROR 233-343 ARE ASSOCIATED WITH TEST 13-MOVING TEST
/PROGRAM TO UPPER CORE AND TRANSFER CONTROL TO TEST

25346	104045	H233	JMS	SETUP
25347	760233		LAW	233
25350	104045	H234	JMS	SETUP
25351	760234		LAW	234
25352	104045	H235	JMS	SETUP
25353	760235		LAW	235
25354	104045	H236	JMS	SETUP
25355	760236		LAW	236
25356	104045	H237	JMS	SETUP
25357	760237		LAW	237
25360	104045	H240	JMS	SETUP
25361	760240		LAW	240
25362	104045	H241	JMS	SETUP
25363	760241		LAW	241
25364	104045	H242	JMS	SETUP
25365	760242		LAW	242
25366	104045	H243	JMS	SETUP
25367	760243		LAW	243
25370	104045	H244	JMS	SETUP
25371	760244		LAW	244
25372	104045	H245	JMS	SETUP
25373	760245		LAW	245
25374	104045	H246	JMS	SETUP

05375	760246		LAW	246
05376	104045	H250	JMS	SETUP
05377	760250		LAW	250
05400	104045	H251	JMS	SETUP
05401	760251		LAW	251
05402	104045	H252	JMS	SETUP
05403	760252		LAW	252
05404	104045	H253	JMS	SETUP
05405	760253		LAW	253
05406	104045	H254	JMS	SETUP
05407	760254		LAW	254
05410	104045	H255	JMS	SETUP
05411	760255		LAW	255
05412	104045	H260	JMS	SETUP
05413	760260		LAW	260
05414	104045	H261	JMS	SETUP
05415	760261		LAW	261
05416	104045	H262	JMS	SETUP
05417	760262		LAW	262
05420	104045	H263	JMS	SETUP
05421	760263		LAW	263
05422	104045	H264	JMS	SETUP
05423	760264		LAW	264
05424	104045	H265	JMS	SETUP
05425	760265		LAW	265
05426	104045	H270	JMS	SETUP
05427	760270		LAW	270
05430	104045	H271	JMS	SETUP
05431	760271		LAW	271
05432	104045	H272	JMS	SETUP
05433	760272		LAW	272
05434	104045	H273	JMS	SETUP
05435	760273		LAW	273
05436	104045	H274	JMS	SETUP
05437	760274		LAW	274
05440	104045	H275	JMS	SETUP
05441	760275		LAW	275
05442	104045	H300	JMS	SETUP
05443	760300		LAW	300
05444	104045	H301	JMS	SETUP
05445	760301		LAW	301
05446	104045	H302	JMS	SETUP
05447	760302		LAW	302
05450	104045	H303	JMS	SETUP
05451	760303		LAW	303
05452	104045	H304	JMS	SETUP
05453	760304		LAW	304
05454	104045	H305	JMS	SETUP
05455	760305		LAW	305
05456	104045	H310	JMS	SETUP
05457	760310		LAW	310
05460	104045	H311	JMS	SETUP
05461	760311		LAW	311
05462	104045	H312	JMS	SETUP
05463	760312		LAW	312

05464	104045	H313	JMS	SETUP
05465	760313		LAW	313
05466	104045	H314	JMS	SETUP
05467	760314		LAW	314
05470	104045	H315	JMS	SETUP
05471	760315		LAW	315
05472	104045	H320	JMS	SETUP
05473	760320		LAW	320
05474	104045	H321	JMS	SETUP
05475	760321		LAW	321
05476	104045	H322	JMS	SETUP
05477	760322		LAW	322
05500	104045	H323	JMS	SETUP
05501	760323		LAW	323
05502	104045	H324	JMS	SETUP
05503	760324		LAW	324
05504	104045	H325	JMS	SETUP
05505	760325		LAW	325
05506	104045	H330	JMS	SETUP
05507	760330		LAW	330
05510	104045	H331	JMS	SETUP
05511	760331		LAW	331
05512	104045	H332	JMS	SETUP
05513	760332		LAW	332
05514	104045	H333	JMS	SETUP
05515	760333		LAW	333
05516	104045	H334	JMS	SETUP
05517	760334		LAW	334
05520	104045	H335	JMS	SETUP
05521	760335		LAW	335
05522	104045	H340	JMS	SETUP
05523	760340		LAW	340
05524	104045	H341	JMS	SETUP
05525	760341		LAW	341
05526	104045	H342	JMS	SETUP
05527	760342		LAW	342
05530	104045	H343	JMS	SETUP
05531	760343		LAW	343
05532	104045	H344	JMS	SETUP
05533	760344		LAW	344

/
/MESSAGES

05534 454305 /
05535 160440 /
05536 170640 /
05537 200123 /
05540 230077 /
05541 454321 /
05542 251103 /
05543 134026 /
05544 052211 /
05545 763140 /
05546 200123 /
05547 237240 /

^END ,SIXBT %END OF PASS0?'

^GVF ,SIXBT %QUICK VERIFY PASS: FOR THOROUGH TESTING'

21.

05557 061722
 05551 402410
 05552 172217
 05553 250712
 05554 402405
 05555 232411
 05556 160700
 05557 402325
 05560 244001
 05561 032340
 05562 704024
 05563 174001
 05564 406000
 05565 770000
 05566 454311
 05567 160405
 05570 304022
 05571 050711
 05572 232405
 05573 224024
 05574 052324
 05575 551501
 05576 111604
 05577 050355
 05600 616555
 05601 046001
 05602 030077
 05603 454323
 05604 052440
 05605 020116
 05606 134015
 05607 170405
 05610 402327
 05611 112403
 05612 104024
 05613 174001
 05614 406054
 05615 400000
 05616 202205
 05617 232340
 05620 220523
 05621 252440
 05622 241005
 05623 164023
 05624 240122
 05625 244543
 05626 007727
 05627 005721
 05630 000000
 05630 777777
 05631 525252
 05632 252525
 05633 000377
 05634 777377
 05635 000777

.SIXBT ' SET ACS 8 TO A 00?'

MTITLE .SIXBT 'X#INDEX REGISTER TEST=MAINDEC=15=00AC0?'

MCONV .SIXBT 'X#SET BANK MODE SWITCH TO A 0. '

.SIXBT 'PRESS RESET THEN STARTX#0?'

SIZE .SIZE /PRINTER TO END OF STORAGE
.END

*L
 *L
 *L
 *L
 *L
 *L

05636	737000	*L
05637	000400	*L
05640	777000	*L
05641	000100	*L
05642	000012	*L
05643	000002	*L
05644	777776	*L
05645	000001	*L
05646	000770	*L
05647	620020	*L
05650	440021	*L
05651	620021	*L
05652	000003	*L
05653	003741	*L
05654	602011	*L
05655	000005	*L
05656	100000	*L
05657	003763	*L
05660	740000	*L
05661	007000	*L
05662	760000	*L
05663	000010	*L
05664	700000	*L
05665	001000	*L
05666	000027	*L
05667	003546	*L
05670	000036	*L
05671	003576	*L
05672	000026	*L
05673	003635	*L
05674	575634	*L
05675	000011	*L
05676	202143	*L
05677	000014	*L
05700	630055	*L
05701	300000	*L
05702	077777	*L
05703	604611	*L
05704	177777	*L
05705	000200	*L
05706	741000	*L
05707	000077	*L
05710	000045	*L
05711	000043	*L
05712	777740	*L
05713	000240	*L
05714	007777	*L
05715	370000	*L
05716	070000	*L
05717	000007	*L
05720	000260	*L

SIZE=05721

NO ERROR LINES

83