

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DJEXB-A-D
PRODUCT NAME: 2K TO 32K PDP-8A PROCESSOR EXERCISER
DATE CREATED: JANUARY 15, 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BRUCE HANSEN

COPYRIGHT 1974
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS, 01754

"THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT."

TABLE OF CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	HARDWARE
2.2	STORAGE
2.3	PREREQUISITE SOFTWARE
3.	RESTRICTIONS
3.1	HARDWARE RESTRICTIONS
3.2	SOFTWARE RESTRICTIONS
4.	STANDARD TEST PROCEDURE
4.1	CHANGING DEVICE IOT CODES
4.2	RUN PDP-8A PROCESSOR EXERCISER
5.	ERRORS
5.1	ERROR HALTS
5.2	ERROR PRINTOUTS
5.3	MEMORY REFERENCE INSTRUCTION ERRORS
5.4	MEMORY REFERENCE TEST INSTRUCTION SETUP
5.5	OPERATE INSTRUCTION ERRORS
5.6	OPERATE TEST INSTRUCTION SETUP
5.7	SERIAL LINE UNIT DATA ERRORS
5.8	12 BIT PARALLEL I/O DATA ERRORS
5.9	ILLEGAL INTERRUPT ERRORS
5.10	INACTIVE DEVICE ERRORS
5.11	BATTERY EMPTY ERRORS
5.12	RELOCATION ERRORS
6.	SWITCH REGISTER SETTINGS
6.1	NORMAL OPERATING SWITCHES
6.2	ERROR SWITCHES
7.	REVISIONS
8.	PROGRAM DESCRIPTION
9.	FLOWCHARTS
10.	LISTING

1.0

ABSTRACT

THE 2K TO 32K PDP-8A PROCESSOR EXERCISER IS A PROGRAM TO CHECK THE EXECUTION OF MEMORY REFERENCE INSTRUCTIONS AND OPERATE INSTRUCTIONS, IN A 2K TO 32K PDP-8A COMPUTER. THE PROGRAM ALSO CHECKS FOR DATA TRANSMISSIONS, AND INTERRUPTS IF A OPTIONS 1 MODULE (M8316) IS CONNECTED IN LOOP BACK MODE AND THE PROGRAM IS INITIALIZED TO TEST THE OPTION 1 MODULE. ALL INSTRUCTIONS, ADDRESSES, MEMORY DATA, AC DATA, MQ DATA AND LINK DATA ARE GENERATED FROM A RANDOM NUMBER GENERATOR.

THE PROGRAM RELOCATES ITSELF A PAGE AT A TIME, UP AND DOWN, WITHIN ANY 2K TO 4K MEMORY FIELD, IF THE COMPUTER CONTAINS MORE THAN 4K OF MEMORY, THE PROGRAM WILL RELOCATE UP AND DOWN BETWEEN MEMORY FIELDS, AT LEAST 2K OF MEMORY IS REQUIRED IN THE LAST EXTENDED MEMORY FIELD FOR THE PROGRAM TO RELOCATE INTO IT.

THE PROGRAM CAN ALSO BE RUN ON THE PDP-8A APT TEST LINE.

2.0

REQUIREMENTS

2.1

HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED FOR THE EXECUTION OF THIS PROGRAM,

PROCESSOR(S): PDP-8A/E/F/M

MEMORY: 2K OF MEMORY TO 32K OF MEMORY IN ANY INCREMENTS OF 1K.

OPTIONS: IF GREATER THAN 4K OF MEMORY, A M837 OR M8317 MODULE (MEMORY EXTENSION/TIME SHARE CONTROL) IS REQUIRED.

IF DATA TRANSMISSIONS AND INTERRUPTS ARE TO BE TESTED, A PDP-8A OPTION BOARD #1 (M8316) IS REQUIRED (PDP-8A ONLY).

SPECIAL: IF A PDP-8A OPTION BOARD #1 IS TO BE TESTED, THE FOLLOWING HARDWARE IS REQUIRED:

- A. ONE QUAD EXTENDER MODULE
- B. ONE BC08R-01 CABLE
- C. ONE EIA TEST CONNECTOR (OPTIONAL)
- D. THREE TERMI-POINT JUMPERS.

2.2

STORAGE

THE PROGRAM INITIALLY OCCUPIES LOCATIONS 0000 TO 0155 AND 0200 TO 3722. LOCATIONS 0156 TO 0176 ARE RESERVED FOR THE RIM LOADER ON COMPUTERS CONTAINING LESS THAN 4K OF MEMORY. THE PROGRAM USES LOCATIONS 0000 TO 0155 AND 3600 TO 3722 FOR PROGRAM INITIALIZATION, ONCE THE PROGRAM HAS BEEN STARTED, THESE LOCATIONS AND THE LOCATIONS RESERVED FOR THE RIM LOADER WILL BE DESTROYED. ALL LOCATIONS OUTSIDE THE PROGRAM AREA

IN THE PROGRAM FIELD AND ANY OTHER FIELD, IF SELECTED, ARE USED AS A TEST AREA. THE TEST AREA IS INITIALLY FILLED WITH HALTS AND REFILLED AFTER EVERY 4096 TEST INSTRUCTIONS HAVE BEEN EXECUTED. IF THE PROGRAM IS ALLOWED TO RELOCATE, IT WILL RELOCATE UP AND BACK WITHIN A 2K TO 4K MEMORY FIELD AND UP AND DOWN BETWEEN MEMORY FIELD IF MORE THAN 4K.

2.3 PREREQUISITE SOFTWARE

PDP-8A CPU TEST
PDP-8A MEMORY TESTS
KM8-A OPTION TEST #2 - IF PDP-8A OPTION BOARD #2 IN SYSTEM
DKC8-AA OPTION TEST #1 - IF PDP-8A OPTION BOARD #1 IN SYSTEM

3.0 RESTRICTIONS

3.1 HARDWARE RESTRICTIONS

- A. MINIMUM OF 2K OF MEMORY TO A MAXIMUM OF 32K OF MEMORY
- B. IF A PDP-8A OPTION BOARD #1 IS TO BE USED WITH THE PROGRAM, THE OPTION BOARD MUST BE CONNECTED IN LOOP BACK FOR THE 12 BIT PARALLEL I/O AND THE SERIAL LINE UNIT.

3.2 SOFTWARE RESTRICTIONS

ALL THE SOFTWARE RESTRICTIONS LISTED BELOW PLUS FIELD AND MEMORY SIZE MUST BE ADHERED TO, OTHERWISE, THERE IS NO GUARANTY WHAT WILL HAPPEN TO THE PROGRAM.

- A. ONCE THE PROGRAM HAS BEEN STARTED, THE PROGRAM LOADER(S) WILL BE DESTROYED.
- B. ONCE THE PROGRAM HAS BEEN STARTED AND IT WAS INITIALIZED TO RUN WITH OR WITHOUT THE HARDWARE FRONT PANEL SWITCH REGISTER, THE PROGRAM CAN NOT BE RE-INITIALIZED UNLESS THE PROGRAM IS RELOADED.
- C. BEFORE EACH PROGRAM START, LOCATION 0021 IN THE PROGRAM FIELD MUST BE INITIALIZED FOR THE AMOUNT OF MEMORY TO BE TESTED. LOCATION 0021 INITIALLY IS PRESET TO 0001 (NO FRONT PANEL SWITCH REGISTER AND 2K OF MEMORY).
- D. ONCE THE PROGRAM HAS RELOCATED INTO ANOTHER MEMORY AREA, AND IT IS DESIRED TO CHANGE MEMORY SIZE, MEMORY SIZE CANNOT BE DECREASED BELOW THE 1K SEGMENT THAT THE PROGRAM IS LOCATED IN.
- E. IF THE FRONT PANEL SWITCH REGISTER WAS SELECTED, ALWAYS STOP THE PROGRAM BY SETTING THE SWITCH REGISTER TO 0400. THIS IS DONE TO INSURE THAT THE PROGRAM IS NOT IN THE PROCESS OF RELOCATING. FOR THOSE SYSTEMS WITHOUT A FRONT PANEL, IT IS BEST TO RELOAD THE PROGRAM.

F. IF THE PROGRAM WAS INITIALIZED TO ANY JK FIELD, THE PROGRAM SHOULD ALSO BE RUN AT 2K. THE REASON FOR THIS IS THAT IN ANY JK MEMORY FIELD, ADDRESSES 2000 TO 3777 WILL NOT BE TESTED.

4.0

STANDARD TEST PROCEDURE

IF THE PDP-8A OPTION BOARD #1 IS TO BE USED IN CONJUNCTION WITH THE PROGRAM DO STEP A BELOW, OTHERWISE GO TO STEP B.

A. DO THE FOLLOWING STEPS FOR THE INITIAL HARDWARE SETUP OF THE PDP-8A OPTION BOARD #1:

1. TURN THE AC POWER OFF ON THE COMPUTER
2. UNPLUG THE PDP-8A OPTION BOARD #1 (M8316) FROM THE COMPUTER
3. PLUG THE QUAD EXTENDER INTO THE SLOT THE M8316 OCCUPIED
4. CONNECT BERG SOCKETS J4 AND J5 ON THE M8316 MODULE IN PARALLEL USING THE BC08R-01 CABLE. IN CABLING THE BC08R-01 CABLE, MAKE SURE THAT THE LETTERING IS FACING UPWARDS ON ONE END OF THE CABLE AND DOWNWARDS ON THE OTHER END.
5. SET ALL THE SWITCHES ON THE M8316 MODULE TO THE OFF POSITION, AND THEN SET S1-5, S1-6, AND S1-8 TO THE ON POSITION.
6. PLUG THE M8316 MODULE INTO THE QUAD EXTENDER.
7. TURN THE AC POWER BACK ON TO THE COMPUTER.
8. GO TO STEP B.

B. IF THE COMPUTER CONTAINS AT LEAST 4K OF MEMORY, LOAD THE PAPER TAPE, MAINDEC-08-DJEXB-A-PB, USING THE STANDARD BINARY LOADER TECHNIQUE AND THEN GO TO STEP D. IF THE COMPUTER CONTAINS LESS THAN 4K OF MEMORY, DEPOSIT THE APPROPRIATE RIM LOADER INTO THE LOCATIONS LISTED BELOW IN FIELD 0.

LOCATION -----	LOW SPEED RIM -----	HIGH SPEED RIM -----
0156	6032	6014
0157	6031	6011
0160	5357	5357
0161	6036	6016
0162	7106	7106
0163	7006	7006
0164	7510	7510
0165	5337	5374
0166	7006	7006
0167	6031	6011
0170	5367	5367
0171	6034	6016
0172	7420	7420
0173	3776	3776
0174	3376	3376
0175	5356	5357

C. PLACE THE PAPER TAPE, MAINDEC-08-DJEXB-A-PM, INTO THE DESIRED READER, "LOAD ADDRESS" TO 0156, PRESS "INIT" AND THEN "RUN", WHEN THE TAPE HAS BEEN LOADED, HALT THE COMPUTER AND GO TO THE NEXT STEP.

D. IF THE PDP-8A OPTION BOARD #1 IS TO BE USED IN CONJUNCTION WITH THE PROGRAM, DO THE ITEMS LISTED BELOW. IF THE PDP-8A OPTION BOARD #1 IS NOT USED, GO TO PARAGRAPH 4.2, RUN PDP-8A PROCESSOR EXERCISER.

1. WITHOUT TURNING POWER OFF ON COMPUTERS CONTAINING RAM MEMORY, TURN OFF THE TELETYPE AND UNPLUG THE TELETYPE CABLE FROM THE J3 BERG SOCKET ON THE M8316 MODULE IF CONNECTED.
2. NOW USING THE TERMINAL POINT JUMPERS, JUMPER THE APPROPRIATE PINS ON THE J3 BERG SOCKET ON THE M8316 MODULE TO THE DESIRED LOOP BACK LISTED BELOW.
 - A. EIA LOOP BACK - PIN F TO PIN J AND PIN E TO PIN M OR USE THE EIA TEST CONNECTOR IF AVAILABLE.
 - B. 20MA LOOP BACK - PIN E TO PIN H, PIN K TO PIN KK, AND PIN S TO PIN AA.
3. GO TO PARAGRAPH 4.2, RUN PDP-8A PROCESSOR EXERCISER

4.1 CHANGING IOT CODES

NOT APPLICABLE

4.2 RUN PDP-8A PROCESSOR EXERCISER

THE PROGRAM MUST "ALWAYS" BE INITIALIZED AT ANY PROGRAM RESTART. ONCE THE PROGRAM IS INITIALIZED TO RUN WITH OR WITHOUT THE HARDWARE FRONT PANEL SWITCH REGISTER, IT CANNOT BE REINITIALIZED AGAIN UNLESS THE PROGRAM IS RELOADED.

IN ORDER TO FIND, INITIALIZE, START AND STOP THE PROGRAM DO THE FOLLOWING STEPS.

- A. IF THE PROGRAM WAS JUST LOADED, THE PROGRAM WILL RESIDE IN THE FIRST 2K OF FIELD 0 AND THE STARTING ADDRESS WILL BE 0200. IF THIS WAS THE CASE GO TO STEP B TO INITIALIZE THE PROGRAM, TO FIND THE PROGRAM AND STARTING ADDRESS OF THE PROGRAM, ONCE THE PROGRAM HAS BEEN STARTED, DO THE FOLLOWING STEPS.
 1. LOAD EXTENDED ADDRESS TO FIELD 0
 2. LOAD ADDRESS TO ADDRESS 0005 AND EXAMINE THAT LOCATION
 3. THE CONTENTS OF LOCATION 0005 WILL CONTAIN THE STARTING ADDRESS OF THE PROGRAM AND THE FIELD THAT THE PROGRAM IS LOCATED IN. THE CONTENTS OF ADDRESS 0005 WILL BE IN THE FOLLOWING FORMAT SAFO. SA EQUALS THE MOST SIGNIFICANT SIX BITS OF THE STARTING ADDRESS, F EQUALS THE FIELD THAT THE PROGRAM IS LOCATED IN, SA00 WILL BE THE NEW STARTING ADDRESS OF THE PROGRAM
 4. LOAD THE INSTRUCTION AND DATA FIELD TO THE FIELD THAT THE PROGRAM IS LOCATED IN (OBTAINED FROM STEP 3 ABOVE).
 5. GO TO STEP B TO INITIALIZE THE PROGRAM
- B. THE PROGRAM WHEN FIRST LOADED IS INITIALIZED TO THE FOLLOWING CONDITIONS; NO HARDWARE FRONT PANEL SWITCH REGISTER, NO PDP-8A OPTION BOARD #1 TESTING, AND MEMORY SIZE OF 2K. TO CHANGE THE INITIAL CONFIGURATION OR IF THE PROGRAM IS TO BE RESTARTED, LOAD ADDRESS TO 0021

IN THE PROGRAM FIELD, NOW DEPOSIT INTO THIS LOCATION THE HARDWARE CONFIGURATION AND MEMORY SIZE TO BE TESTED FROM THE TABLES BELOW,

HARDWARE CONFIGURATION BITS 0 AND 1 = LOCATION 0021

BIT 0=0 USE LOCATION 0020 AS A PSEUDO SWITCH REGISTER
 BIT 0=1 USE THE HARDWARE FRONT PANEL SWITCH REGISTER
 BIT 1=0 DO NOT TEST THE PDP-8A OPTION BOARD #1
 BIT 1=1 TEST THE PDP-8A OPTION BOARD #1

MEMORY SIZE = BITS 7-11 = LOCATION 0021

MEMORY LOC 21 MEMORY LOC 21 MEMORY LOC 21 MEMORY LOC 21

1K	N/A	9K	0010	17K	0020	25K	0030
2K	0001	10K	0011	18K	0021	26K	0031
*3K	0002	*11K	0012	*19K	0022	*27K	0032
4K	0003	12K	0013	20K	0023	28K	0033
5K	0004	13K	0014	21K	0024	29K	0034
6K	0005	14K	0015	22K	0025	30K	0035
*7K	0006	*15K	0016	*23K	0026	*31K	0036
8K	0007	16K	0017	24K	0027	32K	0037

*REFER TO STEP F IN SOFTWARE RESTRICTIONS, PARAGRAPH 3.2.

IF BIT 0 OF LOCATION 0021 WAS SET TO A 0, LOAD ADDRESS TO LOCATION 0020 AND DEPOSIT INTO THIS LOCATION THE SWITCH REGISTER SETTING DESIRED (NORMALLY ALL ZEROES),

- C. TO START THE PROGRAM LOAD ADDRESS TO THE ADDRESS OBTAINED IN STEP A ABOVE, SET THE SWITCH REGISTER TO THE DESIRED SWITCH SETTING (NORMALLY ZEROES), NOW PRESS "INIT" AND THEN "RUN", THE PROGRAM SHOULD NOW RUN.
- D. TO STOP THE PROGRAM "ALWAYS" SET THE SWITCH REGISTER OR PSEUDO SWITCH REGISTER WHICHEVER SELECTED TO 0400. FAILURE TO DO THIS MAY DESTROY THE PROGRAM WHEN IT IS RESTARTED, THE PROGRAM WILL HALT AFTER 4096 TEST INSTRUCTIONS HAVE BEEN EXECUTED IF THE SWITCH REGISTER WAS SET TO 0400. RESETTNG THE SWITCH REGISTER AND PRESSING "INIT" AND THEN "RUN" WILL CAUSE THE PROGRAM TO CONTINUE TESTING.
- E. THE PROGRAM SHOULD RUN UNTIL STOPPED BY THE OPERATOR, IF THE PROGRAM HALTED AND THE SWITCH REGISTER WAS NOT SET TO 0400, REFER TO THE SECTION ON ERRORS.
- F. RUN THIS PROGRAM FOR 30 MINUTES. A PROGRAM PASS WILL RANGE BETWEEN 3 SECONDS TO 14 SECONDS DEPENDING ON MEMORY SIZE AND CYCLE TIME.
- G. TO RESTART THE PROGRAM, DO STEPS A, B AND C.

5.0

ERRORS

ALL ERRORS DETECTED BY THE PROGRAM WILL RESULT IN AN ERROR HALT. REFER TO THE SECTION ON ERROR HALTS TO DETERMINE IF THE ERROR WAS A FAILURE DUE TO A MEMORY REFERENCE INSTRUCTION, OPERATE INSTRUCTION, SERIAL LINE UNIT DATA ERROR, PARALLEL I/O DATA ERROR, ILLEGAL INTERRUPT, BATTERY EMPTY, INACTIVE DEVICE, OR A RELOCATION ERROR,

5.1

ERROR HALTS

TO DETERMINE WHAT TYPE OF ERROR WAS DETECTED BY THE PROGRAM, REFERENCE THE ERROR HALTS LISTED BELOW AND GO TO THE PARAGRAPH DESCRIBING THE ERROR AND FOR THE ERROR RECOVERY. ANY ERROR HALTS WHICH OCCUR AND DO NOT CORRESPOND TO ANY OF THE ADDRESSES LISTED BELOW, ARE CATASTROPHIC ERRORS. THESE ERRORS ARE PROBABLY DUE TO EXECUTION OF A INSTRUCTION TO THE WRONG ADDRESS OR FIELD. THE HEADERS FOR THE ERROR HALTS LISTED BELOW ARE DEFINED AS FOLLOWS:

MIR ERR = MEMORY REFERENCE INSTRUCTION ERROR (AND-TAD-ISZ-DCA-JMS-JMP)
 OPR ERR = OPERATE INSTRUCTION ERROR
 SLU D. ERR = SERIAL LINE UNIT DATA ERROR
 P. I/O D. ERR = 12 BIT PARALLEL I/O DATA ERROR
 ILL. INT = ILLEGAL INTERRUPT ERROR
 INACT. DEV. = INACTIVE DEVICE ERROR
 BAT EMP = BATTERY EMPTY
 SAF0 = STARTING ADDRESS AND FIELD PROGRAM IS LOCATED IN
 ROLL UP = RELOCATION ERROR WHILE ROLLING UP IN A MEMORY FIELD
 ROLL DOWN = RELOCATION ERROR WHILE ROLLING DOWN IN A MEMORY FIELD
 SWAP FIELDS = RELOCATION ERROR DURING RELOCATION TO ANOTHER FIELD

SAF0 = SA IS THE STARTING ADDRESS OF THE PROGRAM (SA00)
 AND F IS DETERMINED BY THE OPERATOR FOR THE
 FIELD WHICH THE PROGRAM HALTED IN. SAF0
 SHOULD AGREE WITH ADDRESS 0005 IN FIELD ZERO.

MIR ERR	OPR ERR	SLU D ERR	P. I/O D. ERR	SAF0
-----	-----	-----	-----	-----
1337	1742	3213	3251	02F0
1537	2142	3413	3451	04F0
1737	2342	3613	3651	06F0
2137	2542	4013	4051	10F0
2337	2742	4213	4251	12F0
2537	3142	4413	4451	14F0
2737	3342	4613	4651	16F0
3137	3542	5013	5051	20F0
3337	3742	5213	5251	22F0
3537	4142	5413	5451	24F0
3737	4342	5613	5651	26F0
4137	4542	6013	6051	30F0
4337	4742	6213	6251	32F0
4537	5142	6413	6451	34F0
4737	5342	6613	6651	36F0
5137	5542	7013	7051	40F0
5337	5742	7213	7251	42F0
5537	6142	7413	7451	44F0

ILL. INT -----	INACT DEV -----	BAT EMP -----	SAF0 -----
3132	3323	3311	02F0
3332	3523	3511	04F0
3532	3723	3711	06F0
3732	4123	4111	10F0
4132	4323	4311	12F0
4332	4523	4511	14F0
4532	4723	4711	16F0
4732	5123	5111	20F0
5132	5323	5311	22F0
5332	5523	5511	24F0
5532	5723	5711	26F0
5732	6123	6111	30F0
6132	6323	6311	32F0
6332	6523	6511	34F0
6532	6723	6711	36F0
6732	7123	7111	40F0
7132	7323	7311	42F0
7332	7523	7511	44F0

RELOCATION ERRORS

ROLL UP -----	ROLL DOWN -----	SWAP FIELDS -----	SAF0 -----
0233	3565	0466	N/A
0433	3765	0666	N/A
0633	4165	1066	N/A
1033	4365	1266	N/A
1233	4565	1466	N/A
1433	4765	1666	N/A
1633	5165	2066	N/A
2033	5365	2266	N/A
2233	5565	2466	N/A
2433	5765	2666	N/A
2633	6165	3066	N/A
3033	6365	3266	N/A
3233	6565	3466	N/A
3433	6765	3666	N/A
3633	7165	4066	N/A
4033	7365	4266	N/A
4233	7565	4466	N/A
4433	7765	4666	N/A

5.2 ERROR PRINTOUTS

NOT APPLICABLE

5.3 MEMORY REFERENCE INSTRUCTION ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX37 FOR MEMORY REFERENCE INSTRUCTION (AND-TAD-ISZ-DCA-JMS-JMP) ERRORS. XX37 WILL BE A COMMON HALT FOR ALL ITEMS LISTED BELOW IN THE TABLE. RECORD THE CONTENTS OF THE AC OF THIS HALT INTO THE FIRST ITEM IN THE TABLE AND THEN PRESS "RUN", CONTINUE WITH THE SEQUENCE UNTIL EACH ITEM IN THE TABLE BELOW IS FILLED.

MEMORY REFERENCE INSTRUCTION INFORMATION TABLE

HALT #	ADDRESS	CONTENTS OF AC	DESCRIPTION
-----	-----	-----	-----
HALT #1	XX37		FIELD THAT PROGRAM PUT INSTRUCTION IN
HALT #2	XX37		INSTRUCTION RETURNED FROM THIS FIELD AFTER EXECUTION OF INSTRUCTION
HALT #3	XX37		EXPECTED PC RETURN FROM INSTRUCTION
HALT #4	XX37		ACTUAL PC RETURN FROM INSTRUCTION
HALT #5	XX37		ADDRESS WHERE INSTRUCTION WAS PLACED
HALT #6	XX37		TEST INSTRUCTION - THE INSTRUCTION WHICH WAS EXECUTED
HALT #7	XX37		REFERENCE ADDRESS - ADDRESS WHICH THE INSTRUCTION WILL REFERENCE, OR IF THE INSTRUCTION IS INDIRECT, THIS ADDRESS WILL CONTAIN THE INDIRECT ADDRESS.
HALT #8	XX37		INDIRECT ADDRESS - THIS IS THE INDIRECT ADDRESS WHICH THE TEST INSTRUCTION WILL REFERENCE, N/A FOR DIRECT ADDRESSING INSTRUCTIONS.
HALT #9	XX37		INITIAL MEMORY DATA - MEMORY DATA WHICH IS PUT INTO REFERENCE ADDRESS OR INDIRECT ADDRESS IF INSTRUCTION IS DIRECT OR INDIRECT, N/A FOR JMP OR JMS INSTRUCTIONS.
HALT #10	XX37		FINAL MEMORY DATA - CONTENTS OF REFERENCE ADDRESS OR INDIRECT ADDRESS AFTER EXECUTION OF INSTRUCTION, FOR A JMP INSTRUCTION, THIS NUMBER SHOULD BE EQUAL TO A CIF X, FOR

* JMS INSTRUCTION, THIS NUMBER SHOULD EQUAL THE INSTRUCTION ADDRESS (HALT #5) PLUS 1,

HALT #11 XX37

THE CONTENTS OF THE AC BEFORE THE EXECUTION OF THE INSTRUCTION

HALT #12 XX37

THE CONTENTS OF THE AC AFTER THE EXECUTION OF THE TEST INSTRUCTION

HALT #13 XX37

THE STATE OF THE LINK, BEFORE THE EXECUTION OF THE INSTRUCTION

HALT #14 XX37

THE STATE OF THE LINK, AFTER THE EXECUTION OF THE TEST INSTRUCTION.

HALT #15 XX37

THE CONTENTS OF THE MQ BEFORE THE TEST INSTRUCTION IS EXECUTED

HALT #16 XX37

THE CONTENT OF THE MQ AFTER THE EXECUTION OF THE TEST INSTRUCTION.

THIS IS THE END OF THE MEMORY REFERENCE INSTRUCTION ERROR INFORMATION. REFER TO MEMORY REFERENCE TEST INSTRUCTION SETUP SECTION, PARAGRAPH 5.4, TO DETERMINE THE TYPE OF ERROR.

TO LOOP ON THIS ERROR, SET THE SWITCH REGISTER OR PSEUDO SWITCH REGISTER WHICHEVER SELECTED TO 7000 AND PRESS "INIT" AND THEN "RUN", THE PROGRAM IS NOW IN A LOOP, LOOPING ON THE SAME CONDITIONS.

5.4 MEMORY REFERENCE TEST INSTRUCTION SETUP

TO DETERMINE THE TYPE OF ERROR, THE OPERATOR MUST UNDERSTAND THE TEST INSTRUCTION SETUP, THE TEST INSTRUCTION SETUPS ARE BROKEN UP INTO GROUPS WHICH ARE LISTED AND DESCRIBED BELOW.

A. AND'S THROUGH DCA'S DIRECT ADDRESSING MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATION 4 OF THIS RANDOM FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM.
3. THE CONTENTS OF THE LINK, AC AND MQ CONTAINS SOME RANDOM NUMBER
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS -1
5. INSTRUCTION ADDRESS -1 = CIF TO PROGRAM FIELD.
6. INSTRUCTION ADDRESS = THE TEST INSTRUCTION
7. INSTRUCTION ADDRESS +1 = JMS I 4 - RETURN TO PROGRAM
8. INSTRUCTION ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM
9. REFERENCE ADDRESS = INITIAL MEMORY DATA - LOCATION THE INSTRUCTION WILL EXECUTE.

B. AND'S THROUGH DCA'S INDIRECT ADDRESS MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATION 4 OF THIS FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM
3. THE CONTENTS OF THE LINK, AC, AND MQ CONTAINS SOME RANDOM NUMBER
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS -1
5. INSTRUCTION ADDRESS -1 = CIF TO PROGRAM FIELD

6. INSTRUCTION ADDRESS = THE TEST INDIRECT INSTRUCTION
7. INSTRUCTION ADDRESS +1 = JMS I 4 - RETURN TO PROGRAM
8. INSTRUCTION ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM
9. REFERENCE ADDRESS = INDIRECT ADDRESS = THE ADDRESS THE INSTRUCTION WILL REFERENCE
10. INDIRECT ADDRESS = INITIAL MEMORY DATA - THE LOCATIONS THE INSTRUCTION WILL EXECUTE

C. JMP'S = DIRECT ADDRESSING MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATION 4 OF THIS RANDOM FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM,
3. THE CONTENTS OF THE LINK, AC, AND MQ CONTAIN SOME RANDOM NUMBER
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS
5. INSTRUCTION ADDRESS = THE TEST JUMP INSTRUCTION
6. REFERENCE ADDRESS = CIF TO PROGRAM FIELD. TEST INSTRUCTION JUMPS TO HERE
7. REFERENCE ADDRESS +1 = JMS I 4 - RETURN TO PROGRAM
8. REFERENCE ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM

D. JMP'S = INDIRECT ADDRESS MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATIONS 4 OF THIS RANDOM FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM,
3. THE CONTENTS OF THE LINK, AC, AND MQ CONTAINS SOME RANDOM NUMBER,
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS
5. INSTRUCTION ADDRESS = THE TEST JMP INDIRECT INSTRUCTION
6. REFERENCE ADDRESS = CONTAINS THE INDIRECT ADDRESS
7. INDIRECT ADDRESS = CIF TO PROGRAM FIELD
8. INDIRECT ADDRESS +1 = JMS I 4 - RETURN TO PROGRAM
9. INDIRECT ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM

E. JMS'S = DIRECT ADDRESS MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATION 4 OF THIS FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM,
3. THE CONTENTS OF THE LINK, AC, AND MQ CONTAINS SOME RANDOM NUMBER,
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS
5. INSTRUCTION ADDRESS = THE TEST JMS INSTRUCTION
6. REFERENCE ADDRESS = SHOULD CONTAIN INSTRUCTION ADDRESS +1 AFTER EXECUTION OF TEST INSTRUCTION
7. REFERENCE ADDRESS +1 = CIF TO PROGRAM FIELD
8. REFERENCE ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM
9. REFERENCE ADDRESS +3 = JMS I 4 - RETURN TO PROGRAM

F. JMS'S = INDIRECT ADDRESS MODE

1. INSTRUCTION SETUP IS PUT IN SOME RANDOM FIELD
2. LOCATION 4 OF THIS FIELD CONTAINS THE RETURN POINTER TO PROGRAM,
3. THE CONTENTS OF THE LINK, AC AND MQ CONTAINS SOME RANDOM NUMBER,
4. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS
5. INSTRUCTION ADDRESS = THE TEST JMS INDIRECT INSTRUCTION
6. REFERENCE ADDRESS = INDIRECT ADDRESS
7. INDIRECT ADDRESS = SHOULD CONTAINS THE INSTRUCTION ADDRESS +1 AFTER EXECUTION OF INSTRUCTION
8. INDIRECT ADDRESS +1 = CIF TO PROGRAM FIELD
9. INDIRECT ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM
10. INDIRECT ADDRESS +3 = JMS I 4 - RETURN TO PROGRAM,

OPERATE INSTRUCTION ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX42 FOR ALL OPERATE INSTRUCTION ERRORS. XX42 WILL BE A COMMON HALT FOR ALL IDEMS LISTED BELOW IN THE TABLE. RECORD THE CONTENTS OF THE AC OF THIS HALT INTO THE FIRST IDEM IN THE TABLE AND THEN PRESS "RUN". CONTINUE WITH THIS SEQUENCE UNTIL EACH IDEM IN THE TABLE BELOW IS FILLED.

OPERATE INSTRUCTION INFORMATION TABLE

<u>HALT #</u>	<u>ADDRESS</u>	<u>CONTENTS OF AC</u>	<u>DESCRIPTION</u>
HALT #1	XX42		FIELD THAT PROGRAM PUT INSTRUCTION IN
HALT #2	XX42		INSTRUCTION RETURNED FROM THIS FIELD AFTER EXECUTION OF INSTRUCTION
HALT #3	XX42		EXPECTED PC RETURN FROM INSTRUCTION
HALT #4	XX42		ACTUAL PC RETURN FROM INSTRUCTION
HALT #5	XX42		ADDRESS WHERE INSTRUCTION WAS PLACED
HALT #6	XX42		TEST INSTRUCTION - THE INSTRUCTION WHICH WAS EXECUTED
HALT #7	XX42		THE CONTENTS OF THE AC BEFORE THE INSTRUCTION WAS EXECUTED
HALT #8	XX42		THE SIMULATED RESULTS OF THE AC, AS CALCULATED BY THE PROGRAM, OF WHAT THE AC SHOULD BE AFTER THE EXECUTION OF THE TEST OPERATE INSTRUCTION
HALT #9	XX42		THE CONTENTS OF THE AC AFTER THE EXECUTION OF THE TEST INSTRUCTION.
HALT #10	XX42		THE CONTENTS OF THE LINK BEFORE THE TEST OPERATE INSTRUCTION WAS EXECUTED.
HALT #11	XX42		THE SIMULATED RESULTS OF THE LINK AFTER THE TEST INSTRUCTION WAS EXECUTED AS CALCULATED BY THE PROGRAM
HALT #12	XX42		THE CONTENT OF THE LINK AFTER THE EXECUTION OF THE TEST INSTRUCTION
HALT #13	XX42		THE CONTENTS OF THE MQ BEFORE THE EXECUTION OF THE TEST INSTRUCTION
HALT #14	XX42		THE SIMULATED RESULTS OF THE MQ, AFTER EXECUTION OF THE TEST OPERATE INSTRUCTION AS CALCULATED BY THE PROGRAM.

HALT #15

THE CONTENTS OF THE MQ AFTER
THE EXECUTION OF THE TEST INSTRUCTION.

THIS IS THE END OF THE OPERATE INSTRUCTION ERROR INFORMATION. ERRORS ENCOUNTERED UNDER THIS SECTION MAY BE DUE TO THE EXECUTION OF THE TEST OPERATE INSTRUCTION OR THE SIMULATION OF TEST INSTRUCTION DONE BY THE PROGRAM. REFER TO PARAGRAPH 5.6 FOR OPERATE TEST INSTRUCTION SETUP.

TO LOOP ON A OPERATE TEST INSTRUCTION ERROR, SET THE SWITCH REGISTER OR PSEUDO SWITCH REGISTER WHICHEVER SELECTED TO 7000 AND PRESS "INIT" AND THEN "RUN", THE PROGRAM IS NOW IN A LOOP, LOOPING ON THE SAME CONDITIONS.

5.6 OPERATE TEST INSTRUCTION SETUP

TO DETERMINE THE TYPE OF ERROR, THE OPERATOR MUST UNDERSTAND THE TEST INSTRUCTION SETUP, THE OPERATE TEST INSTRUCTION SETUP IS LISTED BELOW.

- A. BEFORE THE EXECUTION OF THE TEST OPERATE INSTRUCTION, THE PROGRAM SIMULATES THE RESULTS OF THE EXECUTION OF THE TEST OPERATE INSTRUCTION UPON THE LINK, AC, AND MQ
- B. THE INSTRUCTION SETUP IS PLACED IN SOME RANDOM FIELD
- C. LOCATION 4 OF THIS RANDOM FIELD CONTAINS THE RETURN POINTER TO THE PROGRAM,
- D. THE CONTENTS OF THE LINK, AC AND MQ CONTAINS SOME RANDOM NUMBER.
- E. THE PROGRAM JUMPS TO THE INSTRUCTION ADDRESS -1.
- F. INSTRUCTION ADDRESS -1 = CIF TO PROGRAM FIELD
- G. INSTRUCTION ADDRESS = THE TEST OPERATE INSTRUCTION
- H. INSTRUCTION ADDRESS +1 = JMS I 4 - RETURN TO PROGRAM.
- I. INSTRUCTION ADDRESS +2 = JMS I 4 - RETURN TO PROGRAM.

5.7 SERIAL LINE UNIT DATA ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX13 FOR ALL SERIAL LINE UNIT DATA ERRORS, TO FIND OUT THE WORD THAT WAS TRANSMITTED AND THE WORD THAT WAS READ, DO THE FOLLOWING:

- A. LOAD ADDRESS TO XX32, WHERE XX IS THE MOST SIGNIFICANT SIX BITS OF THE SERIAL LINE UNIT DATA ERROR HALT ADDRESS, EXAMPLE - IF THE PROGRAM HALTED AT 3213, LOAD ADDRESS TO 3232.
- B. EXAMINE ADDRESS XX32 FOR THE WORD TRANSMITTED
- C. EXAMINE NEXT ADDRESS XX33 FOR THE WORD RECEIVED

- D. IF IT IS DESIRED TO LOOP ON THIS DATA PATTERN, DO THE FOLLOWING STEPS, OTHERWISE, GO TO STEP E TO CONTINUE TESTING.
1. LOAD ADDRESS TO THE ERROR HALT ADDRESS, CHANGE THE CONTENTS OF THIS ADDRESS FROM 7402 TO 7300.
 2. LOAD ADDRESS TO XX21, WHERE XX IS THE MOST SIGNIFICANT SIX BITS OF THE SERIAL LINE UNIT DATA ERROR HALT ADDRESS. EXAMPLE - IF THE PROGRAM HALTED AT 3213, LOAD ADDRESS TO 3221, NOW CHANGE XX21 FROM 3232 TO 7300.
 3. NOW SUBTRACT 0200 FROM THE MOST SIGNIFICANT 6 BITS OF THE ERROR HALT ADDRESS, USING THE NEW NUMBER AND ADDING 0045 ONTO IT TO FORM A NEW NUMBER OF XX45, LOAD ADDRESS TO XX45, EXAMPLE - IF PROGRAM HALTED AT 3213, LOAD ADDRESS TO 3045, NOW CHANGE XX45 FROM 3753 TO 7300.
 4. GO TO NEXT STEP E.
- E. TO CONTINUE TESTING OR IF STEPS 1, 2 AND 3 ABOVE WERE DONE TO LOOP ON SAME DATA PATTERN, LOAD ADDRESS TO ONE ADDRESS MORE THAN THE SERIAL LINE UNIT ERROR HALT ADDRESS (XX13+1=XX14), SET THE SWITCH REGISTER OR PSEUDO SWITCH REGISTER, WHICHEVER SELECTED, TO 1000 TO INHIBIT PROGRAM RELOCATION, DO NOT PRESS "INIT" BUT PRESS "RUN". IF STEPS 1, 2 AND 3 ABOVE WERE DONE, THE PROGRAM WILL RUN TRANSMITTING THE SAME DATA PATTERN AS DETECTED BY THE ERROR, IF THE ABOVE STEPS WERE NOT DONE, THE PROGRAM WILL RUN UNTIL ANOTHER DATA ERROR IS ENCOUNTERED.
- F. IF IT IS DESIRED TO CHANGE THE CONTENTS OF LOCATIONS LISTED IN STEPS 1, 2 AND 3 OF STEP D ABOVE, BACK TO THERE ORIGINAL CONTENTS, STOP THE COMPUTER BY SETTING THE SWITCH REGISTER TO 1400, NOW DO STEPS 1, 2 AND 3 IN STEP D ABOVE IN REVERSE, BY REPLACING THE 7300'S WITH THE ORIGINAL CONTENTS, NOW GO TO PARAGRAPH 4.2 TO RESTART THE PROGRAM.

5.8 12 BIT PARALLEL I/O DATA ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX51 FOR ALL 12 BIT PARALLEL I/O DATA ERRORS. TO FIND OUT THE WORD THAT WAS TRANSMITTED AND THE WORD THAT WAS READ, DO THE FOLLOWING STEPS.

- A. LOAD ADDRESS TO XX34, WHERE XX IS THE MOST SIGNIFICANT SIX BITS OF THE 12 BIT PARALLEL I/O DATA ERROR HALT ADDRESS. EXAMPLE - IF THE PROGRAM HALTED AT ADDRESS 3251, LOAD ADDRESS TO 3234.
- B. EXAMINE ADDRESS XX34 FOR THE WORD TRANSMITTED
- C. EXAMINE NEXT ADDRESS XX35 FOR THE WORD READ
- D. IF IT IS DESIRED TO LOOP ON THIS DATA PATTERN, DO THE FOLLOWING STEPS, OTHERWISE GO TO STEP E TO CONTINUE TESTING.
 1. LOAD ADDRESS TO THE ERROR HALT ADDRESS (XX51) AND CHANGE THE CONTENTS FROM 7402 TO 7300.

2. LOAD ADDRESS TO XX76, WHERE XX IS THE MOST SIGNIFICANT SIX BITS OF THE ERROR HALT ADDRESS, EXAMPLE - IF PROGRAM HALTED AT ADDRESS 3251, LOAD ADDRESS TO 3276, NOW CHANGE THE CONTENT OF ADDRESS XX76 FROM 3234 TO 7300.
3. NOW SUBTRACT 0200 FROM THE MOST SIGNIFICANT SIX BITS OF THE ERROR HALT ADDRESS, USING THIS NEW NUMBER AND ADDING 0047 ONTO IT TO FORM A NEW NUMBER OF XX47, LOAD ADDRESS TO XX47, EXAMPLE - IF PROGRAM HALTED AT 3251, LOAD ADDRESS TO 3047, NOW CHANGE XX47 FROM 3754 TO 7300.
4. GO TO NEXT STEP E.
- E. TO CONTINUE TESTING OR IF STEPS 1, 2 AND 3 WERE DONE ABOVE TO LOOP ON SAME DATA PATTERN, LOAD ADDRESS TO ONE ADDRESS MORE THAN THE ERROR HALT ADDRESS (XX51+1=XX52), SET THE SWITCH REGISTER OR PSEUDO SWITCH REGISTER, WHICHEVER SELECTED, TO 1000 TO INHIBIT PROGRAM RELOCATION, DO NOT PRESS "INIT" BUT PRESS "RUN", IF STEPS 1, 2 AND 3 ABOVE WERE DONE, THE PROGRAM WILL RUN TRANSMITTING THE SAME DATA PATTERN AS DETECTED BY THE ERROR, IF THE ABOVE STEPS WERE NOT DONE, THE PROGRAM WILL RUN UNTIL ANOTHER DATA ERROR IS ENCOUNTERED.
- F. IF IT IS DESIRED TO CHANGE THE CONTENTS OF LOCATIONS LISTED IN STEPS 1, 2 AND 3 OF STEP D ABOVE, BACK TO THERE ORIGINAL CONTENTS, STOP THE COMPUTER BY SETTING THE SWITCH REGISTER TO 1400, NOW DO STEPS 1, 2 AND 3 IN STEP D ABOVE IN REVERSE, BY REPLACING THE 7300'S WITH THE ORIGINAL CONTENTS, NOW GO TO PARAGRAPH 4.2 TO RESTART THE PROGRAM.

5.9 ILLEGAL INTERRUPT ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX32 FOR A ILLEGAL INTERRUPT, THIS ERROR HALT WILL OCCUR IF THE PROGRAM INTERRUPTED WITHOUT ONE OF THE FOLLOWING FLAGS BEING SET ON THE PDP-8A OPTION 1 MODULE, SLU XMIT/RECEIVE FLAG, PARALLEL I/O DATA READY FLAG, REAL TIME CLOCK FLAG, OR AC LOW FLAG, THE ERROR MAY ALSO BE CAUSED BY A FLAG GETTING CLEARED ON A INTERRUPT, OR A FLAG FAILING TO SKIP IN THE SKIP CHAIN, TO RECOVER FROM THIS ERROR, RESTART THE PROGRAM (PARAGRAPH 4.2), IF THE ERROR STILL EXISTS, USE A SCOPE TO SEE WHAT OTHER FLAG IS SET BESIDES THE FLAGS ON THE PDP-8A OPTION BOARD#1 MODULE.

5.10 INACTIVE DEVICE ERROR

- A. THE PROGRAM WILL HALT AT XX23 FOR ANY PDP-8A OPTION BOARD #1 INACTIVE DEVICE ERROR, A INACTIVE DEVICE ERROR IS DEFINED AS ANY DEVICE (SERIAL LINE UNIT, PARALLEL I/O, REAL TIME CLOCK) THAT WAS NOT INTERRUPTED WITHIN A GIVEN TIME, TO FIND OUT WHAT DEVICE OR FLAG IS INACTIVE DO THE FOLLOWING:
 1. SUBTRACT 100 FROM THE MOST SIGNIFICANT SIX BITS OF THE ERROR HALT ADDRESS AND THEN ADD 0026 ON IT TO FORM A ADDRESS OF XX26, NOW LOAD ADDRESS TO THIS NUMBER, THIS NUMBER WILL BE USED IN THE NEXT STEP, EXAMPLE - IF THE PROGRAM HALTED AT 3323, LOAD ADDRESS TO 3226.

2. NOW EXAMINE EACH OF THE FOLLOWING ADDRESSES, TO FIND OUT WHICH FLAG OR DEVICE IS INACTIVE, A LOCATION LISTED BELOW THAT IS SET TO ALL ONES IS THE INACTIVE DEVICE
XX26 = SERIAL LINE UNIT'S TRANSMIT FLAG
XX27 = SERIAL LINE UNIT'S RECEIVE FLAG
XX30 = 12 BIT PARALLEL I/O DATA READY FLAG
XX31 = REAL TIME CLOCK FLAG.

- B. TO RECOVER FROM THIS ERROR, RESTART THE PROGRAM (PARAGRAPH 4.2). IF THE PROBLEM STILL EXISTS, USE A SCOPE OR RUN THE PDP-8A OPTION TEST #1 PROGRAM.

5.11 BATTERY EMPTY ERRORS

THE PROGRAM WILL HALT AT ADDRESS XX11, IF A PDP-8A OPTION BOARD #2 IS INSTALLED AND IF THE SIGNAL AC LOW L IS LOW AND THE BATTERY EMPTY F/F IS SET, THIS ERROR IS CAUSED BY LOSING POWER TO THE COMPUTER, WHICH RESULTS IN DRAINING THE BATTERIES, TO RECOVER FROM THIS HALT, ALLOW THE BATTERIES TIME TO RECHARGE AND THEN RELOAD THE PROGRAM.

5.12 RELOCATION ERRORS

ALL ERRORS WHICH ARE ENCOUNTERED BY MOVING THE PROGRAM UP OR DOWN IN A MEMORY FIELD OR BETWEEN FIELDS ARE CATASTROPHIC AND THE PROGRAM MUST BE RELOADED.

6.0 SWITCH REGISTER SETTINGS

6.1 NORMAL OPERATING SWITCHES

SR2=1 (1000) = INHIBIT PROGRAM RELOCATION
SR3=1 (0400) = HALT THE PROGRAM AT THE COMPLETION OF A PROGRAM PASS ONLY IF SR1=0

6.2 ERROR RELATED SWITCHES

SR0=1 (4000) = INHIBIT ERROR HALTS FOR MEMORY REFERENCE OR OPERATE ERRORS,
SR1=1 (2000) = LOOP ON TEST CONDITIONS FOR MEMORY REFERENCE OR OPERATE INSTRUCTIONS,
SR2=1 (1000) = INHIBIT PROGRAM RELOCATION

7.0 REVISIONS

FIRST SUBMISSION OF THE PROGRAM

PROGRAM DESCRIPTION

THE 2K TO 32K PDP-8A PROCESSOR EXERCISER CHECKS THE EXECUTION OF ALL MEMORY REFERENCE AND OPERATE INSTRUCTIONS IN ALL FIELDS AND ADDRESSES. ALL INSTRUCTIONS, FIELDS, AND DATA ARE SELECTED FROM A RANDOM NUMBER GENERATOR. THE PROGRAM FILLS MEMORY WITH HALTS AFTER EVERY 4096 TEST INSTRUCTIONS HAVE BEEN EXECUTED. IF A OPTION #1 MODULE (M8316) IS SELECTED AND CONNECTED IN LOOP BACK MODE, THE MODULE WILL BE TESTED IN INTERRUPT MODE. DATA TRANSMISSIONS AND INTERRUPTS ARE TESTED ON BOTH THE SERIAL LINE UNIT AND THE 12 BIT PARALLEL I/O. ALSO THE REAL TIME CLOCK INTERRUPT IS TESTED. AFTER EVERY 4096 TEST INSTRUCTIONS HAVE BEEN EXECUTED, THE PROGRAM RELOCATES ITSELF, A PAGE AT A TIME, "UP AND DOWN" WITHIN ANY 2K TO 4K MEMORY FIELD. ONCE THE PROGRAM HAS RELOCATED "UP AND DOWN" WITHIN A SPECIFIC MEMORY FIELD, IT WILL RELOCATE UP INTO THE NEXT FIELD IF MORE THAN 4K OF MEMORY EXISTS AND THE NEXT FIELD CONTAINS AT LEAST 2K. THIS PROCEDURE WILL CONTINUE UNTIL THE LAST MEMORY FIELD IS ENTERED, THEN THE PROGRAM WILL RELOCATE ITSELF DOWN A FIELD AT A TIME UNTIL FIELD ZERO IS REACHED, THEN THE ENTIRE SEQUENCE IS REPEATED. REFER TO THE FOLLOWING PARAGRAPHS FOR MORE DETAILED INFORMATION.

REFER TO PARAGRAPH 5.4 FOR MEMORY REFERENCE TEST INSTRUCTION SETUP. THE PROGRAM VERIFIES THE EXECUTION OF ALL MEMORY REFERENCE INSTRUCTIONS (AND-TAD-ISZ-DCA-JMS-JMP) FOR THE FOLLOWING:

- A. THE INSTRUCTION RETURNED TO THE PROGRAM FROM THE CORRECT FIELD
- B. THE INSTRUCTION RETURNED TO THE PROGRAM FROM THE CORRECT ADDRESS
- C. CORRECT ADDRESSING MODES:
 - 1. DIRECT AND INDIRECT ADDRESSING
 - 2. SAME PAGE AND PAGE 0 ADDRESSING
 - 3. AUTO INDEX ADDRESSING
- D. THE CORRECT MEMORY AND AC DATA AFTER THE EXECUTION OF THE TEST INSTRUCTION.
- E. THE LINK DOESN'T CHANGE FOR THE FOLLOWING INSTRUCTIONS
AND, ISZ, DCA, JMS AND JMP
- F. THE MQ DOESN'T CHANGE,

REFER TO PARAGRAPH 5.6 FOR OPERATE TEST INSTRUCTION SETUP. THE PROGRAM SIMULATES THE EXECUTION OF THE TEST "OPERATE" INSTRUCTION AND VERIFIES THE HARDWARE EXECUTION OF THAT SAME OPERATE INSTRUCTION FOR THE FOLLOWING:

- A. THE INSTRUCTION RETURNED TO THE PROGRAM FROM THE CORRECT FIELD
- B. THE INSTRUCTION RETURNED TO THE PROGRAM FROM THE CORRECT ADDRESS
- C. AC DATA RETURNED EQUALS THE SIMULATED AC DATA
- D. THE LINK DATA RETURNED EQUALS THE SIMULATED LINK DATA
- E. THE MQ DATA RETURNED EQUALS THE SIMULATED MQ DATA

THE OPTION #1 MODULE IS EXERCISED IN INTERRUPT MODE IF SELECTED, THE PROGRAM WHEN FIRST STARTED AND AFTER EACH PROGRAM RELOCATION GENERATES RANDOM DATA FOR THE SERIAL LINE UNIT AND THE 12 BIT PARALLEL I/O. THE INTERRUPT ENABLE FLIP-FLOPS ARE THEN SET FOR THE SERIAL LINE UNIT, THE 12 BIT PARALLEL I/O, AND THE REAL TIME CLOCK. THE RANDOM DATA IS THEN TRANSMITTED ON THE SERIAL LINE UNIT AND THE 12 BIT PARALLEL I/O. THE PROGRAM THEN TURNS THE INTERRUPT ON AND JUMPS TO THE MAIN PART OF THE PROGRAM TO GENERATE AND TEST MEMORY REFERENCE AND OPERATE INSTRUCTIONS. WHEN A INTERRUPT OCCURS, THE PROGRAM DOES THE FOLLOWING:

- A. SAVE THE AC, LINK, AND THE INTERRUPTED PC
- B. THE PROGRAM DOES ONE OF THE FOLLOWING DEPENDING ON THE FLAG SET
 - 1. SLU XMIT FLAG = CLEAR XMIT FLAG, GO TO STEP C
 - 2. SLU RECV FLAG = CLEAR RECV FLAG-COMPARE XMIT DATA WITH DATA READ-GENERATE NEW RANDOM DATA AND TRANSMIT IT, GO TO STEP C.
 - 3. PARALLEL I/O = CLEAR DATA READY FLAG-CHECK DATA ACCEPTED IN-COMPARE DATA, GO TO STEP C.
 - 4. RTC FLAG = CLEAR REAL TIME CLOCK FLAG-GENERATE RANDOM DATA FOR PARALLEL I/O AND TRANSMIT IT, GO TO STEP C,
 - 5. AC LOW FLAG = CLEAR THE FLAG-CHECK BATTERY EMPTY F/F IF SET PROGRAM HALTS, IF NOT GO TO STEP C
- C. THE PROGRAM CHECKS ALL DEVICES TO BE ACTIVE, RESTORES THE LINK, THE AC, ISSUES A RMF INSTRUCTION AND RETURNS TO THE PROGRAM WHERE IT WAS INTERRUPTED FROM.
- D. WHEN THE PROGRAM IS READY TO BE RELOCATED, THE PROGRAM WAITS FOR THE FLAGS AND THEN TURNS THE INTERRUPT OFF.

9.0 FLOWCHARTS

NONE

10. LISTING

ATTACHED

/2K TO 32K PDP-8A PROCESSOR EXERCISER
 /
 /MAINDEC=00-DJEXB=A=L
 /
 /COPYRIGHT 1974, DIGITAL EQUIPMENT CORPORATION
 /
 /PROGRAMMER: BRUCE HANSEN

7421 MQL=7421
 7701 ACL=7701
 7604 LAS=7604
 7402 HLT=7402
 6160 SIMCLR=6160 /CLEAR SIMULATOR LOGIC
 6244 RMP=6244
 6035 KIE=6035
 6007 CAF=6007 /CLEAR ALL FLAGS
 6101 SBE=6101 /SKIP ON BATTERY EMPTY
 6102 SPL=6102 /SKIP ON AC LOW
 6103 CAL=6103 /CLEAR AC LOW F/F
 6135 CLLE=6135 /SET INT ENA ON REAL TIME CLOCK IF DATA BIT 11 ON A 1
 6136 CLCL=6136 /CLEAR REAL TIME CLOCK FLAG
 6137 CLSK=6137 /SKIP ON REAL TIME CLOCK FLAG
 6570 DBST=6570 /SKIP ON DATA ACCEPTED CLEAR IT AND DATA AVAILABLE
 6571 DBSK=6571 /SKIP ON DATA READY
 6572 DBRD=6572 /READ THE 12 BIT PARALLEL I/O REGISTER IN TO THE AC
 6573 DBCF=6573 /CLEAR DATA READY=SET DATA ACCEPTED
 6574 DBTO=6574 /LOAD THE 12 BIT PARALLEL I/O BUFFER AND TRANSMIT
 6575 DBSE=6575 /SET PARALLEL I/O INTERRUPT ENABLE F/F
 6576 DBCE=6576 /CLEAR PARALLEL I/O INTERRUPT ENABLE F/F
 6577 DBSS=6577 /ISSUE A STROBE PULSE

0000 *0

0000 0000 0
 0001 6202 CIF 00/XX
 0002 5403 JMP I INT
 0003 3102 INT, INTERS
 0004 0000 RETPNT, 0 /MRI AND OPR RETURN POINTER
 0005 0200 STRFLD, 8CV /STARTING ADDRESS AND FIELD PROGRAM IS LOCATED IN

0010 *10
 0010 0000 AUTO10, 0
 0011 0000 AUTO11, 0

0020 *20
 0020 0000 SWITCH, 0
 0021 0001 OP1SEL, 0001
 0022 0000 OP2SEL, 0000

/SWITCH REGISTER SETTINGS
 /SR0=1 INHIBIT ERROR HALT
 /SR1=1 LOOP ON ERROR OR TEST CONDITIONS
 /SR2=1 INHIBIT PROGRAM RELOCATION
 /SR3=1 HALT AFTER EXECUTION OF A PROGRAM PASS(4096 TEST INSTRUCTIONS)

/LOCATIONS 0005 TO 0177 WILL BE OVERLAYED ONCE THE PROGRAM HAS BEEN STARTED.
 /IF THE PROGRAM HAS BEEN SETUP TO RUN WITH OR WITHOUT THE FRONT PANEL
 /SWITCH REGISTER, IT CANNOT BE REINITIALIZED AGAIN. THE ONLY WAY TO
 /CHANGE THE FRONT PANEL STATUS IS TO RELOAD THE PROGRAM AND REINITIALIZE IT.

/THE FOLLOWING ROUTINE WILL CHANGE "YAD (1) SAVSWR" TO LAS, IF THE
 /OPERATOR SET BIT 0 OF LOCATION 21 TO A ONE.

0023 0000 PATCH, 0
 0024 1136 TAD K5771
 0025 3540 DCA I LOC200
 0026 1137 TAD K5772
 0027 3541 DCA I LOC201
 0030 7340 CLA CLL CMA
 0031 1023 TAD PATCH
 0032 3023 DCA PATCH
 0033 6160 SIMCLR
 0034 1021 TAD OP1SEL /GET THE HARDWARE CONFIGURATION
 0035 7700 SMA CLA /IS THE FRONT PANEL SWITCH REGISTER TO BE USED?
 0036 5052 JMP PATCH1 /CHECK FOR ACT LINE
 0037 1142 TAD MM6
 0040 3143 DCA LASCNT
 0041 1144 TAD LASTAB
 0042 3145 DCA PATMOV
 0043 1545 TAD I PATMOV
 0044 3146 DCA PATMV1
 0045 1147 TAD KLAS
 0046 3546 DCA I PATMV1
 0047 2145 ISZ PATMOV
 0050 2143 ISZ LASCNT
 0051 5043 JMP ,-6
 0052 1022 PATCH1, TAD OP2SEL /CHECK FOR THE ACT LINE BIT
 0053 7700 SMA CLA /IS IT SET ?
 0054 5423 JMP I PATCH /NO RETURN TO THE PROGRAM
 0055 1125 TAD OVRLAY
 0056 3010 DCA AUTO10
 0057 1126 TAD MRIOVR
 0060 3011 DCA AUTO11
 0061 4103 JMS MOVOVR /GO OVERLAY FIRST 5 LOCATIONS OF ERROR
 0062 1127 TAD OVRLY1
 0063 3010 DCA AUTO10
 0064 1130 TAD OPROVR
 0065 3011 DCA AUTO11
 0066 4103 JMS MOVOVR /GO OVERLAY FIRST 5 LOCATIONS OF ERROR
 0067 1132 TAD K7610 /PUT SKIP UNCONDITIONALLY IN OPRRR+1
 0070 3531 DCA I OPRSKP /IN ORDER TO GO TO ERROPR
 0071 1021 TAD OP1SEL /GET THE HARDWARE CONFIGURATION

```

0072 0134 AND CON37 /MASK OFF MEMORY SIZE
0073 1135 TAD MIN37 /CHECK TO SEE IF 32K SELECTED
0074 7640 SEA CLA /IS THERE 32K SELECTED?
0075 5502 JMP I PATCHC /NO, GO TO NEXT BUFFER TO GET NEXT OVERLAY
0076 7240 CLA CMA /SUBTRACT 1K FROM 32K
0077 1021 TAD OP1SEL
0100 3021 DCA OP1SEL /SAVE MEMORY SIZE AS 31K
0101 5502 JMP I PATCHC /CONTINUE THE OVERLAY FOR ACT LINE
0102 3600 PATCHC, PATCH2

0103 0000 MOVQVR, 0
0104 1133 TAD M5
0105 3145 DCA PATHOV
0106 1410 TAD I AUTO10
0107 3411 DCA I AUTO11
0110 2145 ISE PATHOV
0111 5106 JMP ,+3
0112 5503 JMP I MOVQVR

0113 6002 AEROV1, 10F
0114 6272 CIF 70
0115 1767 1767
0116 5717 5717
0117 6520 6520

0120 6002 AEROV2, 10F
0121 6272 CIF 70
0122 1745 1745
0123 5712 5712
0124 6520 6520

0125 0112 OVRLAY, AEROV1=1
0126 1312 HRIOVR, ERROR-1
0127 0117 OVRLY1, AEROV2=1
0130 2305 OPRQVR, ERROPR=1

0131 2745 OPRSKP, OPRERR+1
0132 7610 K7610, SKP CLA
0133 7773 M5, =5
0134 0037 CON37, 37
0135 7741 MIN37, =37

0136 5771 K5771, 5771
0137 5772 K5772, 5772
0140 0200 LOC200, 3CV
0141 0201 LOC201, 3CV+1
0142 7772 MM6, =6
0143 7772 LASCNT, =6
0144 0150 LASTAB, TABLAS
0145 0000 PATHOV, 0
0146 0000 PATHV1, 0
0147 7604 KLAS, LAS

0150 0252 TABLAS, LPCNT

```

```

0151 0274 XCNT+3
0152 0301 ARRANG=4
0153 1342 LOOPSH
0154 2737 LPSW0
0155 2744 OPRERR

0200 *200
/
0200 0000 BGN, 0/JMS PATCH/JMP I XBGRAN
0201 0000 0/JMS PATCH/JMP I XBCCON

0202 0000 CHANGE, 0
0203 1602 TAD I CHANGE /GET THE WORD TO MODIFY
0204 7450 SNA /IS IT EQUAL TO ZERO
0205 5602 JMP I CHANGE /YES ALL DONE MODIFYING
0206 1212 TAD SUBADD /SUBTRACT OR ADD 200
0207 3602 DCA I CHANGE /RESTORE THE MODIFIED WORD
0210 2202 ISE CHANGE
0211 5203 JMP ,+6 /GET THE NEXT WORD TO MODIFY

/
0212 0000 SUBADD, 0
0213 0000 DIRFLG, 0
0214 0200 LOWLIM, 200
0215 7400 M400, =400

/
0216 1245 SWAP1, TAD S2PRG /ROUTINE TO SWAP PROGRAM UP
0217 3202 DCA CHANGE /SAVE PROGRAM SIZE
0220 1375 TAD XENDPR /MODIFIED END OF PROGRAM
0221 3246 DCA CNTR2
0222 12B1 TAD M200
0223 1375 TAD XENDPR
0224 3247 DCA CNTR3 /GET ACTUAL END OF PROGRAM
0225 1647 MOVUP, TAD I CNTR3
0226 3646 DCA I CNTR2
0227 1647 TAD I CNTR3 /COMPARE THE WORD THAT WAS RELOCATED
0230 7041 CIA
0231 1646 TAD I CNTR2
0232 7640 SEA CLA /COMPARE ERROR DURING RELOCATION
0233 7402 HLT
0234 7040 CMA
0235 1247 TAD CNTR3
0236 3247 DCA CNTR3
0237 7040 CMA
0240 1246 TAD CNTR2
0241 3246 DCA CNTR2
0242 2202 ISE CHANGE
0243 5225 JMP MOVUP
0244 5776 JMP I RSCNT

/
0245 4401 S2PRG, BGN=PRGEND-1
0246 0000 CNTR2, 0
0247 0000 CNTR3, 0
0250 0400 K400, 400
0251 7600 M200, =200

```

```

0292 1336 LPCNT, TAD SAVSWR/LAS          /LOOP ON INSTRUCTION IF SR1 =1
0293 7004          RAL
0294 7700          SMA CLA
0295 9271          JMP XCNT          /EXIT, AND BUMP COUNTERS
0296 1762 RESETT, TAD I XINSTR          /IS INSTRUCTION INDIRECT
0297 0250          AND K400
0260 7650          SNA CLA
0261 5763          JMP I XNTIND          /NO, RESET DATA IN REFERENCE ADDRESS
0262 1762          TAD I XINSTR          /YES, REGENERATE REFERENCE ADDRESS
0263 0214          AND LOWLIM          /MASK OUT PAGE BIT
0264 7640          SZA CLA
0265 1764          TAD I XASAVA
0266 1765          TAD I XASAVB
0267 3766          DCA I XREFAD
0270 5767          JMP I XLOOP
0271 2247 XCNT,   ISE CNTRS          /BUMP PASS COUNTER
0272 5770          JMP I RSCNTX
0273 4760          JMS I WAIT          /IF OPTION 1 SELECTED WAIT FOR FLAGS
0274 1336          TAD SAVSWR/LAS          /CHECK SR3 TO HALT AFTER A PROGRAM PASS
0275 7000          RTL
0276 7004          RAL
0277 7710          SPA          CLA
0300 7402          HLT
0301 1336          TAD SAVSWR/LAS          /SR3=1 HALT AT END OF A PROGRAM PASS
0302 7000          RTL
0303 7710          SPA CLA
0304 5776          JMP I RSCNT          /DO NOT RELOCATE IF SR2=1
0305 6224 ARRANG, RIF
0306 7041          CIA          /GET HOME FIELD
0307 1756          TAD I XFLD          /IS IT EQUAL TO LAST FIELD
0310 7640          SZA CLA
0311 7240          CLA CMA          /NO, PROGRAM IS IN A 4K FIELD
0312 7450          SNA          /YES, IN LAST FIELD GET UPPER LIMITS
0313 1773          TAD I XUPERL
0314 3774          DCA I HIGHLM
0315 1213          TAD DIRFLG          /SAVE UPPER LIMIT
0316 7640          SZA CLA          /IS THE PROGRAM ROLLING UP OR ROLLING BACK
0317 5761          JMP I XROLBK          /THE PROGRAM IS ROLLING BACK
0320 1375 ROLLUP, TAD XENDPR          /GET END OF PROGRAM AND COMPARE IT
0321 7040          CMA          /WITH HIGH LIMITS
0322 1774          TAD I HIGHLM
0323 7650          SNA CLA
0324 5761          JMP I XROLBK          /THIS IS NEEDED FOR A 1K FIELD OTHER THAN 0
0325 5327          JMP SETFLG          /NO, SET REVERSE FLAG
0326 7240          CLA CMA          /-1 IF GOING REVERSE; 0 IF FORWARD
0327 3213 SETFLG, DCA DIRFLG          /ROLLING UP OR ROLLING BACK?
0330 1213          TAD DIRFLG
0331 7640          SZA CLA
0332 1215          TAD H400
0333 1214          TAD LOWLIM          /ROLLING BACK IF DIRECTIONN FLAG = -1
0334 3212          DCA SUBADD          /ROLLING UP IF FLAG = 0
0335 5355          JMP ACHNG          /SAVE 200 OR -200
    
```

```

0336 0000 SAVSWR, 0
/
0337 0000 F0INIT, 0
0340 6201          CDF 00          /CHANGE DATA FIELD TO FIELD 0
0341 6224          RIF
0342 1340          TAD F0INIT+1          /READ THE INSTRUCTION FIELD
0343 7001          TAC          /GET THE CDF INSTRUCTION
0344 3745          DCA I CIFFD0          /MAKE IT A CIF TO PROGRAM FIELD
0345 0001          CIFFD0, INT-2          /PUT IT IN LOCATION 1 OF FIELD 0
0346 7240          CLA
0347 1745          TAD I CIFFD0          /SET THE AC TO ALL ONE'S
0350 3352          DCA ,+2          /CHANGE CIF BACK TO CDF PROGRAM FIELD
0351 4757          JMS I SETINT          /PUT IT IN NEXT LOCATION
0352 7402          HLT/CDF
0353 5737          JMP I F0INIT          TO PROGRAM FIELD
/                                     /RETURN TO PROGRAM
0355 *355
/
0355 4202 ACHNG, JMS CHANGE
/
0356 1144          XFLD, FL0LIM
0357 2137          SETINT, INTSET
0360 3357          WAIT, WAITEN
0361 0401          XROLBK, ROLBAK
0362 0746          XINSTR, INSTR
0363 0625          XNTIND, NOTIND
0364 1146          XASAVA, ASAVA
0365 1147          XASAVB, ASAVB
0366 0747          XREFAD, REFA0
0367 0602          XLOOP, LOOP10+1
0370 1001          RSCNTX, GENFLD
0371 3027          XBCGRN, BGNCON+1
0372 3026          XBGCON, BGNCON
0373 1550          XUPERL, UPRLIM
0374 1145          HIGHLM, HG4LIM
0375 3576          XENDPR, PRGEND
0376 3424          RSCNT, STARTP
0377 0000          0
/
/
0400 0400 *400
0400 5351          JMP AACHNG
/
0401 1367          ROLBAK, TAD BEGIN          /GET BEGINNING OF PROGRAM AND COMPARE IT
0402 7041          CIA          /WITH THE LOW LIMIT
0403 1770          TAD I XLWLMIM
0404 7640          SZA CLA          /
0405 5771          JMP I RTFLGR          /IS IT EQUAL
0406 3772          DCA I RTFLG          /NO, ROLL THE PROGRAM BACK
0407 1773          TAD I MAXFLD          /SET DIRECTION FLAG TO FORWARD
0410 7650          SNA CLA          /IS THE PROGRAM LIMIT ONLY 2K-4K
0411 5774          JMP I RTFLG          /YES, DO NOT SWAP BUT ROLL THE PROGRAM UP
0412 1300          TAD FLDFLG          /SWAP THE PROGRAM UP OR DOWN
    
```

```

0413 7640      SZA CLA
0414 5222      JMP SWAPDN      /SWAP THE PROGRAM DOWN
0415 6224      SWAPUP, RIF      /GET PROGRAM FIELD
0416 1301      TAD K10      /ADD 1 FIELD TO IT
0417 7041      CIA
0420 1773      TAD I MAXFLD
0421 5753      JMP I CSWPUP
0422 6224      SWAPDN, RIF      /GET HOME FIELD
0423 7450      SNA      /IS IT EQUAL TO FIELD 0
0424 5215      JMP SWAPUP      /YES,SWAP THE PROGRAM UP
0425 1303      TAD M10      /SUBTRACT 1 FIELD
0426 7640      SZA CLA      /IS IT EQUAL TO FIELD 0?
0427 5232      JMP SFLDFG-1      /NO,SET FLAG TO REVERSE AND SWAP DOWN
0430 3300      DCA FLDFLG      /YES,BUT SWAP DOWN AND SET FLAG TO FORWARD
0431 5236      JMP ,+5      /GO SWAP IT
0432 7240      CLA CMA
0433 3300      SFLDFG, DCA FLDFLG      /FIELD FLAG=0 SWAP UP;-1 SWAP DOWN
0434 1300      TAD FLDFLG      /SWAPPING UP OR DOWN
0435 7640      SZA CLA
0436 1302      TAD M20      /SWAPPING DOWN
0437 1301      TAD K10      /SWAPPING UP
0440 3276      DCA NEWDFA+1      /SAVE 10 OR -10
0441 6224      RIF      /GET HOME FIELD
0442 1276      TAD NEWDFA+1      /ADD OR SUBTRACT A FIELD
0443 1326      TAD B6201
0444 3257      DCA NEWDTF      /PUT 62X1 IN THE SWAP ROUTINE
0445 6224      RIF      /GET HOME FIELD
0446 1326      TAD B6201
0447 3263      DCA SWPFLD      /TO RETURN BACK TO HOME FIELD
0450 1257      TAD NEWDTF
0451 3275      DCA NEWDFA
0452 1775      SWPUP, TAD I XSIZE      /GET PROGRAM SIZE
0453 3276      DCA NEWDFA+1
0454 1770      TAD I XLWLM      /GET BEGINNING ADDRESS OF PROGRAM
0455 3304      DCA RETHR      /SAVE IT
0456 1704      TAD I RETHR      /GET WORD FROM HOME DF
0457 7402      NEWDTF, HLT/COF      /CHANGE TO NEW DATA FIELD
0460 3704      DCA I RETHR      /PUT THE WORD IN NEW FIELD
0461 1704      TAD I RETHR      /COMPARE THE MOVE
0462 7041      CIA
0463 7402      SWPFLD, HLT/COF      /CHANGE BACK TO OWN FIELD
0464 1704      TAD I RETHR
0465 7640      SZA CLA      /ARE THEY EQUAL?
0466 7402      HLTFIL, HLT      /ERROR DURING RELOCATING TO ANOTHER FIELD
0467 2304      ISZ RETHR
0470 2276      ISZ NEWDFA+1
0471 5256      JMP NEWDTF-1
0472 2257      ISZ NEWDTF      /MAKE 62X2
0473 1257      TAD NEWDTF
0474 3276      DCA ,+2
0475 7402      NEWDFA, HLT/COF      /CHANGE TO NEW DATA FIELD
0476 7402      HLT/CIF
0477 5776      JMP I XGO
    
```

```

0500 0000      FLDFLG, 0
0501 0010      K10, 10
0502 7760      M20, -20
0503 7770      M10, +10
    /
0504 0000      RETHR, 0      /JMS RETURN FROM INSTRUCTION
0505 3327      DCA FILALL      /SAVE AC RETURN DATA
0506 6214      RDF      /GET DATA FIELD INSTRUCTION WAS IN
0507 3325      DCA RETFLD      /SAVE IT
0510 7402      HLT/COF      /RETURN TO PROGRAM FIELD
0511 7701      AC_      /READ IN THE MQ
0512 3756      DCA I RTMQD      /SAVE THE MQ
0513 7010      RAR      /GET THE LINK INTO AC BIT 0
0514 3757      DCA I RTLINK
0515 1760      TAD I BINSTR      /GO TEST THE INSTRUCTION
0516 7006      RTL
0517 7006      RTL
0520 0343      AND B7
0521 1324      TAD BGTST
0522 3323      DCA ,+1
0523 0000      0
    /
0524 5761      BGTST, JMP I TSTINS
0525 0000      RETFLD, 0
0526 6201      B6201, 6201
    /ROUTINE TO FILL THE WHOLE FIELD WITH HALTS
0527 0000      FILALL, 0
0530 3304      DCA RETHR
0531 1754      TAD I XSTFLD
0532 1326      TAD B6201
0533 3340      DCA CDHLT1
0534 6224      RIF
0535 1326      TAD B6201
0536 3344      DCA CDHLT2
0537 1266      TAD HLTFIL
0540 7402      CDHLT1, HLT/COF
0541 3704      DCA I RETHR
0542 2304      ISZ RETHR
0543 0007      B7, 7
0544 7402      CDHLT2, HLT/COF
0545 2755      ISZ I ZLIMIT
0546 5337      JMP ,+7
0547 5727      JMP I FILALL
    /
0551 0551
    /
0551 4752      AACHNG, JMS I XCHNGE
    /
0552 0202      XCHNGE, CHANGE
0553 1116      CSWPUP, DECSWP
0554 0247      XSTFLD, CNTRS
0555 1145      ZLIMIT, HGHLIM
    
```

```

0556 2753 RTM0D, HQ00NE
0557 2751 RTLINK, LINKDN
0560 0746 BINSTR, INSTR
0561 1201 YSTINS, ANDTST
0562 1223 YSTIN1, TADTST
0563 1234 YSTIN2, ISETST
0564 1255 YSTIN3, DCA TST
0565 1267 YSTIN4, JMTST
0566 1304 YSTIN5, JMTST
0567 0200 BEGIN, BGV
0570 0214 XLWLM, LOWLM
0571 0326 RTFLGR, SETFLG=1
0572 0213 RTFLG, DIRFLG
0573 1144 MAXFLD, FLDLM
0574 0327 RTFLGF, SETFLG
0575 0245 XSIZE, SZPRG
0576 3424 XGO, STARTP
0577 0000 Z

0600 *600
/

0600 5366 JMP ACHG

0601 3350 LOOPID, DCA INDA0 /SAVE THIS WORD AS INDIRECT ADDRESS
0602 1360 TAD K7770 /CHECK FOR AUTO-INDEX
0603 1347 TAD REFAD
0604 7510 SPA /WAS IT LESS THAN 10
0605 5211 JMP NOTAUT /YES, NOT AUTO-INDEX
0606 7161 CIA STL
0607 1343 TAD A7
0610 7630 SZL CLA /WAS IT WITHIN AUTO BOUNDARY
0611 7610 SKP CLA /NO, NOT AUTO-INDEX
0612 7340 CLA CLL CMA /AUTO INDEX, SUBTRACT 1 FROM INDIRECT ADDRESS
0613 1350 TAD INDA0
0614 3310 DCA SETRET /SAVE INDIRECT ADDRESS
0615 1354 TAD RANFLD
0616 1356 TAD K6201 /CHANGE TO A RANDOM DATA FIELD
0617 3220 DCA ,+1
0620 7402 HLT/CFD
0621 1310 TAD SETRET /GET INDIRECT ADDRESS
0622 3747 DCA I REFAD /PUT INDIRECT ADDRESS INTO REF ADD
0623 1350 TAD INDA0
0624 3347 DCA REFAD /MAKE REFAD=INDA0
0625 7330 NOTIND, CLA CLL CML RAR
0626 1346 TAD INSTR
0627 7630 SZL CLA /WHAT TYPE OF INSTR
0630 5265 JMP JMPJMS /IT WAS A JMP OR JMS
0631 1354 TAD RANFLD
0632 1356 TAD K6201
0633 3234 DCA ,+1
0634 7402 HLT/CFD /CHANGE TO A RANDOM DATA FIELD
0635 1351 TAD DATATH /GET INITIAL MEMORY DATA AND PUT IT IN
0636 3747 DCA I REFAD /REF ADD OR INDIRECT ADD FOR AND THROUGH DCA
0637 7240 OPRINT, CLA CMA /SUBTRACT 1 FROM INSTRUCTION ADDRESS
    
```

```

0640 1345 TAD ADDR /AND SAVE IT
0641 3344 DCA HOMCIF
0642 6224 RIF
0643 1357 TAD K6202 /SET UP HOME INSTRUCTION FIELD
0644 3744 DCA I HOMCIF /IN INSTRUCTION ADDRESS=1 FOR AND=DCA
0645 7301 CLA CLL IAC
0646 1345 TAD ADDR
0647 4310 JMS SETRET /SETUP RETURN, INSTR ADD+1,+2=4400 FOR AND=DCA
/LOCATION 0 CONTAINS RETURN POINTER
/PUT INSTRUCTION IN INSTRUCTION ADDRESS

0650 1346 NOTJJ, TAD INSTR
0651 3745 DCA I ADDR
0652 1355 TAD SAVLNK
0653 7104 CLL RAL
0654 1353 TAD MQDATA /GET THE RANDOM MQ DATA
0655 7421 MQL /AND LOAD IT INTO THE MQ
0656 7200 CLA /SAFETY CLEAR THE AC IN CASE MQL DOESN'T
0657 1354 TAD RANFLD /MAKE UP A CIF TO A RANDOM FIELD
0660 1357 TAD K6202
0661 3263 DCA ,+2
0662 1352 TAD DATAHR /GET THE AC DATA INTO THE AC
0663 7402 HLT/CIF /D.F. HAS BEEN CHANGED NOW CHANGE I.F.
0664 5744 JMP I HOMCIF /GO EXECUTE INSTRUCTION IN RANDOM FIELD

//FOR AND'S THROUGH DCA'S DIRECTS THE INSTRUCTION SETUP IS AS FOLLOWS:
/
/SOME RANDOM FIELD
/LOCATION 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INST ADD=1= CIF TO PROGRAM FIELD
/INST ADD = TEST INSTRUCTION
/INST ADD+1= JMS I 4
/INST ADD+2= JMS I 4
/
/REF ADD = INITIAL MEMORY DATA, THIS IS THE LOC THE INST WILL REFERENCE
//FOR AND'S THROUGH DCA'S INDIRECTS THE INST SETUP IS AS FOLLOWS
/
/SOME RANDOM FIELD
/LOCATION 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INSTR ADD=1= CIF TO HOME FIELD
/INST ADD = TEST INSTRUCTION
/INST ADD+1= JMS I 4
/INST ADD+2= JMS I 4
/
/REF ADD = INDIRECT ADDRESS
/
/IND ADD = INITIAL MEMORY DAA

0665 1346 JMPJMS, TAD INSTR /GET THE INSTRUCTION
0666 7006 RTL /IS IT A JMP OR JMS?
0667 7700 SMA CLA
    
```



```

0670 7001 IAC /JMS ADD 1 TO REFERENCE ADDRESS FOR CIF INST
0671 1347 TAD REFAD /GET REFERENCE ADDRESS
0672 3310 DCA SETRET /AND SAVE IT FOR THE CIF INSTRUCTION
0673 1354 TAD RANFLD /MAKE CDF INST TO THE RANDOM FIELD
0674 1356 TAD K6201
0675 3276 DCA ,+1
0676 7402 HLT/CDF /CHANGE TO RANDOM DATA FIELD
0677 6224 RIF
0700 1357 TAD K6202 /MAKE A CIF INSTRUCTION TO HOME FIELD
0701 3710 DCA I SETRET /PUT IT IN REFERENCE ADD OR INDIRECT ADD
0702 7001 IAC
0703 1310 TAD SETRET
0704 4310 JMS SETRET /SETUP LOC 4 AND JMS I 4 IN APPROPRIATE PLACES
0705 1345 TAD ADDR /GET INSTRUCTION ADDRESS
0706 3344 DCA HQMCIF /SAVE IT
0707 5250 JMP NOTJJ /GO GET INSTRUCTION AND SETUP
//THE INSTRUCTION SETUP FOR JMP DIRECTS IS AS FOLLOWS:
//
/SOME RANDOM FIELD
/LOC 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INST ADD =JMP INSTRUCTION
/
/REF ADD =CIF TO PROGRAM FIELD
/REF ADD+1 =JMS I 4
/REF ADD+2 =JMS I 4
//THE INSTRUCTION SETUP FOR JMP INDIRECTS IS AS FOLLOWS:
//
/SOME RANDOM FIELD
/LOC 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INST ADD =JMP INDIRECT INSTRUCTION
/
/REF ADD =INDIRECT ADDRESS
/
/IND ADD =CIF TO PROGRAM FIELD
/IND ADD+1 =JMS I 4
/IND ADD+2 =JMS I 4
//THE INSTRUCTION SETUP FOR JMS DIRECTS IS AS FOLLOWS:
//
/SOME RANDOM FIELD
/LOC 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INST ADD =JMS DIRECT INSTRUCTION
/
/REF ADD =SOME UNKNOWN NUMBER
/REF ADD+1 =CIF TO PROGRAM FIELD
/REF ADD+2 =JMS I 4

```

```

/REF ADD+3 =JMS I 4
//THE INSTRUCTION SETUP FOR JMS INDIRECTS IS AS FOLLOWS:
//
/SOME RANDOM FIELD
/LOC 4 OF THIS FIELD EQUALS RETURN POINTER TO PROGRAM FIELD
/THE AC EQUALS SOME RANDOM NUMBER
/INST ADD =JMS INDIRECT INSTRUCTION
/
/REF ADD =INDIRECT ADDRESS
/
/IND ADD =SOME UNKNOWN NUMBER
/IND ADD+1=CIF TO PROGRAM FIELD
/IND ADD+2=JMS I 4
/IND ADD+3=JMS I 4
//THIS ROUTINE SETS UP LOC 0 IN SOME FIELD FOR RETURN POINTER
/TO THE PROGRAM AND ALSO SETS UP THE JMS I 0'S AFTER THE EXECUTION OF THE
/INSTRUCTION.
0710 0000 SETRET, 0
0711 3362 DCA JMSLOC
0712 7301 CLA CLL IAC
0713 1362 TAD JMSLOC
0714 3363 DCA JMSLOC
0715 1364 TAD KJMS
0716 3762 DCA I JMSLOC
0717 1364 TAD KJMS
0720 3763 DCA I JMSLOC
0721 1376 TAD JMSRET
0722 3761 DCA I FLORET
0723 5710 JMP I SETRET

0724 4773 ERROR2, JMS I ZGETWD
0725 4770 JMS I YHALT /FINAL MEMORY DATA
0726 1352 TAD DATAHR /AC DATA BEFORE EXECUTION OF INSTR
0727 4770 JMS I YHALT /AC DATA RETURNED
0730 1774 TAD I EFIND
0731 4770 JMS I YHALT /INITIAL LINK BEFORE EXEC OF INSTR
0732 1355 TAD SAVLNK /
0733 4770 JMS I YHALT /LINK AFTER EXEC OF INSTR
0734 1771 TAD I FLINK
0735 4770 JMS I YHALT /INITIAL HQ DATA
0736 1353 TAD HQDATA
0737 4770 JMS I YHALT /BUMP COUNTER AND RETURN
0740 1772 TAD I FMQDAT
0741 4770 JMS I YHALT
0742 5775 JMP I ZCNT

/
0743 0007 A7, 7
0744 0000 HQMCIF, 0 /ADDRESS OF THE HOME CIF
0745 0000 ADDR, 0 /THE ADDRESS OF THE INSTRUCTION

```

```

0746 0000 INSTR, 0 /THE INSTRUCTION TO TEST
0747 0000 REFAD, 0 /THE ADDRESS THE INSTR, SHOULD REFERENCE
0750 0000 INDAD, 0 /THE INDIRECT ADDRESS
0751 0000 DATATH, 0 /THE DATA IN THE ADDRESS IF AND > DCA
0752 0000 DATAH, 0 /THE DATA IN THE AC AND > JMP
0753 0000 MQDATA, 0 /THE DATA IN THE MQ
0754 0000 RANFLD, 0
0755 0000 SAVLNK, 0
0756 6201 K6201, 6201
0757 6202 K6202, 6202
0760 7770 K7770, 7770
0761 0004 FLDRFT, 4
0762 0000 JMSLOC, 0
0763 0000 JMSLOC, 0
0764 4404 KJMS, JMS I 4
/
0766 /
*766 /
0766 4767 ACHG, JMS I ARERNG /ROUTINE TO ULTER ADDRESSES
/
0767 0202 ARERNG, CHANGE
0770 1336 YHALT, HALT
0771 2751 FLINK, LINKDN
0772 2753 FMOQAT, MQNONE
0773 2112 ZGETWD, GETWD
0774 0527 ZFIND, FIALL
0775 0252 ZCNT, LPONT
0776 0504 JMSRET, RETHR
0777 0000 0
/
1000 1000 /
1000 5350 *1000 JMP AICHG
/
1001 4763 GENFLD, JMS I ARANDY /GET A RANDOM FIELD
1002 0325 AND K70 /MASK WORD FOR FIELD BITS
1003 0327 AND FLDMSK /MASK WORD FOR FIELD
1004 1330 TAD CONFLD /CONSTRAINT WORD FOR FIELD
1005 0325 AND K70
1006 3756 DCA I FLDRAN
1007 1756 TAD I FLDRAN /COMPARE RANDOM FIELD WITH UPPER LIMITS
1010 7041 CIA
1011 1344 TAD FLDLIM
1012 7510 SPA /WITHIN LIMITS ?
1013 5201 JMP GENFLD /NO REGERATE A NEW FIELD
1014 7640 SEA CLA /WAS IT THE LAST MEMORY FIELD
1015 7240 CLA CMA /NO SET UPPER BOUNDARY = TO 7777
1016 7450 SNA /
1017 1757 TAD I XUPLIM /GET THE UPPER LIMIT OF LAST FIELD
1020 3345 DCA HGHLIM /SAVE THE UPPER BOUNDARY
1021 1345 TAD HGHLIM /SETUP A NUMBER FOR BOUNDARY COMPARE
1022 7041 CIA
1023 1326 TAD ADD11
1024 3760 DCA I XBNDCN /SAVE THE NUMBER FOR CHECKING BOUNDRIES

```

```

1025 4763 MEMDAT, JMS I ARANDY /GENERATE RANDOM MEMORY DATA FOR AND>DCA
1026 0335 AND MDTMSK /MASK WORD FOR MEMORY DATA
1027 1336 TAD CONMDT /CONSTRAINT WORD
1030 3771 DCA I ADATAT /SAVE IT
1031 4763 ACDATA, JMS I ARANDY /GENERATE RANDOM AC DATA
1032 0337 AND ACDMSK /MASK WORD
1033 1340 TAD CONACD /CONSTRAINT WORD
1034 3772 DCA I ADATAH /SAVE THE AC DATA WORD
1035 7010 RAR /MOVE THE LINK INTO AC BIT 0
1036 3774 DCA I LNKSVA /SAVE THE LINK
1037 4763 GENMQD, JMS I ARANDY /GENERATE RANDOM MQ DATA
1040 0341 AND MQMSK /MASK WORD FOR MQ DATA
1041 1342 TAD CONMQD /CONSTRAINT WORD FOR MQ DATA
1042 3773 DCA I AMQDAT /SAVE THE MQ DATA WORD
1043 4763 GENAOD, JMS I ARANDY /GENERATE RANDOM ADDRESS FOR INSTRUCTION
1044 0345 AND HGHLIM /MASK OFF ADDRESS BITS FOR THIS FIELD
1045 0331 AND ADRMSK /MASK WORD FOR INSTRUCTION ADDRESS
1046 1332 TAD CONADR /CONSTRAINT WORD
1047 4764 JMS I ABNRY1 /IS IT WITHIN LIMITS
1050 5243 JMP GENAOD /NO, TRY AGAIN
1051 3766 DCA I AADDRS /THIS IS THE INSTRUCTIONS ADDRESS
1052 1766 TAD I AADDRS
1053 0305 AND CONST1
1054 3346 DCA ASAVA /SAVE PAGE BITS FOR FORMING REFERENCE ADDRESS
1055 4755 GENINS, JMS I XGENTI /GENERATE RANDOM INSTRUCTION
1056 4764 JMS I ABNRY1 /IS IT WITHIN LIMITS
1057 5305 JMP CONST1 /NO, TRY AGAIN
1060 4761 JMS I ASAME1 /COMPARE TO ADDR
1061 5305 JMP CONST1 /THERE EQUAL OR TO CLOSE TRY AGAIN
1062 3770 DCA I AREFAD /STORE REFERENCE ADDRESS
1063 6214 RDF
1064 1323 TAD C6201
1065 3765 DCA I XRETHR /PUT QDF HOME FIELD INTO INSTRUCTION RETURN
1066 1767 TAD I AINSTR
1067 0324 AND A400 /INSTR = INSTRUCTION TO TEST
1070 7650 SNA CLA /WAS INSTR INDIRECT
1071 5775 JMP I ANTIND /NO, NOT INDIRECT GO SETUP TEST CONDITIONS
1072 4763 GENIND, JMS I ARANDY /GENERATE RANDOM INDIRECT ADDRESS
1073 0345 AND HGHLIM /MASK OFF ADDRESS BITS FOR THIS FIELD
1074 0333 AND INDMSK /MASK WORD FOR INDIRECT ADDRESS
1075 1334 TAD CONIND /CONSTRAINT WORD FOR INDIRECT
1076 4764 JMS I ABNRY1 /IS IT WITHIN BOUNDARIES
1077 5312 JMP CONST2 /NO, TRY AGAIN
1100 4761 JMS I ASAME1 /COMPARE TO ADDR
1101 5312 JMP CONST2 /TRY AGAIN
1102 4762 JMS I ASAME2 /COMPARE TO REPAD
1103 5312 JMP CONST2 /TRY AGAIN
1104 5776 JMP I ALOPID /GO SETUP TEST CONDITIONS
/
1105 7600 CONST1, 7600
1106 1343 TAD CONFLG
1107 7640 SEA CLA
1110 5243 JMP GENAOD

```

```

1111 5255      JMP GENINS
1112 1343      /
1113 7710      CONST2, TAD CONFLG
1114 5243      SPA CLA
1115 5272      JMP GENADD
1115 5272      JMP GENIND
/
1116 7510      DECSWP, SPA
1117 5752      JMP I FLODFGR
1120 7650      SNA CLA
1121 5754      JMP I CHK1KF
1122 5753      JMP I FLODFG
/IS IT WITHIN FIELD LIMITS
/NO,SET DIRECTION OF SWAP TO REVERSE
/HAS IT THE LAST FIELD?
/GO CHECK TO SEE IF NEXT FIELD IS 1K
/NO,SET DIRECTION OF SWAP TO FORWARD

1123 6201      /
1124 0400      C6201, 6201
1125 0070      A400, 400
1126 0011      K70, 70
1127 7777      ADD11, 11
1128 0000      FLOMSK, 7777
1131 7777      CONFLD, 0
1132 0000      ADRMSK, 7777
1133 7777      CONADR, 0
1134 0000      INMSK, 7777
1135 7777      CONIND, 0
1136 0000      MOTMSK, 7777
1137 7777      CONMDT, 0
1140 0000      ACDMSK, 7777
1141 7777      CONACD, 0
1142 0000      MQDMSK, 7777
1143 0000      CONMQD, 0000
1144 0000      CONFLG, 0
1145 0000      FLDLIM, 0
1146 0000      HGHLIM, 0
1147 0000      ASAVA, 0
1147 0000      ASAVB, 0
/
1150 1150      *1150
/
1150 4751      A1CHG, JMS I A1RRNG
/
1151 0202      A1RRNG, CHANGE
1152 0432      FLODFGR, SFLODFG-1
1153 0433      FLODFG, SFLODFG
1154 2555      CHK1KF, FLJCHK
1155 1601      XGENTI, INSGEN
1156 0754      FLDRAN, RANFLD
1157 1550      XUPLIM, UPRLIM
1160 1551      XBNDCN, BNDCON
1161 1463      ASAME1, SAME1
1162 1473      ASAME2, SAME2
1163 1401      ARANDY, RANDY
1164 1435      ABNRY1, BNDRY1
1165 0510      XRETHR, RETHR+4
1166 0745      AADDRS, ADDR5
1167 0746      AINSTR, INSTR

```

```

1170 0747      AREFAD, REFAO
1171 0751      ADATA, DATATH
1172 0752      ADATAH, DATAHR
1173 0753      AMQDAT, MQDATA
1174 0755      LNKSVA, SAVLNK
1175 0625      ANTIND, NOTIND
1176 0601      ALOPID, LOOPID
1177 0000      /
/
1200 1200      *1200
/
1200 5347      JMP BCHNG
/GO ULTER
/
1201 4755      ANDTST, JMS I TSTPC
1202 1775      TAD I BDATTH
1203 0776      AND I BDATHR
1204 7041      CIA
1205 1772      TAD I DATFN
1206 7640      COMPAR, SEA CLA
1207 5313      JMP ERROR
1210 1760      TAD I LINKSV
1211 7041      CIA
1212 1761      TAD I LINKRT
1213 7640      SEA CLA
1214 5313      JMP ERROR
1215 1763      TAD I MQDAT
1216 7041      CIA
1217 1762      TAD I DONEMQ
1220 7640      SEA CLA
1221 5313      JMP ERROR
1222 5765      JMP I BLPCNT
/
1223 4755      TADTST, JMS I TSTPC
1224 7340      CLA CLL CMA
1225 0775      AND I BDATTH
1226 1776      TAD I BDATHR
1227 7041      CIA
1230 1772      TAD I DATFN
1231 7640      SEA CLA
1232 5313      JMP ERROR
1233 5765      JMP I BLPCNT
/
1234 7301      ISZTST, CLA CLL IAC
1235 1775      TAD I BDATTH
1236 7650      SNA CLA
1237 7001      IAC
1240 4755      JMS I TSTPC
1241 1776      TAD I BDATHR
1242 7041      CIA
1243 1772      TAD I DATFN
1244 7640      SEA CLA
1245 5313      JMP ERROR
1246 1774      TAD I BREFAD
1247 3756      DCA I XBSAVA
/SHOULD THE ISZ SKIP
/YES
/CHECK FOR CORRECT PC
/
/DID AC CHANGE ON ISZ
/AC FAILED ON ISZ
/GET INCREMENTED DATA WORD

```

```

1250 4757 JMS I XGETWD
1251 7041 CIA
1252 7001 IAC
1253 1775 TAD I BDATTH
1254 5206 JMP COMPAR /DID ISE WORK

1255 4755 DCATST, JMS I TSTPC /CHECK PC FROM RETURN
1256 1774 TAD I BREFAD
1257 3756 DCA I XBSAVA
1260 4757 JMS I XGETWD
1261 7041 CIA
1262 1776 TAD I BDATHR
1263 7640 SEA CLA /DID DCA WORK
1264 5313 JMP ERROR /DCA FAILED
1265 1772 TAD I DATFN /DID AC CLEAR ON DCA
1266 5206 JMP COMPAR /??

1267 4755 JMSTST, JMS I TSTPC /CHECK PC FROM RETURN
1270 1776 TAD I BDATHR
1271 7041 CIA
1272 1772 TAD I DATFN
1273 7640 SEA CLA /DID JMS CHANGE AC
1274 5313 JMP ERROR /JMS CHANGED AC
1275 1774 TAD I BREFAD
1276 3756 DCA I XBSAVA
1277 4757 JMS I XGETWD
1300 7041 CIA
1301 7001 IAC
1302 1766 TAD I BADDRS
1303 5206 JMP COMPAR /DID JMS WORK

1304 4755 JMPTST, JMS I TSTPC /CHECK PC FROM RETURN
1305 1776 TAD I BDATHR
1306 7041 CIA
1307 1772 TAD I DATFN
1310 5206 JMP COMPAR /DID JMP AFFECT THE AC

1311 0000 PCSAVE, 0
1312 0200 C200, 200

1313 5342 ERROR, JMP LOOPSW /CHECK SR0 TO INHIBIT ERROR HALT
1314 1764 TAD I XRNFLD
1315 4336 JMS HALT /FIELD THAT INSTRUCTION WAS PUT IN
1316 1773 TAD I XRETFL
1317 4336 JMS HALT /PROGRAM RETURNED FROM THIS FIELD
1320 1311 TAD PCSAVE
1321 4336 JMS HALT /EXPECTED PC RETURN
1322 1754 TAD I RETURN
1323 4336 JMS HALT /ACTUAL PC RETURN
1324 1766 TAD I BADDRS
1325 4336 JMS HALT /INSTRUCTION ADDRESS
1326 1767 TAD I FINSTR
1327 4336 JMS HALT /INSTRUCTION
1330 1767 TAD I FINSTR
    
```

```

1331 0312 AND C200
1332 7640 SEA CLA
1333 1770 ERRPSR, TAD I ZASAVA
1334 1771 TAD I ZASAVB
1335 5753 JMP I XERROR /GET REST OF ERROR INFORMATION

1336 0000 HALT, 0
1337 7402 HLT /ERROR INFORMATION IN AC
1340 7200 CLA
1341 5736 JMP I HALT

/
/INHIBIT ERROR HALT IF SR0 IS SET TO A ONE

1342 1751 LOOPSW, TAD I SWRSV/LAS /CHECK THE SWITCH REGISTER
1343 7700 SMA CLA /IS IT SET
1344 5314 JMP ERROR+1 /NO, GO HALT ON ERROR WITH INFO IN AC
1345 5752 JMP I ERRRET /GO CHECK LOOP ON INSTRUCTION SWITCH

/
*1347
/
1347 4750 BCHNG, JMS I BRERNG
/
1350 0202 BRERNG, CHANGE
1351 0336 SWRSV, SAVSWR
1352 0252 ERRRET, LPCNT
1353 1415 XERROR, ERROR1
1354 0504 RETURN, RETHR
1355 0073 TSTPC, PCTST
1356 2145 XBSAVA, BSAVA
1357 2112 XGETWD, GETWD
1360 0755 LINKSV, SAVLNK
1361 2751 LINKRT, LINKON
1362 2753 DONEHQ, MQDONE
1363 0753 MQDAT, MQDATA
1364 0754 XRNFLD, RANFLD
1365 0252 BLPCNT, LPCNT
1366 0745 BADDRS, ADDR5
1367 0746 FINSTR, INSTR
1370 1146 ZASAVA, ASAVA
1371 1147 ZASAVB, ASAVB
1372 0527 DATFN, FILALL
1373 0525 XRETFL, RETFLD
1374 0747 BREFAD, REPAD
1375 0751 BDATTH, DATAH
1376 0752 BDATHR, DATAHR
1377 0000 0

/
1400 *1400
/
1400 5362 JMP CCHNG
/
1401 0000 RANDY, 0
1402 7301 CLA CLL IAC
1403 1343 TAD RAN1
    
```

```

1404 1344 TAD RAN2
1405 7106 CLL RTL
1406 3343 DCA RAN1
1407 1344 TAD RAN2
1410 7012 RTR
1411 1343 TAD RAN1
1412 3344 DCA RAN2
1413 1344 RANDY1, TAD RAN2
1414 5601 JMP I RANDY
    
```

```

1415 3774 ERROR1, DCA I CREFAD
1416 1774 TAD I CREFAD
1417 4765 JMS I XHALT /REFERENCE ADDRESS
1420 1771 TAD I ZINDAD /INDIRECT ADDRESS IF ANY
1421 4765 JMS I XHALT
1422 1767 TAD I CDATAT /INITIAL MEMORY DATA
1423 4765 JMS I XHALT
1424 1766 TAD I ZINSTR
1425 0347 AND C400
1426 7650 SNA CLA
1427 5232 JMP ,+3
1430 1771 TAD I ZINDAD
1431 3774 DCA I CREFAD
1432 1774 TAD I CREFAD
1433 3770 DCA I ZBSAVA
1434 5772 JMP I XERR2 /GO GET REST OF INFORMATION
    
```

/THIS SECTION OF THE SUBROUTINE CHECKS FOR ILLEGAL ADDRESSES WHICH
/ARE AS FOLLOWS:0000 = 0006 AND UPPER TEST AREA LIMIT,+1 AND -2.

```

1435 0000 BNDRY1, 0
1436 3354 DCA CSAVB
1437 1354 TAD CSAVB /GET THE NUMBER
1440 1345 TAD MM7 /SUBTRACT 7 FROM IT
1441 7100 CLL /CLEAR OUT THE LINK
1442 1351 TAD BNDCON /ADD IN BOUNDRY CONSTANT=6012,4012,2012,0012
1443 7630 SZL CLA
1444 5635 JMP I BNDRY1 /ILLEGAL ADDRESS, RETURN TO RANDOM NUMBER GENERATOR
    
```

/THIS SECTION OF SUBROUTINE CHECKS FOR ILLEGAL ADDRESS WHICH ARE
/THE PROGRAM AREA-3 TO PROGRAM END +1

```

1445 7346 BNDOK1, CLA CLL CMA RTL
1446 1376 TAD PRGBC
1447 7041 CIA
1450 1354 TAD CSAVB
1451 7510 SPA
1452 5257 JMP BNDOK2
1453 7161 CIA STL
1454 1352 TAD PRGSIZ
1455 7620 SNL CLA
1456 5635 JMP I BNDRY1
1457 2235 BNDOK2, ISZ BNDRY1
    
```

```

1460 7340 CLA CLL CMA
1461 0354 AND CSAVB
1462 5635 JMP I BNDRY1
    
```

```

1463 0000 / SAME1, 0
1464 3355 DCA CSAVC
1465 1775 TAD I CADDRS
1466 3353 DCA CSAVA
1467 4303 JMS TSAME
1470 2263 ISZ SAME1
1471 1355 TAD CSAVC
1472 5663 JMP I SAME1
    
```

```

1473 0000 / SAME2, 0
1474 3355 DCA CSAVC
1475 1774 TAD I CREFAD
1476 3353 DCA CSAVA
1477 4303 JMS TSAME
1500 2273 ISZ SAME2
1501 1355 TAD CSAVC
1502 5673 JMP I SAME2
    
```

```

1503 0000 / TSAME, 0
1504 7344 CLA CLL CMA RAL
1505 1355 TAD CSAVC
1506 7041 CIA
1507 1353 TAD CSAVA
1510 7510 SPA
1511 5320 JMP INSOK
1512 7161 CIA STL
1513 1356 TAD C5
1514 7620 SNL CLA
1515 2303 ISZ TSAME
1516 7420 SNL
1517 2303 ISZ TSAME
1520 7300 INSOK, CLA CLL
1521 5703 JMP I TSAME
    
```

```

1522 0000 / LIMITS, 0
1523 1021 TAD OP1SEL /GET MEMORY SIZE FROM HARDWARE CONFIGURATION
1524 0346 AND K37 /MASK OFF MEMORY BITS
1525 7104 CLL RAL
1526 3350 DCA UPRLIM
1527 1350 TAD UPRLIM
1530 0360 AND C70
1531 3773 DCA I XFLDLM
1532 1350 TAD UPRLIM
1533 0357 AND C7
1534 7112 CLL RTR
1535 7012 RTR
1536 1361 TAD C1777
1537 3350 DCA UPRLIM
1540 1020 TAD SWITCH
1541 3764 DCA I SAVE$H
    
```

```

1542 5722          JMP I LIMITS
/
1543 1234        RAN1, 1234
1544 5670        RAN2, 5670
1545 7771        MM7, 7
1546 0037        K37, 37
1547 0400        C400, 400
1550 0000        UPRLIM, 0
1551 0000        BNDCON, 0
1552 3402        PRGSIZ, PRGEND+4-BGN
1553 0000        CSAVA, 0
1554 0000        CSAVB, 0
1555 0000        CSAVC, 0
1556 0005        C5, 0005
1557 0007        C7, 7
1560 0070        C70, 70
1561 1777        C1777, 1777
/
1562 1562        *1562
COHNG, 4763      JMS I CRERNG          /ROUTINE TO ULTER
/
1563 0202        CRERNG, CHANGE
1564 0336        SAVESW, SAVSWR
1565 1336        XHALT, HALT
1566 0746        EINST, INSTR
1567 0751        COATAT, DATATH
1570 2145        BBSAVA, BSAVA
1571 0750        EINDAD, INJAD
1572 0724        XERR2, ERROR2
1573 1144        XFLDLM, FLJLIM
1574 0747        CREFAD, REFAD
1575 0745        CADDRS, ADDR5
1576 0200        PRGBG, BGN
1577 0000        /
/

/RANDOM OPERATES=GROUP 1 = GROUP2 = AND MQ OPERATES

1600          *1600
/
1600 5347      JMP          PCHNG
/

1601 0000      INSGEN, 0          /ROUTINE TO GENERATE A RANDOM INSTRUCTION
1602 4755      JMS I BRANDY      /GO GENERATE A RANDOM NUMBER
1603 0242      AND INSHSK      /MASK WORD FOR INSTRUCTION
1604 1243      TAD CONINS      /CONSTRAINT WORD FOR INSTRUCTION
1605 3754      DCA I EINST      /SAVE THE INSTRUCTION
1606 6201      CDF 00          /CHANGE DATA FIELD TO FIELD 0
1607 6224      RIF            /READ THE INSTRUCTION FIELD
1610 1356      TAD START      /GET THE STARTING ADDRESS
1611 3612      DCA I ADDR5     /PUT FIELD AND STARTING ADDRESS INTO LOC 5
    
```

```

1612 0005      ADDR5, STRFLD      /ADDRESS 5 OF FIELD 0 = STARTING ADDRESS AND PRG FIELD
1613 6224      RIF            /READ THE INSTRUCTION FIELD
1614 1206      TAD ADDR5-4      /GET THE CDF INSTRUCTION
1615 3216      DCA ,+1          /PUT CDF TO PROGRAM FIELD IN NEXT LOCATION
1616 7402      HLT/CDF         /CHANGE DF BACK TO PROGRAM FIELD
1617 1754      TAD I EINST      /CHECK TO SEE IF IT WAS A IOT
1620 0244      AND K7000
1621 1245      TAD M6000
1622 7450      SNA
1623 5202      JMP INSGEN+1     /IT WAS A IOT REGENERATE A NEW INSTRUCTION
1624 1244      TAD K7000       /IS IT AN OPERATE INSTRUCTION
1625 7650      SNA CLA
1626 5256      JMP OPRBGN      /YES IT WAS AN OPERATE
1627 1351      TAD MRIPNT      /GET THE RETURN POINTER FOR MRI INSTRUCTIONS
1630 3753      DCA I EJMSTR     /SAVE IT
1631 1754      TAD I EINST      /NOT A IOT OR OPERATE
1632 0246      AND K177       /CREATE A REFERENCE ADDRESS
1633 3761      DCA I AASAVB
1634 1754      TAD I EINST      /GET THE INSTR
1635 0247      AND A200        /PAGE ZERO OR SAME PAGE
1636 7640      SZA CLA
1637 1760      TAD I AASAVA
1640 1761      TAD I AASAVB
1641 5601      JMP I INSGEN     /RETURN AND CHECK IT

1642 7777      INSHSK, 7777
1643 0000      CONINS, 0
1644 7000      K7000, 7000
1645 2000      M6000, -6000
1646 0177      K177, 177
1647 0200      A200, 200
1650 0400      B400, 400
1651 0014      A14, 14
1652 7764      NEG14, -14
1653 0001      A1, 1
1654 0006      BP6, 6
1655 7721      K7721, 7721

1656 1352      OPRBGN, TAD OPRPNT /GET THE RETURN POINTER FOR OPR INSTRUCTIONS
1657 3753      DCA I EJMSTR     /SAVE IT
1660 1754      TAD I EINST
1661 0250      AND B400
1662 7640      SZA CLA
1663 5272      JMP ILL0P2
1664 1754      ILL0P1, TAD I EINST /OP1-CHECK BITS 8 AND 9 TO BE ON A ONE
1665 0251      AND A14
1666 1252      TAD NEG14
1667 7650      SNA CLA
1670 5202      JMP INSGEN+1     /ILLEGAL-REGENERATE A NEW INSTRUCTION
1671 5306      JMP ILLM0+3     /GO SETUP RANDOM AC AND MQ DATA

1672 1754      ILL0P2, TAD I EINST /IS THE INSTR A MQ OR OP2 INSTR
1673 0253      AND A1
1674 7640      SZA CLA
1675 5303      JMP ILLM0       /INSTR IS A MQ INSTR CHECK FOR ILLEGAL INSTR
    
```

```

1676 1754 TAD I EINSTR /IS THE INSTR A OSR OR HLT
1677 0254 AND BP6
1700 7440 SEA
1701 5202 JMP INQGEN+1 /INSTR IS A OSR, OR HLT REGENERATE
1702 5306 JMP ILLMQ+3 /GO SET UP SIMULATED AC DATA AND MQ

1703 1754 ILLMQ, TAD I EINSTR /GET THE INSTRUCTION
1704 0255 AND K7721 /MASK OUT FOR LEGAL MQ INSTRUCTIONS
1705 3754 DCA I EINSTR /AND SAVE IT

1706 1762 TAD I XDATAH
1707 3763 DCA I XSIMAC /PUT INITIAL WORD IN SIMULATED AC
1710 1764 TAD I XSVLNK
1711 3765 DCA I XSMLNK /PUT INITIAL LINK IN SIMULATED LINK
1712 1787 TAD I INTMOD /GET THE RANDOM MQ DATA
1713 3766 DCA I XSIMMQ /PUT INITIAL MQ DATA IN SIMULATED MQ
1714 7326 CLA CLL CML RTL /SET UP INSTRUCTION RETURN POINTER
1715 1773 TAD I OADDRS /GET THE INSTRUCTION ADDRESS AND ADD 2
1716 3767 DCA I XEXPRT /SET UP EXPECTED RETURN UNLESS A SKIP
1717 6214 RDF /READ THE DATA FIELD
1720 1333 TAD 06201 /ADD IN THE COP INSTRUCTION
1721 3774 DCA I XRTOPF /SET UP A LOC TO RETURN TO OWN DATA FIELD
1722 1754 TAD I EINSTR /IS THE INSTRUCTION A OP1 OR OP2
1723 0250 AND B400
1724 7650 SNA CLA
1725 5770 JMP I XSMOP1 /OP1 GO SIMULATE THE INSTRUCTION
1726 1754 TAD I EINSTR /IS THE INSTR A MQ INSTR
1727 0253 AND A1
1730 7650 SNA CLA
1731 5771 JMP I XSMOP2 /OP2- GO SIMULATE THE INSTRUCTION
1732 5772 JMP I XSMQI /MQ- GO SIMULATE THE MQ INSTR

1733 6201 06201, 6201

/
1734 1766 OPERR1, TAD I XSIMMQ /GET THE SIMULATED MQ
1735 4341 JMS HLTOPR
1736 1775 TAD I GMQDON /GET THE FINAL MQ
1737 4341 JMS HLTOPR
1740 5776 JMP I GLPSW0 /GO LOOK AT SR0 TO LOOP ON INSTR

/
1741 0000 HLTOPR, 0
1742 7402 HLT
1743 7300 CLA CLL
1744 5741 JMP I HLTOPR

/
1747 *1747

/
1747 4750 FCHNG, JMS I FRERNG

/
1750 0202 FRERNG, CHANGE
1751 0504 MRIPNT, RETHR
    
```

```

1752 2675 OPRPNT, OPRRET
1753 0776 ZJMSRT, JMSRET
1754 0746 EINSTR, INSTR
1755 1401 BRANDY, RANDY
1756 0200 START, BGV
1757 0753 INTMOD, MQDATA
1760 1146 AASAVA, ASAVA
1761 1147 AASAVB, ASAVB
1762 0752 XDATAH, DATAHR
1763 2755 XSIMAC, SIMAC
1764 0755 XSVLNK, SAVLNK
1765 2756 XSMLNK, SIMLNK
1766 2757 XSIMMQ, SIMMQ
1767 2754 XEXPRT, EXPRET
1770 0001 XSMOP1, SIMOP1
1771 2201 XSMOP2, SIMOP2
1772 2252 XSMQI, SIMQI
1773 0745 OADDRS, ADDR5
1774 2705 XRTOPF, RETTOP
1775 2753 GMQDON, MQDONE
1776 2737 GLPSW0, LPSW0
1777 0000 0

/
2000 *2000

2000 5347 JMP GCHNG

/BEGINNING OF OPERATE GROUP ONE SIMULATION

2001 1762 SIMOP1, TAD I CINSTR /GET THE INSTRUCTION
2002 0271 AND POS200 /IS BIT 4 SET TO CLEAR THE AC
2003 7640 SEA CLA
2004 3773 DCA I OSIMAC /YES, CLEAR OUT THE SIMULATED AC
2005 1762 TAD I CINSTR /GET THE INSTRUCTION
2006 0267 AND K100 /IS BIT 5 SET TO CLEAR THE LINK
2007 7640 SEA CLA
2010 3774 DCA I OSMLNK /YES, CLEAR THE SIMULATED LINK
2011 1762 TAD I CINSTR /GET THE INSTRUCTION
2012 0266 AND K40 /IS BIT 6 SET TO COMPLEMENT THE AC
2013 7640 SEA CLA
2014 4763 JMS I XSMCMA /YES GO SIMULATE A CMA
2015 1762 TAD I CINSTR /GET THE INSTR
2016 0265 AND K20 /IS BIT 7 SET TO COMPLEMENT THE LINK
2017 7640 SEA CLA
2020 4764 JMS I XSMCML /YES, GO SIMULATE A CML
2021 1762 TAD I CINSTR /GET THE INSTRUCTION
2022 0261 AND K1 /IS BIT 11 SET TO INCREMENT THE AC
2023 7640 SEA CLA
2024 4765 JMS I XSMIAC /YES GO SIMULATE IAC
2025 1762 TAD I CINSTR /GET THE INSTRUCTION
2026 0262 AND K2 /IS BIT 10 SET TO RTR OR RTL
2027 7640 SEA CLA
    
```

```

2030 5242      JMP      SIMTWC /YES GO CHECK TO SEE WHICH ONE
2031 1762      TAD I   CINSTR /GET THE INSTRUCTION
2032 0264      AND    K14  /IS IT A ROTATE LEFT OR RIGHT
2033 1272      TAD    NEG10 /RAR?
2034 7450      SNA
2035 4766      JMS I   XSMRAR /YES GO SIMULATE A ROTATE RIGHT
2036 1263      TAD    K4    /NO,RAL?
2037 7690      SNA    CLA
2040 4767      JMS I   XSMRAL /YES,GO SIMULATE A ROTATE LEFT
2041 5254      JMP      OPRSET /GO TEST THE INSTRUCTION

2042 1762      SIMTWC; TAD I CINSTR /GET THE INSTRUCTION
2043 0264      AND    K14  /BIT 8 AND 9 = 0
2044 7450      SNA
2045 4770      JMS I   XSMBSW /YES,GO SIMULATE A BYTE SWAP
2046 1272      TAD    NEG10 /RTR?
2047 7450      SNA
2050 4771      JMS I   XSMRTR /YES, GO SIMULATE A ROTATE TWICE RIGHT
2051 1263      TAD    K4    /RTL?
2052 7650      SNA    CLA
2053 4772      JMS I   XSMRTL /YES,GO SIMULATE A ROTATE TWICE LEFT

2054 1776      OPRSET; TAD I OFIELD
2055 1270      TAD    D6201
2056 3257      DCA    ,+1
2057 7402      HLT/CDP /CHANGE TO THE RANDOM DATA FIELD
2060 5775      JMP I   INTOPR /GO SETUP THE OPERATE INSTRUCTION

2061 0001      K1,    1
2062 0002      K2,    2
2063 0004      K4,    4
2064 0014      K14,   14
2065 0020      K20,   20
2066 0040      K40,   40
2067 0100      K100,  100
2070 6201      D6201,  6201
2071 0200      POS200, 200
2072 7770      NEG10, -10

/
2073 0000      PCTST, 0
2074 7001      IAC
2075 1754      TAD I   XJMSLC
2076 3755      DCA I   XPCSAV
2077 1755      TAD I   XPCSAV
2100 7041      CIA
2101 1756      TAD I   XRETPC
2102 7640      SZA    CLA
2103 5761      JMP I   MRIERR
2104 1757      TAD I   FLOXRN
2105 7041      CIA
2106 1760      TAD I   FLXRET
2107 7640      SZA    CLA
2110 5761      JMP I   MRIERR
2111 5673      JMP I   PCTST

```

```

2112 0000      /
2113 1757      GETWD, 0
2114 1344      TAD I   FLDXRN
2115 3316      DCA    A6201
2116 7402      HLT/CDP ,+1
2117 1745      TAD I   BSAVA
2120 3345      DCA    BSAVA
2121 6224      RIF
2122 1344      TAD    A6201
2123 3324      DCA    ,+1
2124 7402      HLT/CDP
2125 1345      TAD    BSAVA
2126 5712      JMP I   GETWD

/
2127 0000      RANCON, 0
2130 1752      TAD I   ZCNFLG
2131 7650      SNA    CLA
2132 5727      JMP I   RANCON
2133 1753      TAD I   XWDMOV
2134 7402      HLT
2135 7604      LAS
2136 5727      JMP I   RANCON

/
2137 0000      INTSET, 0
2140 1351      TAD    XINT
2141 3743      DCA I   TINT
2142 5737      JMP I   INTSET
2143 0003      TINT,  INT
/

2144 6201      A6201,  6201
2145 0000      BSAVA,  0
/

/
2147 4750      *2147
GCHNG, JMS I GRERNG

/
2150 0202      GRERNG, CHANGE
2151 3102      XINT,  INTERS
2152 1143      ZCNFLG, COVFLG
2153 3573      XWDMOV, MOVWDX
2154 0762      XJMSLC, JMSLOC
2155 1311      XPCSAV, PCSAVE
2156 0504      XRETPC, RETHR
2157 0754      FLDXRN, RANFLD
2160 0525      FLXRET, RETFLD
2161 1313      MRIERR, ERROR
2162 0746      CINSTR, INSTR
2163 2401      XSMCHA, SIMCHA
2164 2426      XSMCHL, SIMCHL

```



```

2165 2435 XSMIAC, SIMIAC
2166 2442 XSMRAR, SIMRAR
2167 2461 XSMRAL, SIMRAL
2170 2477 XSMBSW, SIMBSW
2171 2520 XSHRTR, SIMRTR
2172 2537 XSHRTL, SIMRTL
2173 2755 OSIMAC, SIMAC
2174 2756 OSMLNK, SIMLNK
2175 0637 INTOPR, OPRINT
2176 0754 OFIELD, RANFLD
2177 0000 0

2200 *2200
2200 5343 JMP HCHNG
    
```

/BEGINNING OF OPERATE GROUP 2 SIMULATION

```

2201 3251 SIMOP2, DCA SKPFLG /CLEAR THE SKIP FLAG
2202 1745 SMACHK, TAD I DINSTR
2203 0243 AND Z100
2204 7650 SNA CLA
2205 5211 JMP SZACHK
2206 4746 JMS I XSMSMA
2207 5211 JMP SZACHK
2210 5226 JMP SETSKP
2211 1745 SZACHK, TAD I DINSTR
2212 0244 AND Z40
2213 7650 SNA CLA
2214 5220 JMP SNLCHK
2215 4747 JMS I XSMSZA
2216 5220 JMP SNLCHK
2217 5226 JMP SETSKP
2220 1745 SNLCHK, TAD I DINSTR
2221 0245 AND Z20
2222 7650 SNA CLA
2223 5227 JMP COMCHK
2224 4750 JMS I XSMSNL
2225 5227 JMP COMCHK
2226 2251 SETSKP, ISE SKPFLG
2227 1745 COMCHK, TAD I DINSTR
2230 0250 AND POS10
2231 7640 SZA CLA
2232 7240 CLA CMA
2233 1251 TAD SKPFLG
2234 7640 SZA CLA
2235 2751 ISE I ZEXPRT
2236 1745 TAD I DINSTR
2237 0246 AND Z200
2240 7640 SZA CLA
2241 3752 DCA I XACSIM
2242 5753 JMP I ZSETOP /GO SETUP AND TEST INSTR
    
```

```

2243 0100 Z100, 100
2244 0040 Z40, 40
2245 0020 Z20, 20
2246 0200 Z200, 200
2247 0320 Z320, 320
2250 0010 POS10, 10
2251 0000 SKPFLG, 0
    
```

/BEGINNING OF OPERATE GROUP 2 MQ INSTRUCTION SIMULATION

```

2252 1745 SIMMQI, TAD I DINSTR /GET THE INSTRUCTION
2253 0247 AND Z320 /MASK OUT FOR LEGAL BITS 4,5 &7
2254 7450 SNA
2255 5753 JMP I ZSETOP /INSTRUCTION IS A NOP
2256 1304 TAD NEG20 /SUBTRACT 20
2257 7450 SNA
2260 5754 JMP I XSHMQL /GO SIMULATE A MQL
2261 1305 TAD M60
2262 7450 SNA
2263 5755 JMP I XSHMQA /GO SIMULATE A MQA
2264 1304 TAD NEG20
2265 7450 SNA
2266 5756 JMP I XSHSWP /GO SIMULATE A SWP
2267 1305 TAD M60
2270 7450 SNA
2271 5757 JMP I XSMCLA /GO SIMULATE A CLA
2272 1304 TAD NEG20
2273 7450 SNA
2274 5760 JMP I XSMCAM /GO SIMULATE A CAM
2275 1305 TAD M60
2276 7450 SNA
2277 5761 JMP I XSMACL /GO SIMULATE A ACL
2300 1304 TAD NEG20
2301 7650 SNA CLA
2302 5762 JMP I XCLSWP /GO SIMULATE A SWP,CLA
2303 7402 HLT /NONE OF THE ABOVE
2304 7760 NEG20, =20
2305 7720 M60, =60

2306 1763 /
2307 4764 ERROPR, TAD I GRANFL /GET THE RANDOM DATA FIELD
2310 1765 JMS I OPRHLT
2311 4764 TAD I GOPRET /GET THE ACTUAL RETURN FIELD
2312 1751 JMS I OPRHLT
2313 4764 TAD I ZEXPRT /GET THE EXPECTED RETURN PC
2314 1766 TAD I GACTRT /GET THE ACTUAL RETURN PC
2315 4764 JMS I OPRHLT
2316 1767 TAD I GADDRS /GET THE INSTRUCTION ADDRESS
2317 4764 JMS I OPRHLT
2320 1745 TAD I DINSTR /GET THE INSTRUCTION
2321 4764 JMS I OPRHLT
2322 1770 TAD I GOATAH /GET THE INITIAL AC DATA
2323 4764 JMS I OPRHLT
2324 1752 TAD I XACSIM /GET THE SIMULATED AC
    
```

```

2325 4764 JMS I OPRHLT
2326 1771 TAD I GDATAD /GET THE FINAL AC
2327 4764 JMS I OPRHLT
2330 1772 TAD I GSVLNK /GET THE INITIAL LINK
2331 4764 JMS I OPRHLT
2332 1773 TAD I GSMLNK /GET THE SIMULATED LINK
2333 4764 JMS I OPRHLT
2334 1774 TAD I GLNKDN /GET THE FINAL LINK
2335 4764 JMS I OPRHLT
2336 1775 TAD I GMDAT /GET THE INITIAL MQ DATA
2337 4764 JMS I OPRHLT
2340 5776 JMP I OPRER1

```

```

/
2343 *2343
/
2343 4744 HCHNG, JMS I HRERNG
/
2344 0202 HRERNG, CHANGE
2345 0746 DINSTR, INSTR
2346 2611 XSMSMA, SIMSMA
2347 2601 XSMSZA, SIMSZA
2350 2626 XSMSNL, SIMSNL
2351 2754 ZEXPRT, EXPRET
2352 2755 XACSIM, SIMAC
2353 2054 ZSETOP, OPRSET
2354 2640 XSMML, SIMML
2355 2644 XSMQA, SIMQA
2356 2652 XSMWP, SIMSWP
2357 2661 XSMCLA, SIMCLA
2360 2663 XSMCAM, SIMCAM
2361 2666 XSMACL, SIMACL
2362 2671 XCLSWP, CLASWP
2363 0754 GRANFL, RANFLD
2364 1741 OPRHLT, HLTOPR
2365 2752 GOPRET, OPRETF
2366 2675 GACTRT, OPRRET
2367 0745 GADDRS, ADDR
2370 0752 GDATAH, DATAHR
2371 2750 GDATAD, DATAON
2372 0755 GSVLNK, SAVLNK
2373 2756 GSMLNK, SIMLNK
2374 2751 GLNKDN, LINKDN
2375 0753 GMDAT, MQDATA
2376 1734 OPRER1, OPERR1
2377 0000 0

```

```

2400 *2400
/
2400 5367 JMP ICHNG
/

```

/ROUTINE TO SIMULATE A COMPLEMENT

```

2401 0000 SIMCHA, 0
2402 1224 TAD M14
2403 3225 DCA CNT
2404 3226 DCA SIMCML
2405 1775 TAD I BSIMAC
2406 7104 CLL RAL
2407 3235 DCA SIMIAC
2410 7420 SNL
2411 2226 ISE SIMCML
2411 1226 TAD SIMCML
2413 2225 ISE CNT
2414 5216 JMP ,=2
2415 5222 JMP ENDCMA
2416 7104 CLL RAL
2417 3226 DCA SIMCML
2420 1235 TAD SIMIAC
2421 5206 JMP SIMCMA+5
2422 3775 ENDCMA, DCA I BSIMAC
2423 5601 JMP I SIMCMA

```

```

2424 7764 M14, =14
2425 0000 CNT, 0
/ROUTINE TO SIMULATE A CML

```

```

2426 0000 SIMCML, 0
2427 1776 TAD I BSMLNK /GET THE SIMULATED LINK
2430 7650 SNA CLA /IS IT A 0
2431 1234 TAD K4000 /YES, MAKE IT A ONE
2432 3776 DCA I BSMLNK /SAVE IT
2433 5626 JMP I SIMCML

```

```

2434 4000 K4000, 4000
/ROUTINE TO SIMULATE A IAC

```

```

2435 0000 SIMIAC, 0
2436 2775 ISE I BSIMAC /BUMP THE SIMULATED AC
2437 5241 JMP ,=2
2440 4226 JMS SIMCML
2441 5635 JMP I SIMIAC

```

/ROUTINE TO SIMULATE A RAR

```

2442 0000 SIMRAR, 0
2443 7300 CLA CLL /CLEAR OUT A LINK AND THE AC
2444 1776 TAD I BSMLNK /GET THE SIMULATED LINK
2445 7004 RAL /PUT IT IN THE LINK
2446 1260 TAD NEG6
2447 3225 DCA CNT
2450 1775 TAD I BSIMAC /GET THE SIMULATED AC
2451 7006 RTL /ROTATE 12 PLACES TO THE LEFT
2452 2225 ISE CNT
2453 5251 JMP ,=2
2454 3775 DCA I BSIMAC /SAVE THE SIMULATED ROTATE

```

```

2455 7010      RAR      /GET THE LINK
2456 3776      DCA I BSMLNK /SAVE THE LINK
2457 5642      JMP I SIMRAR /RETURN

2460 7772      NEG6,   =6

/ROUTINE TO SIMULATE A RAL

2461 0000      SIMRAL, 0
2462 7300      CLA      CLL
2463 1776      TAJ I BSMLNK /GET THE SIMULATED LINK
2464 7004      RAL      /PUT IT IN THE LINK
2465 1260      TAJ      NEG6
2466 3225      DCA      CNT
2467 1775      TAJ I BSIMAC /GET THE SIMULATED AC
2470 7012      RTR      /ROTATE IT RIGHT 12 TIMES
2471 2225      ISE      CNT
2472 5270      JMP      ,-2
2473 3775      DCA I BSIMAC /SAVE THE SIMULATED ROTATE
2474 7010      RAR
2475 3776      DCA I BSMLNK /SAVE THE SIMULATED LINK
2476 5661      JMP I SIMRAL /RETURN
    
```

```

/ROUTINE TO SIMULATE A BYTE SWAP

2477 0000      SIMBSW, 0
2500 7300      CLA      CLL
2501 1776      TAJ I BSMLNK /GET THE SIMULATED LINK
2502 7010      RAR
2503 7012      RTR
2504 7012      RTR
2505 1775      TAJ I BSIMAC /GET THE SIMULATED AC
2506 0317      AND      K7700
2507 1775      TAJ I BSIMAC /GET IT AGAIN
2510 7006      RTL
2511 7006      RTL
2512 7006      RTL
2513 3775      DCA I BSIMAC /SAVE THE SIMULATED BYTE SWAP
2514 7010      RAR
2515 3776      DCA I BSMLNK /SAVE THE LINK
2516 5677      JMP I SIMBSW /RETURN
2517 7700      K7700, 7700
    
```

```

/ROUTINE TO SIMULATE RTR

2520 0000      SIMRTR, 0
2521 7300      CLA      CLL
2522 1776      TAJ I BSMLNK /GET THE SIMULATED LINK
2523 7004      RAL      /PUT IT IN THE LINK
2524 1336      TAJ      M13
2525 3225      DCA      CNT /SETUP A COUNTER FOR 11 RAL'S
2526 1775      TAJ I BSIMAC /GET THE SIMULATED AC
2527 7004      RAL
2530 2225      ISE      CNT
    
```

```

2531 5327      JMP      ,-2
2532 3775      DCA I BSIMAC /SAVE THE SIMULATED ROTATED AC
2533 7010      RAR
2534 3776      DCA I BSMLNK /SAVE THE SIMULATED LINK
2535 5720      JMP I SIMRTR /RETURN
    
```

```

2536 7765      M13,   =13
    
```

```

/ROUTINE TO SIMULATE RTL

2537 0000      SIMRTL, 0
2540 7300      CLA      CLL
2541 1776      TAJ I BSMLNK /GET THE SIMULATED LINK
2542 7004      RAL      /AND PUT IT IN THE LINK
2543 1336      TAJ      M13
2544 3225      DCA      CNT /SET UP A COUNTER TO DO 11 RAR'S
2545 1775      TAJ I BSIMAC /GET THE SIMULATED AC
2546 7010      RAR
2547 2225      ISE      CNT
2550 5346      JMP      ,-2
2551 3775      DCA I BSIMAC /SAVE THE SIMULATED ROTATED AC
2552 7010      RAR
2553 3776      DCA I BSMLNK /SAVE THE SIMULATED ROTATED LINK
2554 5737      JMP I SIMRTL
    
```

```

/
2555 1771      FLDCHK, TAJ I UPPERL /IS THE LAST FIELD = 1K
2556 1365      TAJ      M1777
2557 7640      SEA      CLA
2558 5772      JMP I TSFLDF /NO GO SWAP IT UP
2561 6224      RIF      /READ THE INSTRUCTION FIELD
2562 7640      SEA      CLA /IS IT EQUAL TO FIELD 0
2563 5773      JMP I TSWPDN /NO, GO SWAP THE PROGRAM DOWN
2564 5774      JMP I ROLFLG /YES, DO NOT SWAP FIELDS BUT ROLL UP
/
    
```

```

2565 6001      M1777, =1777
    
```

```

/
2567 4770      ICHNG, JMS I IRERNG
/
    
```

```

2570 0202      IRERNG, CHANGE
2571 1550      UPPERL, UPRLIM
2572 0433      TSFLDF, SFLODF
2573 0422      TSWPDN, SWAPDN
2574 0327      ROLFLG, SETFLG
2575 2755      BSIMAC, SIMAC
2576 2756      BSMLNK, SIMLNK
2577 0000      0
/
    
```

```

2600 *2600
/
    
```

2600 5364 JMP JCHNG

/ROUTINE TO SIMULATE A SEA

2601 0000 SIMSEA, 0
 2602 7240 CLA CMA
 2603 1770 TAD I CSIMAC
 2604 3223 DCA ACUTST
 2605 2223 ISZ ACUTST
 2606 5601 JMP I SIMSEA
 2607 2201 ISZ SIMSEA
 2610 5601 JMP I SIMSEA

/ROUTINE TO SIMULATE A SMA

2611 0000 SIMSMA, 0
 2612 1770 TAD I CSIMAC
 2613 0225 AND D4000
 2614 1224 TAD K7777
 2615 3223 DCA ACUTST
 2616 2223 ISZ ACUTST
 2617 5221 JMP ,+2
 2620 5611 JMP I SIMSMA
 2621 2211 ISZ SIMSMA
 2622 5611 JMP I SIMSMA

2623 0000 ACUTST, 0
 2624 7777 K7777, -1
 2625 4000 D4000, 4000

/ROUTINE TO SIMULATE A SNL

2626 0000 SIMSNL, 0
 2627 1771 TAD I CSMLNK
 2630 0225 AND D4000
 2631 1224 TAD K7777
 2632 3223 DCA ACUTST
 2633 2223 ISZ ACUTST
 2634 5236 JMP ,+2
 2635 5626 JMP I SIMSNL
 2636 2226 ISZ SIMSNL
 2637 5626 JMP I SIMSNL

/ROUTINE TO SIMULATE A MQL

2640 1770 SIMMQL, TAD I CSIMAC /GET THE SIMULATED AC
 2641 3772 DCA I CSIMMQ /PUT IT IN THE SIMULATED MQ
 2642 3770 DCA I CSIMAC /CLEAR OUT THE SIMULATED AC
 2643 5773 JMP I COPRST /GO EXECUTE THE INSTRUCTION

/ROUTINE TO SIMULATE MQA

2644 1772 SIMMQA, TAD I CSIMMQ /GET THE SIMULATED MQ
 2645 7040 CMA /COMPLEMENT THE RESULTS

2646 0770 AND I CSIMAC /MASK RESULTS WITH SIMULATED AC
 2647 1772 TAD I CSIMMQ /INCLUSIVE OR THE SIMULATED MQ
 2650 3770 DCA I CSIMAC /THE SIMULATED AC = INCLUSIVE OR OF MQ & AC
 2651 5773 JMP I COPRST /GO EXECUTE THE INSTR.

/ROUTINE TO SIMULATE A SWP

2652 1770 SIMSWP, TAD I CSIMAC /GET THE SIMULATED AC
 2653 3223 DCA ACUTST /AND SAVE IT
 2654 1772 TAD I CSIMMQ /GET THE SIMULATED MQ
 2655 3770 DCA I CSIMAC /AND PUT IT IN THE SIMULATED AC
 2656 1223 TAD ACUTST /GET THE SIMULATED AC
 2657 3772 DCA I CSIMMQ /AND PUT IT IN THE SIMULATED MQ
 2660 5773 JMP I COPRST /GO EXECUTE THE INSTRUCTION

/ROUTINE TO SIMULATE A CLA

2661 3770 SIMCLA, DCA I CSIMAC /CLEAR THE SIMULATED AC
 2662 5773 JMP I COPRST /GO EXECUTE THE INSTRUCTION

/ROUTINE TO SIMULATE A CAM

2663 3770 SIMCAM, DCA I CSIMAC /CLEAR THE SIMULATED AC
 2664 3772 DCA I CSIMMQ /CLEAR THE SIMULATED MQ
 2665 5773 JMP I COPRST /GO EXECUTE THE INSTRUCTION

/ROUTINE TO SIMULATE A ACL

2666 1772 SIMACL, TAD I CSIMMQ /GET THE SIMULATED MQ
 2667 3770 DCA I CSIMAC /PUT IT IN THE SIMULATED AC
 2670 5773 JMP I COPRST /GO EXECUTE THE INSTR

/ROUTINE TO SIMULATE A CLA:SWP

2671 1772 CLASWP, TAD I CSIMMQ /GET THE SIMULATED MQ
 2672 3770 DCA I CSIMAC /PUT IT IN THE SIMULATED AC
 2673 3772 DCA I CSIMMQ /CLEAR THE SIMULATED MQ
 2674 5773 JMP I COPRST /GO EXECUTE THE INSTRUCTION

/OPRRET, 0 /RETURN HERE FROM OPERATE INSTRUCTION

2674 3350 DCA DATADN /SAVE THE FINAL AC
 2677 7010 RAR
 2700 3351 DCA LINKDN /SAVE THE FINAL LINK
 2701 6214 RDP /GET THE RANDOM DATA FIELD
 2702 3352 DCA OPRETF /SAVE THE OP FROM OPERATE RETURN
 2703 7701 ACL /GET THE FINAL MQ DATA
 2704 3353 DCA MQDONE /SAVE IT
 2705 7402 HLT/COF /CHANGE DF BACK TO PROGRAM FIELD
 2706 1775 OPRQOM, TAD I XFIELD /GET THE EXPECTED INSTRUCTION FIELD
 2707 7041 CIA
 2710 1352 TAD OPRETF /GET THE FIELD INSTRUCTION RETURNED FROM
 2711 7640 SEA CLA
 2712 5344 JMP OPRERR /PROGRAM RETURNED FROM THE WRONG FIELD
 2713 1354 TAD EXPRET /GET THE EXPECTED RETURN PC

```

2714 7041 CIA
2715 1275 TAO OPRRET /GET THE ACTUAL RETURN PC
2716 7640 SEA CLA
2717 5344 JMP OPRERR /EXPECTED PC DOES NOT AGREE WITH ACTUAL
2720 1397 TAO SIMMQ /GET THE SIMULATED MQ
2721 7041 CIA
2722 1353 TAO MQDONE /GET THE ACTUAL MQ
2723 7640 SEA CLA
2724 5344 JMP OPRERR /ERROR,ACTUAL MQ DOES NOT EQUAL SIMULATED MQ
2725 1356 TAO SIMLNK /GET THE SIMULATED LINK
2726 7041 CIA
2727 1351 TAO LINKDN /GET THE ACTUAL LINK
2730 7640 SEA CLA
2731 5344 JMP OPRERR /ERROR SIMULATED AND ACTUAL LINK ARE NOT EQUAL
2732 1355 TAO SIMAC /GET THE SIMULATED AC
2733 7041 CIA
2734 1350 TAO DATAON /GET THE ACTUAL AC RETURNED
2735 7640 SEA CLA
2736 5344 JMP OPRERR /SIMULATED AND ACTUAL AC DO NOT AGREE
2737 1766 LPSW0, TAO I SRSV/LAS /IS SRI SET TO LOOP ON THE INSTRUCTION
2740 7004 RAL
2741 7710 SPA CLA
2742 5774 JMP I XINSGN /YES GO LOOP ON THE INSTRUCTIN
2743 5776 JMP I XXCNT /GO BUMP INSTRUCTION COUNTER

2744 1766 OPRERR, TAO I SRSV/LAS /CHECK SR0=1 TO INHIBIT ERROR HALT
2745 7710 SPA CLA /IS IT SET
2746 5337 JMP LPSW0 /YES, CHECK LOOP SWITCH
2747 5767 JMP I OERROR /NO, GO HALT WITH ERROR INFORMATION IN AC

2750 0000 DATADN: 0
2751 0000 LINKDN: 0
2752 0000 OPRETF: 0
2753 0000 MQDONE: 0
2754 0000 EXPRET: 0
2755 0000 SIMAC: 0
2756 0000 SIMLNK: 0
2757 0000 SIMMQ: 0
/
/
2764 *2764
/
2764 4765 JCHNG, JMS I JRNRNG
/
2765 0202 JRNRNG, CHANGE
2766 0336 SRSV, SAVSWR
2767 2306 OERROR, ERROPR
2770 2755 CSIMAC, SIMAC
2771 2756 CSMLNK, SIMLNK
2772 2757 CSIMMQ, SIMMQ
2773 2054 CPRST, OPRSET
2774 1606 XINSGN, INSGEN+5
2775 0754 XFIELD, RANFLD
2776 0271 XXCNT, XCNT
2777 0000 0
    
```

```

/
3000 *3000
/
3000 5346 JMP KCHNG
/
/ROUTINE TO FILL MEMORY WITH HALTS AROUND THE PROGRAM

3001 1771 FILRND, TAO I CONTLN /GET THE UPPER LIMIT COUNTER
3002 3225 DCA TEMP /SAVE IT
3003 1772 TAO I FLDCNT /CHECK TO SEE IF IT IS FIELD 0
3004 7690 SNA CLA /IS IT FIELD 0?
3005 1223 TAO MM4 /YES, SUBTRACT 4 FROM THE BEGINNING ADDRESS
3006 1373 TAO ABGN
3007 7041 CIA /NEGATE THE NUMBER FOR A COUNTER
3010 3771 DCA I CONTLN /SAVE IT
3011 1772 TAO I FLDCNT /CHECK TO SEE IF IT IS FIELD 0
3012 7690 SNA CLA /IS IT FIELD 0?
3013 1224 TAO KKK4 /YES, START FILLING FIELD 0 AT ADDRESS 4
3014 4774 JMS I ZFILL /FILL THE FIRST HALF OF PROGRAM FIELD
3015 1225 TAO TEMP /GET THE UPPER LIMIT COUNTER
3016 1375 TAO ENDOFP /ADD END OF PROGRAM TO IT
3017 3771 DCA I CONTLN /SAVE THIS NUMBER AS THE COUNTER
3020 1375 TAO ENDOFP /GET THE ADDRESS TO START FILLING MEMORY
3021 4774 JMS I ZFILL /WITH HALTS
3022 5776 JMP I XADD1 /RETURN FOR NEXT FIELD

3023 7774 MM4, =4
3024 0004 KKK4, 4
3025 0000 TEMP, 0

3026 7240 BGNCON, CLA CMA /CONSTRAINT STARTING ADDRESS
3027 3790 DCA I XCNFLG /RANDOM STARTING ADDRESS
3030 4751 JMS I XLIMIT /SETUP MEMORY LIMITS
3031 1021 TAO OP1SEL
3032 3234 DCA SELOP1
3033 5762 JMP I CONSET /RETURN TO PROGRAM TO SETUP MASK AND CONSTRAINT WORDS

3034 0000 SELOP1, 0

3035 0000 SETOP1, 0
3036 1234 TAO SELOP1
3037 7004 RAL /CHECK TO SEE IF OPTION 1 WAS SELECTED
3040 7700 SMA CLA /HAS IT ?
3041 5635 JMP I SETOP1 /NO, JUST RUN MRI AND OPR
3042 3301 DCA KILL
3043 4792 JMS I OPRAND /START SETTING UP OPTION 1
3044 0300 AND KK377 /MASK BITS 4-11
3045 3793 DCA I SLUXMT /SAVE THE RANDOM NUMBER FOR SLU
3046 4792 JMS I OPRAND /GENERATE A RANDOM NUMBER FOR PARALLEL I/O
3047 3794 DCA I PIOXMT /SAVE THE WORD FOR THE PARALLEL I/O
3050 7240 CLA CMA /SET ALL PROGRAM FLAGS TO INACTIVE STATE
3051 3795 DCA I FLGXMT /SLU XMIT FLAG
3052 7240 CLA CMA
3053 3796 DCA I FLGREC /SLU RECEIVE FLAG
    
```

```

3054 7240 CLA CMA
3055 3757 DCA I FLGP10 /PARALLEL I/O FLAG
3056 7240 CLA CMA
3057 3760 DCA I FLGRYC /REAL TIME CLOCK FLAG
3060 1277 TAJ MM55
3061 3761 DCA I DVINAC /SETUP A DEVICE INACTIVE COUNTER
3062 7301 CLA CLL IAC /SET DATA 11 TO A ONE
3063 6035 KIE /SET SLU INTERRUPT ENABLE
3064 6135 CLLE /SET RTC INTERRUPT ENABLE
3065 7300 CLA CLL
3066 6575 DBSE /SET PARALLEL I/O INT ENABLE
3067 1753 TAJ I SLUXMT /GET THE WORD TO BE TRANSMITTED BY SLU
3070 6046 TLS /CLEAR XMIT FLAG AND TRANSMIT WORD
3071 7200 CLA
3072 1754 TAJ I P10XMT /GET THE WORD TO BE TRANSMITTED BY PARALLEL I/O
3073 6574 DBTD /TRANSMIT IT
3074 7301 CLA CLL IAC /SET A PROGRAM FLAG TO SIGNIFY TO RTC WHEN
3075 3763 DCA I P1ORDY /TO TRANSMIT ANOTHER CHARACTER ON PARALLEL I/O
3076 5635 JMP I SETOP1 /RETURN TO PROGRAM

3077 7723 MM55, -55
3100 0377 KK377, 377
3101 0000 KILL, 0
    
```

/INTERRUPT SERVICE ROUTINE

```

3102 3341 INTERS, DCA INTAC /SAVE THE AC
3103 7010 RAR /GET THE LINK INTO BIT 0
3104 3342 DCA INTLNK /SAVE THE LINK
3105 1743 TAJ I ADDRS0 /GET THE INTERRUPT PC
3106 3344 DCA INTRET /SAVE IT
3107 6224 RIF /READ THE INSTRUCTION FIELD
3110 1345 TAJ KKCDF /ADD CDF INSTRUCTION TO BITS 6-8
3111 3312 DCA ,+1 /PUT CDF TO PROGRAM FIELD IN NEXT LOCATION
3112 7402 HLT/ODF /TO PROGRAM FIELD
3113 6041 TSF /SKIP ON SLU XMIT FLAG
3114 7410 SKP
3115 5765 JMP I SERXMT /GO SERVICE SLU XMIT FLAG
3116 6031 KSF /SKIP ON SLU RECEIVE FLAG
3117 7410 SKP
3120 5766 JMP I SERREC /GO SERVICE SLU RECEIVE FLAG
3121 6571 DBSK /SKIP ON PARALLEL I/O DATA READY FLAG
3122 7410 SKP
3123 5767 JMP I SERPIO /GO SERVICE PARALLEL I/O
3124 6137 CLSK /SKIP ON REAL TIME CLOCK FLAG
3125 7410 SKP
3126 5770 JMP I SERRTC /GO SERVICE REAL TIME CLOCK FLAG
3127 6102 SPL /SKIP ON AC LOW P/F
3130 7410 SKP
3131 5764 JMP I POWERF /POWER FAILURE GO CLEAR AC LOW AND RETURN
3132 7402 HLT /ILLEGAL INTERRUPT
3133 1342 RETPRG, TAJ INTLNK /GET THE LINK
3134 7104 CLL RAL /RESTORE IT
3135 1341 TAJ INTAC /RESTORE THE AC
3136 6244 RMP /RESTORE MEMORY FIELDS
    
```

```

3137 6001 ION /TURN THE INTERRUPT ON
3140 5744 JMP I INTRET /RETURN TO PROGRAM

3141 0000 INTAC, 0
3142 0000 INTLNK, 0
3143 0000 ADDRS0, 0
3144 0000 INTRET, 0
3145 6201 KKCDF, CDF 00

/
3146 *3146
/
3146 4747 KCHNG, JMS I KRERNG
/
3147 0202 KRERNG, CHANGE
3150 1143 XGNFLG, CONFLG
3151 1922 XLIMIT, LIMITS
3152 1401 OPRAND, RANDY
3153 3232 SLUXMT, XMTSLU
3154 3234 P10XMT, XMTPIO
3155 3226 FLGXMT, XMTFLG
3156 3227 FLGREC, RECFLG
3157 3230 FLGP10, P10FLG
3160 3231 FLGRYC, RTCFLG
3161 3337 DVINAC, INACDV
3162 3401 CONSET, SETCON
3163 3254 P1ORDY, RDPPIO
3164 3304 POWERF, POWFAL
3165 3201 SERXMT, XMTSER
3166 3204 SERREC, REC SER
3167 3237 SERPIO, P10SER
3170 3255 SERRTC, RTC SER
3171 1145 CONTLN, HGHLIM
3172 0247 FLDCNT, CNTR3
3173 0200 ABGN, BGN
3174 0527 ZFILL, FILALL
3175 3576 ENDOFP, PRSEND
3176 3436 XADD1, ADDONE
3177 0000 0

/
3200 *3200
/
3200 5367 JMP LCHNG
/
/SERIAL LINE UNIT TRANSMIT SERVICE ROUTINE

3201 3226 XMTSER, DCA XMTFLG /SET SLU XMIT FLAG ACTIVE
3202 6042 TCF /CLEAR TRANSMIT FLAG
3203 5313 JMP CHKACT /CHECK ALL DEVICES TO BE ACTIVE

/SERIAL LINE UNIT RECEIVER SERVICE ROUTINE

3204 3227 REC SER, DCA RECFLG /SET SLU RECEIVER FLAG ACTIVE
    
```

```

3205 6036      KR3      /CLEAR RECIVE FLAG AND READ THE BUFFER
3206 3233      DCA      RECFLU  /SAVE THE WORD
3207 1232      TAD      XMTSLU  /COMPARE WORD TRANSMITTED WITH WORD READ
3210 7041      CIA
3211 1233      TAD      RECFLU
3212 7640      SZA      CLA      /ARE THEY EQUAL?
3213 7402      HLT/JMP I PSRERR//DATA ERROR SLU
3214 1773      TAD I    KILLIT
3215 7640      SZA      CLA
3216 5271      JMP      OUT    /KILL INTERRUPTS UNTIL RELOCATION OCCURS
3217 4341      JMS      RANDOM  /GO GENERATE A NEW WORD
3220 0236      AND      CC377  /MASK BITS 4-11
3221 3232      DCA      XMTSLU  /SAVE WORD TO BE TRANSMITTED
3222 1232      TAD      XMTSLU  /GET THE WORD
3223 6046      TLS      /TRANSMIT IT
3224 7300      CLA      CLL
3225 5313      JMP      CHKACT  /CHECK ALL DEVICES TO BE ACTIVE

```

```

3226 0000      XMTFLG, 0
3227 0000      RECFLG, 0
3230 0000      PIOFLG, 0
3231 0000      RTCFLG, 0
3232 0000      XMTSLU, 0
3233 0000      RECFLU, 0
3234 0000      XMTPIO, 0
3235 0000      RECP10, 0
3236 0377      CC377, 377

```

/PARALLEL I/O INTERRUPT SERVICE ROUTINE

```

3237 3230      P1OSER, DCA      PIOFLG  /SET PARALLEL I/O ACTIVE FLAG
3240 6572      DBRD      /READ THE PARALLEL I/O BUFFER
3241 6573      DBCF      /CLEAR DATA READY FLAG SET DATA ACCEPTED
3242 3235      DCA      RECP10  /SAVE THE WORD READ
3243 6570      DBST      /SKIP AND CLEAR DATA ACCEPTED AND DATA AVAILABLE
3244 7402      HLT/JMP I PSRERR/DBCF FAILED TO SET DATA ACCEPTED
3245 1234      TAD      XMTPIO  /COMPARE THE WORD TRANSMITTED WITH THE WORD READ
3246 7041      CIA
3247 1235      TAD      RECP10
3250 7640      SZA      CLA      /ARE THEY EQUAL?
3251 7402      HLT/JMP I PSRERR/NO, DATA ERROR PARALLEL I/O
3252 3254      DCA      RDYPIO  /SET A PROGRAM FLAG TO SIGNIFY TO RTC, THAT
                          /A NEW WORD IS NEEDED FOR PARALLEL I/O
3253 5313      JMP      CHKACT  /CHECK ALL DEVICES TO BE ACTIVE

```

```

3254 0000      RDYPIO, 0

```

/REAL TIME CLOCK INTERRUPT SERVICE ROUTINE

```

3255 3231      RTCSER, DCA      RTCFLG  /SET REAL TIME CLOCK FLAG TO ACTIVE
3256 6136      CL3L      /CLEAR CLOCK FLAG
3257 7000      NOP/JMS I ACTLIN / THIS LOCATION USED IF ACT LINE AND OPTION 1 SELECTED
3260 1254      TAD      RDYPIO  /GET PARALLEL I/O STATUS
3261 7640      SZA      CLA      /TRANSMIT ANOTHER WORD?
3262 5313      JMP      CHKACT  /NO, MUST BE INACTIVE GO CHECK ALL DEVICES

```

```

3263 2254      ISZ      RDYPIO  /SET PROGRAM FLAG TO INACTIVE
3264 1773      TAD I    KILLIT  /GET THE KILL FLAG
3265 7650      SNA      CLA      /WAS IT SET
3266 5275      JMP      OUT+4  /NO CONTINUE RUNNING
3267 6135      CLLE      /YES = CLEAR RTC AND P I/O INT ENA
3270 6576      DB3E

```

```

3271 2303      OUT,    ISZ      CNTEND
3272 5776      JMP I    PRGRET  /WAIT FOR NEXT INTERRUPT
3273 6002      IOF      /TURN THE INTERRUPT OFF
3274 5774      JMP I    RELGO  /RETURN TO PROGRAM FOR RELOCATION OR RUN
3275 4341      JMS      RANDOM  /GENERATE A NEW WORD FOR PARALLEL I/O
3276 3234      DCA      XMTPIO  /SAVE IT
3277 1234      TAD      XMTPIO  /GET THE NEW WORD
3300 6574      DBTD      /TRANSMIT IT
3301 7300      CLA      CLL
3302 5313      JMP      CHKACT  /CHECK OTHER DEVICES TO BE ACTIVE

```

```

3303 0000      CNTEND, 0

```

/POWER FAIL INTERRUPT SERVICE ROUTINE

```

3304 6103      POWFAL, CAL      /CLEAR AC LOW P/F
3305 6102      SPL      /SKIP ON AC LOW AS A LEVEL
3306 7410      SKP
3307 6101      SBE
3310 7410      SKP
3311 7402      HLT      /BATTERY EMPTY - ITS ALL OVER
3312 5776      JMP I    PRGRET  /RETURN TO THE PROGRAM

3313 1226      CHKACT, TAD      XMTFLG  /CHECK ALL DEVICES TO BE INTERRUPTING
3314 1227      TAD      RECFLG
3315 1231      TAD      RTCFLG
3316 1230      TAD      PIOFLG
3317 7650      SNA      CLA      /ARE THEY ?
3320 5324      JMP      RESET  /YES, RESET ALL FLAGS TO INACTIVE
3321 2337      ISZ      INACDV  /BUMP INACTIVE COUNTER
3322 5776      JMP I    PRGRET  /RETURN TO THE PROGRAM
3323 7402      HLT/JMP I PSRERR/ONE OR MORE DEVICES ARE INACTIVE
3324 7340      RESET,  CLA CLL  CMA      /SET ALL DEVICES TO INACTIVE
3325 3226      DCA      XMTFLG
3326 7240      CLA      CMA
3327 3227      DCA      RECFLG
3330 7240      CLA      CMA
3331 3230      DCA      PIOFLG
3332 7240      CLA      CMA
3333 3231      DCA      RTCFLG
3334 1340      TAD      MMM55
3335 3337      DCA      INACDV  /RESET INACTIVE COUNTER
3336 5776      JMP I    PRGRET  /RETURN TO THE PROGRAM

3337 0000      INACDV, 0

```

```

3340 7723   MMM55,  =55

3341 0000   RANDOM, 0
3342 7301   CLA CLL IAC
3343 1355   TAD RANN1
3344 1356   TAD RANN2
3345 7106   CLL RTL
3346 3355   DCA RANN1
3347 1356   TAD RANN2
3350 7012   RTR
3351 1355   TAD RANN1
3352 3356   DCA RANN2
3353 1356   TAD RANN2
3354 5741   JMP I RANDOM

3355 1234   RANN1, 1234
3356 5670   RANN2, 5670

3357 0000   WAITEN, 0
3360 1775   TAD I OP1      /WAS OPTION 1 SELECTED
3361 7004   RAL
3362 7700   SMA CLA
3363 5757   JMP I WAITEN   /NO, RETURN TO RELOCATION
3364 2773   ISZ I KILLIT
3365 5365   JMP           /WAIT FOR DEVICES TO BECOME IN ACTIVE

3367   *3367

3369 4770   /
          LCHNG, JMS I LRERNG
          /
3370 0202   LRERNG, CHANGE
3371 1333   PSRERR, ERRPSR
3372 1320   ACTLIN, ERROR+5
3373 3101   KILLIT, KILL
3374 0274   RELGO, XCVT+3
3375 3034   OP1, SELUP1
3376 3133   PRGRET, RETPRG
3377 0000   0

3400   *3400

3400 5275   /
          JMP DCHNG

3401 1271   /
          SETCON, TAD M7      /ROUTINE TO SET UP CONSTRAINT WORDS
3402 3347   DCA UPDOWN
3403 1325   TAD TABLE
3404 3373   DCA MOVWDX
3405 1773   CONRAN, TAD I MOVWDX
3406 3374   DCA MVWDPG
3407 4710   JMS I XRANCN
3410 7040   CMA
3411 3774   DCA I MVWDPG
3412 2373   ISZ MOVWDX

```

```

3413 1773   TAD I MOVWDX
3414 3374   DCA MVWDPG
3415 4710   JMS I XRANCN
3416 3774   DCA I MVWDPG
3417 2373   ISZ MOVWDX
3420 2347   ISZ UPDOWN
3421 5005   JMP CONRAN
3422 3720   DCA I XDRFLG
3423 3721   DCA I XFIDFG
3424 4705   STARTP, JMS I OP1SET
3425 7344   CLA CLL CMA RAL
3426 3706   DCA I ENDCNT
3427 4707   JMS I INITF0
3430 6001   IOV           /TURN THE INTERRUPT ON
3431 5240   JMP FILL
3432 7300   CLA CLL
3433 1272   TAD INSCNT      /NUMBER OF INSTRUCTIONS BEFORE RELOCATION
3434 3722   DCA I XCNTN3
3435 5724   RESCNT, JMP I XGNFLD /GO, BABY GO !!!

3436 1274   ADDONE, TAD C10
3437 1722   TAD I XCNTN3
3440 3722   FILL, DCA I XCNTN3 /START WITH FIELD 0
3441 1722   TAD I XCNTN3 /IS THIS FIELD =TO LAST FIELD OR OVER
3442 7041   CIA
3443 1711   TAD I 2FLDLM
3444 7510   SPA
3445 5232   JMP RESCNT-3
3446 7650   SNA CLA
3447 1712   TAD I ZUPLIM /YES, SET UPPER LIMITS (=1777- -7777)
3450 7041   CIA
3451 3713   DCA I ZHIGH /IF NOT LAST FIELD UPPER LIMITS=0
3452 6224   RIF /READ THE INSTRUCTION FIELD
3453 7041   CIA /NEGATE THE FIELD
3454 1722   TAD I XCNTN3 /GET THE FIELD TO FILL
3455 7650   SNA CLA /IS IT EQUAL TO THE PROGRAM FIELD
3456 5715   JMP I XFLRND /YES FILL AROUND THE PROGRAM
3457 1722   TAD I XCNTN3 /GET THE FIELD TO FILL
3460 7650   SNA CLA /IS IT EQUAL TO FIELD ZERO
3461 7307   CLA CLL IAC RTL /YES ADD FOUR TO FILL COUNTER
3462 1713   TAD I ZHIGH /GET COUNTER
3463 3713   DCA I ZHIGH /RESTORE IT
3464 1722   TAD I XCNTN3 /GET THE FIELD TO FILL
3465 7650   SNA CLA
3466 7307   CLA CLL IAC RTL /START FILLING AT ADDRESS 4 FOR FIELD 0
3467 4714   JMS I XFILL /FILL THE WHOLE FIELD
3470 5236   JMP ADDONE

3471 7771   M7, =7
3472 0000   INSCNT, 0
3473 0200   K200, 200
3474 0010   C10, 10

3475   *3475

```



```

3475 1720 DGHNG, TAD I XDRFLG
3476 7640 SZA CLA
3477 1317 TAD XSWAP2
3500 7450 SMA
3501 1316 TAD XSWAP1
3502 3347 DCA UPDWN
3503 4704 JMS I DRERNG
3504 0202 DRERNG, CHANGE
3505 3035 OP1SET, SETOP1
3506 3303 ENDCNT, CNTEND
3507 0337 INITF0, F0INIT
3510 2127 XRANON, RANCON
3511 1144 ZFLDLM, FL0LIM
3512 1550 ZUPLIM, UPRLIM
3513 1145 ZHIGH, HGHLIM
3514 0527 XFILL, FILALL
3515 3001 XFLRND, FLRND
3516 0216 XSWAP1, SWAP1
3517 3550 XSWAP2, SWAP2
3520 0213 XDRFLG, DIRFLG
3521 0500 XFDFG, FLDFLG
3522 0247 XGNTR3, CNTR3
3523 0200 PRGBGN, BGV
3524 1001 XGNFLD, GENFLD
3525 3526 TABLE, XFDMSK
3526 1127 XFDMSK, FLJMSK
3527 1130 XGNFLD, CONFLD
3530 1131 XADRS, ADRMSK
3531 1132 XGNADR, CONADR
3532 1642 XINSMS, INSMK
3533 1643 XGNINS, CONINS
3534 1133 XINDMS, INDMSK
3535 1134 XGNIND, CONIND
3536 1135 XMDTMS, MDTHSK
3537 1136 XGNMDT, CONMDT
3540 1137 XACDMS, ACDMSK
3541 1140 XGNACD, CONACD
3542 1141 XHQDMS, HQDMSK
3543 1142 XGNHQD, CONHQD
3544 3424 XRSCNT, STARTP
3545 0000 0
3546 5747 JMP I ,+1
3547 7402 HLT /ADDRESS OF SWAP ROUTINE
3550 1375 SWAP2, TAD SIZPRG
3551 3347 DCA UPDWN
3552 1323 TAD PRGBGN
3553 3373 DCA MOVWDX
3554 1323 TAD PRGBGN
3555 1273 TAD K200
3556 3374 DCA MVWDPG
3557 1774 MOVWDX, TAD I MVWDPG
3560 3773 DCA I MOVWDX
3561 1774 TAD I MVWDPG /COMPARE THE WORDS BEING RELOCATED

```

```

3562 7041 CIA
3563 1773 TAD I MOVWDX
3564 7640 SZA CLA
3565 7402 HLT /COMPARE ERROR DURING RELOCATION
3566 2373 ISZ MOVWDX
3567 2374 ISZ MVWDPG
3570 2347 ISZ UPDWN
3571 5357 JMP MOVWDX
3572 5744 JMP I XRSCNT
3573 0000 MOVWDX, 0
3574 0000 MVWDPG, 0
3575 4401 SIZPRG, BGV-PRGEND-1
3576 3576 PRGEND, .

0200 *200
0201 4023 JMS PATCH
0201 4023 JMS PATCH

3600 *3600
/
3600 1021 PATCH2, TAD OP1SEL /GET THE HARDWARE CONFIGURATION
3601 7004 RAL /PUT OPTION1 BIT IN BIT 0
3602 7700 SMA CLA /IS OPTION 1 SELECTED?
3603 5233 JMP SLOWRN /NO, OVERLAY SECTIONS OF RANDY AND ERROR1
3604 1276 TAD K4772 /YES, SET UP A TALK LOOP TO PROM
3605 3677 DCA I OP1HND /PUT JMS I ACTLIN IN RTCSE+2
3606 1300 TAD OP1OVR
3607 3010 DCA AUTO10
3610 1301 TAD ERROR5
3611 3011 DCA AUTO11
3612 1250 TAD HM20
3613 3145 DCA PATHOV
3614 1410 TAD I AUTO10
3615 3411 DCA I AUTO11
3616 2145 ISZ PATHOV
3617 5214 JMP ,=0
3620 1275 TAD R5771
3621 3651 DCA I OVR1
3622 1275 TAD R5771
3623 3652 DCA I OVR2
3624 1275 TAD R5771
3625 3653 DCA I OVR3
3626 1275 TAD R5771
3627 3653 DCA I OVR3
3630 1275 TAD R5771
3631 3654 DCA I OVR4
3632 5423 JMP I PATCH
/THIS SECTION OF CODE WILL OVERLAY LAST 2 LOCATIONS OF RANDY ALL OF ERROR1 EXCEPT LAST 2 LOC.

3633 1246 SLOWRN, TAD ACTOVR
3634 3010 DCA AUTO10
3635 1247 TAD STRRND

```

3636	3011	DCA	AUTO11		
3637	1250	TAD	MM20		
3640	3145	DCA	PATHOV		
3641	1410	TAD	AUTO10		
3642	3411	DCA	AUTO11		
3643	2145	ISE	PATHOV		
3644	5241	JMP	,=3		
3645	5423	JMP	PATCH		
3646	3654	ACTOVR,	ACGDOV=1		
3647	1412	SFRRND,	RAVDY1=1		
3650	7760	MM20,	=20		
3651	3213	OVR1,	RECSER+7		
3652	3244	OVR2,	PIOSER+5		
3653	3251	OVR3,	PIOSER+12		
3654	3323	OVR4,	RESET=1		
3655	2230	/	ACGDOV,	2230	
3656	5226			5226	
3657	2231			2231	
3660	5226			5226	
3661	6002			10F	
3662	7240			CLA	CMA
3663	3231			3231	
3664	3230			3230	
3665	6272			CIF	70
3666	4632			4632	
3667	6001			10V	
3670	1344	CONTRD,		1344	
3671	5601			5601	
3672	0000			0	
3673	7777			=1	
3674	6500			6500	
3675	5771	R5771,		5771	
3676	4772	K4772,		4772	
3677	3257	OP1HND,		RTCSER+2	
3700	3701	OP1OVR,		OVR0P1=1	
3701	1317	ERROR0,		ERROR+4	
3702	0000	/		OVR0P1,	0
3703	2331			2331	
3704	5720			5720	
3705	1332			1332	
3706	3331			3331	
3707	6272			CIF	70
3710	4730			4730	
3711	5720			5720	
3712	6500			6500	
3713	7634			=144	
3714	7634			=144	
3715	6002			10F	
3716	1333			1333	
3717	6272			CIF	70

3720	5737	5737
3721	6520	6520
	0200	*200
		5

```

0000 11111100 11000000 11111111 11111111 11111111 11111111 11111111 11111111
0100 11111111 11111111 11111111 11111111 11111111 11111100 00000000 00000000

0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 11110111 11111111 11111111

0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111111 01111111 11111111 11111111

0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111111 11111111 11111011 11111111

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111101 11111111 11111111 11111111

1400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

1600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1700 11111111 11111111 11111111 11111111 11111001 11111111 11111111 11111111

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111101 11111111 11111111 11111111

2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 11111111 10011111 11111111 11111111 11111111

2400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111111 11111111 11111111 11111111 11111111 11111111 11111101 11111111

2600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2700 11111111 11111111 11111111 11111111 11111111 11111111 00001111 11111111

3000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

3200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111111 11111111 11111111 11111111 11111111 11111111 11111101 11111111

3400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111110

3600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3700 11111111 11111111 11000000 00000000 00000000 00000000 00000000 00000000

```

```

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

```

A1	1653	BCHNG	1347	CONACO	1140	ERROR1	1415
A14	1651	BDATHR	1376	CONADR	1132	ERROR2	0724
A1CHG	1190	BDATTH	1375	CONFLD	1130	ERROR5	3701
A1RRNG	1191	BEGIN	0547	CONFLG	1143	ERRPR	1333
A200	1647	BGN	0200	CONIND	1134	ERRRET	1392
A400	1124	BGNCON	3026	CONINS	1643	EXPRET	2754
A6201	2144	BGOTST	0524	CONMDT	1136	F0INIT	0337
A7	0743	BINSTR	0540	CONMOD	1142	FCHNG	1747
AACHNG	0551	BLPCNT	1365	CONRAN	3405	FILALL	0527
AADDRS	1166	BNDCON	1551	CONSET	3162	FILL	3440
AASAVA	1760	BNDOK1	1445	CONST1	1105	FILRND	3001
AASAVB	1761	BNDOK2	1457	CONST2	1112	FINSTR	1367
ABGN	3173	BNDRY1	1435	CONTLN	3171	FLOCHK	2555
ABNRY1	1164	BP6	1654	CONTRD	3670	FLOCNT	3172
ACDATA	1031	BRANDY	1755	COPRST	2773	FLODF0F	1153
ACDMSK	1137	BREFAD	1374	CREFAD	1574	FLODF0R	1152
ACGDOV	3655	BRERNG	1350	CRERNG	1563	FLODFLG	0500
ACHG	0766	BSAVA	2145	CSAVA	1553	FLODLIM	1144
ACHNG	0355	BSIMAC	2575	CSAVB	1554	FLODMBK	1127
ACL	7701	BSMLNK	2576	CSAVC	1555	FLODRAN	1156
ACTLIN	3372	C10	3474	CSIMAC	2770	FLODRET	0761
ACTOVR	3646	C1777	1561	CSIMMQ	2772	FLODXRN	2157
ACUTST	2623	C200	1312	CSMLNK	2771	FLGPI0	3157
ADATAH	1172	C400	1547	CSWPUP	0553	FLGREC	3156
ADATAT	1171	C5	1596	D4000	2625	FLGRTC	3160
ADD11	1126	C6201	1123	D6201	2070	FLGXMT	3155
ADDONE	3436	C7	1557	DATADN	2750	FLINK	0771
ADDRS	0745	C70	1560	DATADR	0752	FLXRET	2160
ADDRS0	3143	CAADDRS	1575	DATATH	0751	FMQDAT	0772
ADDRS5	1612	CAF	6007	DATFN	1372	FRERNG	1750
ADMSK	1131	CAL	6103	DBCE	6576	GACTRT	2366
AEROV1	0113	CC377	3236	DBCF	6573	GADDRS	2367
AEROV2	0120	CCHNG	1562	DBRD	6572	GCHNG	2147
AINSTR	1167	CDATAT	1567	DBSE	6575	GDATA0	2371
ALOPID	1176	CDHLT1	0540	DBSK	6571	GDATAH	2370
AMQDAT	1173	CDHLT2	0544	DBSS	6577	GENADD	1043
ANDTST	1201	CHANGE	0202	DBST	6570	GENFLD	1001
ANTIND	1175	CHK1KF	1154	DBTD	6574	GENIND	1072
ARANDY	1163	CHKACT	3313	DCATST	1255	GENINS	1055
AREFAD	1170	CIFFD0	0345	DCHNG	3475	GENMOD	1037
ARERNG	0767	CINSTR	2162	DECSWP	1116	GETW0	2112
ARRANG	0305	CLASWP	2671	DINSTR	2345	GLNKDN	2374
ASAME1	1161	CLCL	6136	DIRFLG	0213	GLPSW0	1776
ASAME2	1162	CLLE	6135	DONEMQ	1362	GMQDAT	2375
ASAVA	1146	CLSK	6137	DREERNG	3504	GMQDN	1775
ASAVB	1147	CNT	2425	DVINAC	3161	GOPRET	2365
AUTO10	0010	CNTEND	3303	EINSTR	1794	GRANFL	2363
AUTO11	0011	CNTR2	0246	ENDCMA	2422	GRERNG	2150
B400	1650	CNTR3	0247	ENDCNT	3506	GSMLNK	2373
B6201	0526	COMCHK	2227	ENDOFF	3175	GSVLNK	2372
B7	0543	COMPAR	1206	ERROPR	2306	HALT	1336
BADDRS	1366	CON37	0134	ERROR	1313	HCHNG	2343

HGHLIM	1145	K5771	0136	MIN37	0135	OPRSKP	0131
HIGHLM	0374	K5772	0137	MM20	3650	OSIMAC	2173
HLT	7402	K6201	0756	MM4	3023	OSMLNK	2174
HLTFIL	0466	K6202	0757	MM55	3077	OUT	3271
HLTOPR	1741	K70	1125	MM6	0142	OVR1	3651
HMOCIF	0744	K7000	1644	MM7	1545	OVR2	3652
HRERNG	2344	K7610	0132	MMH55	3340	OVR3	3653
ICHNG	2567	K7700	2517	MOVOWN	3557	OVR4	3654
ILLMO	1703	K7721	1655	MOVQVR	0103	OVRLAY	0125
ILLOP1	1664	K7770	0760	MOVUP	0225	OVRLY1	0127
ILLOP2	1672	K7777	2624	MOVWDX	3573	OVR0P1	3702
INACOV	3337	KCHNG	3146	MQDAT	1363	PATCH	0023
INDAD	0750	KIE	6035	MQDATA	0753	PATCH1	0052
INDMSK	1133	KILL	3101	MQDMSK	1141	PATCH2	3600
INITF0	3507	KILLIT	3373	MQDONE	2753	PATCHC	0102
INSCNT	3472	KJMS	0764	MQL	7421	PATMOV	0145
INSGEN	1601	KK377	3100	MRIERR	2161	PATMV1	0146
INSM5K	1642	KKCDF	3145	MRIQVR	0126	PCSAVE	1311
INSOK	1520	KKK4	3024	MRIPTNT	1751	PCTS?	2073
INSTR	0746	KLAS	0147	MVWDPG	3574	PI0FLG	3230
INT	0003	KRERNG	3147	NEG10	2072	PI0RBY	3163
INTAC	3141	LAS	7604	NEG14	1652	PI0SER	3237
INTERS	3102	LASCNT	0143	NEG20	2304	PI0XMT	3154
INTLNK	3142	LASTAB	0144	NEG6	2460	POS10	2250
INTMOD	1757	LCHNG	3367	NEWDF0	0475	POS200	2071
INTOPR	2175	LIMITS	1522	NEWDTF	0457	POWERF	3164
INTRET	3144	LINKDN	2791	NOTAUT	0611	POWFAL	3304
INTSET	2137	LINKRT	1361	NOTIND	0625	PRGBG	1576
IRERNG	2570	LINKSV	1360	NOTJJ	0650	PRGBGN	3523
ISZTST	1234	LKSAV	1174	O6201	1733	PRGEND	3576
JCHNG	2764	LOC200	0140	OADDRS	1773	PRGRET	3376
JMPJMS	0665	LOC201	0141	OERR0R	2767	PRGSIZ	1552
JMPTST	1304	LOOPID	0601	OFIELD	2176	PSRERR	3371
JMSLOC	0762	LOOPSW	1342	OP1	3375	R5771	3675
JMSLOD	0763	LOWLIM	0214	OP1HND	3677	RAN1	1543
JMSRET	0776	LPNT	0252	OP1OVR	3700	RAN2	1544
JMSTST	1267	LPSW0	2737	OP1SEL	0021	RANCON	2127
JRERNG	2765	LRERNG	3370	OP1SET	3505	RANDOM	3341
K1	2061	M10	0503	OP2SEL	0022	RANDY	1401
K100	2067	M13	2536	OPERR1	1734	RANDY1	1413
K14	2064	M14	2424	OPRAND	3152	RANFLD	0754
K177	1646	M1777	2565	OPRBGN	1656	RANN1	3355
K2	2062	M20	0502	OPRCOM	2706	RANN2	3356
K20	2065	M200	0251	OPRER1	2376	RDYPI0	3254
K200	3473	M400	0215	OPRERR	2744	RECFLG	3227
K37	1546	M5	0133	OPRETF	2752	RECPI0	3235
K4	2063	M60	2305	OPRHLT	2364	RECSER	3204
K40	2066	M6000	1645	OPRINT	0637	RECSLU	3233
K400	0250	M7	3471	OPROVR	0130	REFAD	0747
K4000	2434	MAXFLD	0573	OPRPNT	1752	RELGO	3374
K4772	3676	MDTMSK	1135	OPRRET	2675	RESCNT	3435
		MEMDAT	1025	OPRSET	2054	RESET	3324

RESETI	0256	SIWOP1	2001	WAIT	0360	XHODMS	3542
RETFLD	0525	SIWOP2	2201	WAITEN	3397	XMTFLG	3226
RETHR	0504	SIWRA1	2461	XACDMS	3540	XMTPIO	3234
RETPNT	0004	SIWRA2	2442	XACSIM	2392	XMTSER	3201
RETPRG	3133	SIWRTL	2937	XADD1	3176	XMTSLU	3232
RETTDF	2705	SIWRTL	2920	XADRMS	3530	XNTIND	0363
RETURN	1354	SIWMA	2611	XASAVA	0364	XPCSAV	2155
RTFIA	3176	SIWNL	2626	XASAVB	0365	XRANGN	3510
RHF	6244	SIWSP	2692	XBGCON	0372	XREFAD	0366
ROLBAK	0401	SIWSEA	2601	XBGRAN	0371	XRETFI	1373
ROLFLG	2574	SIWTHC	2042	XBNDCN	1160	XRETHR	1165
ROLLUP	0320	SIWPRG	3575	XBSAVA	1396	XRETPC	2196
RSCNT	0376	SKPFLG	2291	XCHNGE	0592	XRNFLD	1364
RSCNTX	0370	SLDWRN	3633	XCLSWP	2362	XROLBK	0361
RTDFLG	3231	SLJXMT	3153	XCNACD	3541	XRSCNT	3544
RTCSER	3255	SMACHK	2202	XCNADR	3531	XRTOPF	1774
RTFLG	0572	SNLCHK	2220	XCNFLD	3527	XSIMAC	1763
RTFLGF	0574	SP	6102	XCNFLG	3190	XSIMHQ	1766
RTFLGR	0571	SRSVA	2766	XCNIND	3535	XSIZE	0575
RTLINK	0557	START	1756	XCNINS	3533	XSMAGL	2361
RTMOD	0556	STARTP	3424	XCNHDT	3537	XSMBSW	2170
SAME1	1463	STRFLD	0005	XCNHOD	3543	XSMCAM	2360
SAME2	1473	STRND	3647	XCNT	0271	XSMCLA	2357
SAVESW	1564	SUBADD	0212	XCNTRJ	3522	XSMCMA	2163
SAVLNK	0795	SWAP1	0216	XDATAH	1762	XSMCML	2164
SAVSWR	0336	SWAP2	3550	XDRFLG	3520	XSMIAC	2165
SBE	6101	SWAPON	0422	XENDPR	0375	XSMLNK	1765
SELOP1	3034	SWAPUP	0415	XERR2	1972	XSMMOA	2355
SERPIO	3167	SWITCH	0020	XERROR	1393	XSMMO1	1772
SERREC	3166	SWFLD	0463	XEXPT	1767	XSMOQL	2354
SERRTC	3170	SWPUP	0452	XFDMSK	3526	XSMOP1	1770
SERXMT	3165	SWRSVA	1351	XFIELD	2775	XSMOP2	1771
SETCON	3401	SEACHK	2211	XFILL	3514	XSMRAL	2167
SETFLG	0327	SEPRG	0245	XFLD	0396	XSMRAR	2166
SETINT	0357	TA3LAS	0150	XFLDFG	3521	XSMRTL	2172
SETOP1	3035	TA3LC	3525	XFLDLN	1573	XSMRTR	2171
SETRET	0710	TAJST	1223	XFLRND	3515	XSMSMA	2346
SETSCKP	2226	TEHP	3025	XGENTI	1195	XSMSNL	2350
SFLDFG	0433	TINT	2143	XGETWO	1397	XSMSWP	2356
SIMAC	2795	TSME	1503	XGNFLD	3524	XMSZA	2347
SIMACL	2666	TSFLOF	2572	XGO	0576	XSTFLD	0554
SIMBSW	2477	TSTIN1	0562	XHALT	1565	XSVLNK	1764
SIMCAN	2663	TSTIN2	0563	XINDMS	3534	XSWAP1	3516
SIMCLA	2661	TSTIN3	0564	XINSCN	2774	XSWAP2	3517
SIMCLR	6160	TSTIN4	0565	XINSMX	3532	XUPERL	0373
SIMCMA	2401	TSTIN5	0566	XINSTR	0342	XUPLIM	1157
SIMCML	2426	TSTINS	0561	XINT	2191	XWMOV	2153
SIMIAC	2435	TSTIPC	1355	XJMSLO	2154	XXCNT	2776
SIMLNK	2756	TSAPDN	2573	XLIMIT	3151	YHALT	0770
SIMMQ	2757	UPJMN	3547	XLOOP	0367	Z100	2243
SIMMQA	2644	UPPERL	2571	XLWLM	0570	Z20	2245
SIMMQI	2252	UPRLIM	1550	XMDTMS	3536	Z200	2246
SIMMQL	2640						

Z320	2247
Z40	2244
ZASAVA	1370
ZASAVB	1371
ZBSAVA	1570
ZCNFLG	2152
ZCNT	0775
ZEXPT	2351
ZFILL	3174
ZFIND	0774
ZFLDLN	3511
ZGETWO	0773
ZHIGH	3513
ZINDAD	1571
ZINSTR	1566
ZJMSRT	1753
ZLIMIT	0555
ZSETOP	2353
ZUPLIM	3512

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 14 SECONDS

3K CORE USED